Сергей Кашаев

1С:Предприятие 8.1 учимся программировать на примерах

4-е издание

Санкт-Петербург «БХВ-Петербург» 2011

Кашаев С. М.

К31 1С:Предприятие 8.1. Учимся программировать на примерах. — 4-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2011. — 368 с.: ил. + CD-ROM

ISBN 978-5-9775-0701-1

Обучение разработке конфигураций и программированию на платформе 1С:Предприятие 8.1 построено на практических примерах различной сложности. Основной упор сделан на подробное описание технологии разработки конфигураций с использованием конструкторов и встроенного языка программирования. Большое внимание уделено языку запросов и построению разнообразных отчетов, передаче информации между системой 1С:Предприятие 8.1 и популярными офисными приложениями и др. Тематика примеров в основном связана с учетом денежных средств, в частности, рассмотрены задачи автоматизации учета денежных поступлений для организаций, предоставляющих платные образовательные услуги.

В четвертом издании добавлена глава по особенностям новой платформы 1С:Предприятие 8.2 и включен материал по оформлению в цвете табличных частей справочников и документов.

Компакт-диск содержит разработанные в книге конфигурации, а также упоминаемые документы, созданные в приложениях Microsoft Word и Excel.

Для начинающих 1С-программистов и опытных пользователей

УДК 681.3.06 ББК 32.973.26-018.2

Главный редактор	Екатерина Кондукова
Зам. главного редактора	Евгений Рыбаков
Зав. редакцией	Григорий Добин
Редактор	Анна Кузьмина
Компьютерная верстка	Ольги Сергиенко
Корректор	Зинаида Дмитриева
Дизайн серии	Игоря Цырульникова
Оформление обложки	Елены Беляевой
Зав. производством	Николай Тверских

Группа подготовки издания:

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.01.11. Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 29,67. Тираж 2500 экз. Заказ № "БХВ-Петербург", 190005. Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

> Отпечатано с готовых диапозитивов в ГУП "Типография "Наука" 199034, Санкт-Петербург, 9 линия, 12

Оглавление

Введение	7
Читателям четвертого издания книги	8
Общий взгляд на систему 1С:Предприятие	8
Краткое содержание книги	11
Компакт-диск	12
От автора	13
Глава 1. Представление об объектах системы 1С:Предприятие 1	15
Архитектура системы 1С:Предприятие	16
Взгляд на встроенный язык программирования	18
Постановка задачи	21
Окно запуска	21
Создание новой информационной базы	22
Перечисления	25
Справочники	27
Справочник Должности	28
Справочник Специалисты	30
Справочник Курсы	38
Печатная форма справочника	45
Внесение изменений в печатную форму	53
Справочник Клиенты	58
Документы	56
Документ Внесение оплаты	56
Заполнение закладки Данные	57
Форма документа	58
Программирование формы документа	72
Создание печатной формы документа ВнесениеОплаты	75
Список документов	76
Отчеты	77
Разработка макета	78
Процедура заполнения отчета	32

Отчет по клиентам	85
Отчет по клиентам с детализацией курсов	
Итоги знакомства с конфигуратором	
Глава 2. Документы	
Подготовка конфигурации	
Документ Заказ	
Программное создание и запись документов	
Поиск документов	
Включение в условие отбора параметра	100
Упорядочивание	102
Программная установка свойств документов	103
Заполнение документа на основании имеющихся данных	104
Обработка событий	107
Глара 3 Располог накондания	108
1 лава 3. 1 стистры накопления	100
Изменения в покументе Поступлания	110
Расход по регистру накопления	
Оборотный регистр накопления	
Получение выборки информации из регистра	
Получение выобрки информации из регистра	121
	122
Глава 4. Регистры сведений	125
Организация непериодических регистров сведений	125
Создание справочника Номенклатура	128
Документ Поступление	128
Процедура установки ответственного в поле документа	130
Создание периодического регистра сведений Цены	132
Использование регистра Цены в документе Поступление	134
Глава 5 Программирование форм	136
Постановка запачи	136
Представление информации о тарифах	
Вепомогательный справонных ВидТарифа	130
Веномогательный справочник <i>Виотарифи</i>	140
Контроль ошибок в форме регистра Тарифы	140 1/1
Справонник Логоворы	1/13
Справо плик Досоворог Регистр свелений Кентаниции	144
Форма обработки иля учета кританний по оплате	1/16
Форма обработки для учета квитанции по оплате	
Процедуры для работы с формон	
Процедура, выполняемая при высоре договора	
Процетура расцета суммы кританным	
процедура растота суммы квитанции. Запись информации в регистр <i>Кецтанции</i>	159
Форма регистрации квитанний	159
Формирование документа Microsoft Word	101
Формирование документа инстозон и она	105
Разработка процедуры заполнения бланка	104
Заполнение бланка с лвумя квитанниями	107
Эаполныние олапка с двушл квитапцияши	170

Нзык запросов. Основные конструкции. 174 Обработка ИзвелечениеИИ формации. 175 Конструкции ВЫБР АТЬ и ИЗ	Глава 6. Извлечение информации с помощью запросов	
Обработка ИзвлечениеИнформации 175 Конструкции ВЫБР АТЬ и ИЗ	Язык запросов. Основные конструкции	
Конструкции ВЫБРАТЬ и ИЗ	Обработка ИзвлечениеИнформации	
Упорядочивание в запросе	Конструкции ВЫБРАТЬ и ИЗ	
Отбор максимальных (минимальных) значений 180 Ключевое слово <i>ПЕРАРХИЯ</i> 182 Исключение в отборе повторяющейся информации. 183 Ключевое слово <i>ГДЕ</i> 184 Использование логических операторов в запросе. 186 Аргетатные функции в запросах 188 Исключение в отборе в запросах 192 Ключевое слово <i>МЕЖДУ</i> 194 Функция расчета итогов в запросах 197 Объединение результатов нескольких запросов 200 Боле сложные конструкции языка запросов 202 Отбор специалистов по указанной должности 202 Отбор курсов преподавателя 205 Соединение табли при организации запроса 206 Окоре внешнее соединение 210 Полное внешнее соединение 210 Полючевое слово <i>ВЫБОР</i> 213 Зстроенные функция языка запросов 216 Функция <i>ПОДСТРОКА</i> 216 Функция <i>ПОДСТРОКА</i> 216 Функция <i>ПОДСТРОКА</i> 216 Функция <i>ПОДСКДАте.</i> 216 Функция <i>ПРЕДСТАВЛЕНИЕ</i> 219 Опч	Упорядочивание в запросе	
Ключевое слово ИЕРАРХИЯ 182 Исключение в отборе повторяющейся информации. 183 Ключевое слово ГДЕ. 184 Использование погических операторов в запросе. 186 Агрегатные функции в запросах. 182 Использование параметров в запросах. 192 Ключевое слово МЕЖДУ. 194 Функция расчета итогов в запросах. 197 Объединение результатов нескольких запросов. 200 Огор специалистов по указанной должности. 202 Отбор специалистов по указанной должности. 202 Отбор курсов преподавателя. 205 Соединение таблиц при организации запроса. 206 Левое внешнее соединение. 210 Полюе ввешешее соединение. 210 Полюе внешнее соединение. 210 Полюе внешнее соединение. 213 Зстроенные функции языка запросов. 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 219 Результат использования запросов. 221 Слючевое слово ВЫБОР 212 Отчет по остаткам товаров. 221 Лич	Отбор максимальных (минимальных) значений	
Исключение в отборе повторяющейся информации. 183 Ключевое слово Г/ДЕ. 184 Использование логических операторов в запросе. 186 Агретатные функции в запросах 188 Использование параметров в запросах 192 Ключевое слово МЕЖДУ 194 Функция расчета итогов в запросах 197 Объединение результатов нескольких запросов 200 Боле с сложные конструкции языка запросов. 202 Отбор специалистов по указанной должности 202 Отбор курсов преподавателя. 205 Соединение таблиц при организации запроса 206 Левое внешнее соединение 208 Правое внешнее соединение 212 Слючевое слово <i>BL/БОР</i> 213 Зстроенные функции языка запросов 216 Функция <i>ПОДСТРОКА</i> 216 Функция <i>ПОДСТРОКА</i> 216 Функция <i>ЛобавитьКДате</i> 217 Функция <i>ПОДСТРОКА</i> 218 Функция <i>ПОДСТРОКА</i> 219 Раультат использования запросов 212 Спочевое слово <i>BL/БОР</i> 213 Встроенные функция <i>ПОДСТРОКА</i> 216 Функция	Ключевое слово ИЕРАРХИЯ	
Ключевое слово ГДЕ	Исключение в отборе повторяющейся информации	
Использование логических операторов в запросе	Ключевое слово ГДЕ	
Агрегатные функции в запросах. 188 Использование параметров в запросах 192 Ключевое слово <i>МЕЖДУ</i> 194 Функция расчета и погов в запросах 197 Объединение результатов нескольких запросов 200 Более сложные конструкции языка запросов 202 Отбор специалистов по указанной должности 202 Отбор специалистов по указанной должности 202 Отбор курсов преподавателя. 205 Соединение соединение 206 Левое внешнее соединение 208 Правое внешнее соединение 210 Полное внешнее соединение 212 Слючевое слово <i>BbISOP</i> 213 Вастроенные функции языка запросов 216 Функция <i>ЛОДСТРОКА</i> 216 Функция <i>ЛодавитьКДате</i> 217 Функция <i>ЛодовитьКДате</i> 218 Функция <i>ЛодСТРОКА</i> 219 Средатоть/дательсова 212 Слочевое словования запросов 219 Результат использования запросов 219 Результат использования запросов 221 Става 7. Отчеты и система компоновки данных 222 Отчет по остат	Использование логических операторов в запросе	
Использование параметров в запросах 192 Ключевое слово МЕЖДУ 194 Функция расчета итогов в запросах 197 Объединение результатов нескольких запросов 200 Солее сложные конструкции языка запросов 202 Отбор специалистов по указанной должности 202 Отбор курсов преподавателя 205 Соединение таблиц при организации запроса 206 Левое внешнее соединение 208 Правое внешнее соединение 210 Полное внешнее соединение 212 Слючевое слово ВЫБОР 213 Зстроенные функции языка запросов 213 Зстроенные функции языка запросов 216 Функция ЛОДСТРОКА 216 Функция Лобавить Дат 218 Функция ЛОДСТРОКА 219 Оператор ПОДОБНО 219 Ореатор ПОДОБНО 219 Отчет по остаткам товаров. 221 Стиотьзования запросов 232 Отчет в виде таблицы 235 Отчет в виде таблицы 235 Отчет в виде дайлицы 235 Отчет по остаткам товаров. 232 Отчет	Агрегатные функции в запросах	
Ключевое слово МЕЖДУ 194 Функция расчета итогов в запросах 197 Объединение результатов нескольких запросов 200 Солбор специалистов по указанной должности 202 Отбор курсов преподавателя 205 Соединение таблиц при организации запроса 206 Левое внешнее соединение 208 Правое внешнее соединение 208 Полное внешнее соединение 212 Спочевое слово ВЫБОР 213 Застроенные функции языка запросов 216 Функция ЛОДСТРОКА 216 Функция ЛАССАВИТ 217 Функция ЛАССАВИТ 218 Функция ЛАССАВИТ 219 Оператор ПОДОБНО 219 Оператор ПОДОБНО 219 Оператор Виде диаграммы 235 Отчеть в виде диаграммы 235 Отчеть в виде диаграмм	Использование параметров в запросах	
Функция расчета итогов в запросах 197 Объединение результатов нескольких запросов 200 Солее сложные конструкции языка запросов 202 Отбор курсов преподавателя 202 Отбор курсов преподавателя 205 Соединение таблиц при организации запроса 206 Левое внешнее соединение 208 Правое внешнее соединение 210 Слючевое слово ВЫБОР 213 Зстроенные функции языка запросов 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 219 Оператор ПОДОБНО 219 Оператор ПОДОБНО 219 Орчеть и система компоновки данных. 222 Отчеть и виде таблицы 235 Отчет по остаткам товаров 224 Аспользование конструкторов настроек. 232 Отчет по документам Заявка 249 Орчеть по оплаченным Квитанциям 245 Отчет по оплаченным квитанциям. 244 Отчет по оплаченным квитанциям. 245 Отчет по оплаченным квитанциям. 246 Отчет по квитанциям с группировкой 256 <td>Ключевое слово МЕЖДУ</td> <td></td>	Ключевое слово МЕЖДУ	
Объединение результатов нескольких запросов 200 Более сложные конструкции языка запросов. 202 Отбор специалистов по указанной должности 202 Отбор курсов преподавателя. 205 Соединение таблиц при организации запроса 206 Левое внешнее соединение 208 Правое внешнее соединение 210 Полное внешнее соединение 212 Слючевое слово ВЫБОР 213 Встроенные функции языка запросов 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 217 Функция ЛОДСТРОКА 218 Функция ЛОДОБНО 219 Оператор ПОДОБНО 219 Оператор ПОДОБНО 219 Результат и спользования запросов 221 Глава 7. Отчеты и система компоновки данных 222 Отчет по остаткам товаров 239 Рормирование отчета по поступлениям с итоговой информацией 241 Формирование в отчете двух таблиц 245 Отчет по одкументам Заявка 245 Отчет по одкументам Заявка 244 Отчет по одкументам Заявка	Функция расчета итогов в запросах	197
Более сложные конструкции языка запросов 202 Отбор специалистов по указанной должности 202 Отбор курсов преподавателя 205 Соединение таблиц при организации запроса 206 Левое внешнее соединение 208 Правое внешнее соединение 210 Полное внешнее соединение 212 Слючевое слово ВЫБОР 213 Встроеные функции языка запросов 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 217 Функция ЛобавитьКДате. 217 Функция РазностьДат 218 Функция ПРЕДСТАВЛЕНИЕ 219 Результат и спользования запросов 221 Глава 7. Отчеты и система компоновки данных 222 Отчет по остаткам товаров 224 Использование конструкторов настроек 235 Отчет в виде таблицы 235 Отчет по остаткам Заявка 245 Отчет по одкументам Заявка 245 Отчет по одкументам Заявка 244 Ориерование в отчете двух таблиц 245	Объединение результатов нескольких запросов	
Отбор специалистов по указанной должности 202 Отбор курсов преподавателя. 205 Соединение таблиц при организации запроса 206 Левое внешнее соединение 208 Правое внешнее соединение 210 Полное внешнее соединение 211 Спочевое слово <i>BЫБОР</i> 213 Встроенные функции языка запросов 216 Функция <i>ПОДСТРОКА</i> 216 Функция <i>ЛобавитьКДате</i> 217 Функция <i>РазностьДат</i> 218 Функция <i>ПОДСТАА</i> 219 Лератор <i>ПОДОБНО</i> 219 Результат использования запросов 221 Стчет по остаткам товаров 222 Отчет в виде таблицы 235 Отчет в виде таблицы 235 Отчет по оступлениям с итоговой информацией 244 Спользование конструкторов настроек 232 Отчет по документа <i>Заявка</i> 245 Отчет по одкумента <i>Заявка</i> 245 <t< td=""><td>Более сложные конструкции языка запросов</td><td></td></t<>	Более сложные конструкции языка запросов	
Отбор курсов преподавателя	Отбор специалистов по указанной должности	
Соединение таблиц при организации запроса 206 Левое внешнее соединение 208 Правое внешнее соединение 210 Полное внешнее соединение 212 Слючевое слово ВЫБОР 213 Застроенные функция языка запросов 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 217 Функция ЛОДСТРОКА 218 Функция ЛОДСТРОКА 219 Оператор ПОДОБНО 219 Результат использования запросов 221 Глава 7. Отчеты и система компоновки данных 222 Отчет по остаткам товаров 224 Аспользование конструкторов настроек 235 Отчет в виде диаграммы 245 Отчет по документам Заявека 245 Отчет по остаткам заявека 245 Отчет по остаракам. 245 Отчет по оплаченным квитанциям. 245 Отчет по оплаченным квитанциям. 254	Отбор курсов преподавателя	
Левое внешнее соединение 208 Правое внешнее соединение 210 Полное внешнее соединение 212 Спючевое слово ВЫБОР 213 Застроенные функция языка запросов 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 216 Функция Лобавить КДате 217 Функция Разность Дат. 218 Функция ПРЕДСТАВЛЕНИЕ 219 Орчетар ПОДОБНО 219 Результат использования запросов 221 Глава 7. Отчеты и система компоновки данных 222 Отчет по остаткам товаров. 221 Отчет в виде диаграммы. 235 Отчет в виде диаграммы. 235 Отчет по документам Заявска 244 Опоравние отчете двух таблиц. 245 Отчет по документам Заявска 245 Отчет по оплаченным квитанциям 245 Отчет по оплаченным квитанциям 254 Отчет по оплаченным квитанциям 258 Отчет по квитанциям с группировкой 261 Отчет по квитанциям с группировкой 261	Соединение таблиц при организации запроса	
Правое внешнее соединение 210 Полное внешнее соединение 212 Спочевое слово ВЫБОР 213 Зстроенные функции языка запросов 216 Функция ЛОДСТРОКА 217 Функция ПРЕДСТАВЛЕНИЕ 219 Оператор ПОДОБНО 219 Результат использования запросов 221 Глава 7. Отчеты и система компоновки данных 222 Отчет по остаткам товаров 224 Аспользование конструкторов настроек 232 Отчет в виде таблицы 235 Отчет в виде диаграммы 239 Рормирование отчета по поступлениям с итоговой информацией 241 Орормирование отчете двух таблиц 245 Отчет по документам Заявка 245 Отчет по оплаченным квитанциям 254 Отчет по оплаченным квитанциям 2	Левое внешнее соединение	
Полное внешнее соединение 212 Спючевое слово ВЫБОР 213 Зстроенные функции языка запросов 216 Функция ПОДСТРОКА 216 Функция ЛОДСТРОКА 217 Функция Разность/Дат 218 Функция ПРЕДСТАВЛЕНИЕ 219 Dreparop ПОДОБНО 219 Результат использования запросов. 221 Глава 7. Отчеты и система компоновки данных 222 Drчет по остаткам товаров. 224 Аспользование конструкторов настроек 232 Drчет в виде диаграммы. 235 Drчет в виде диаграммы. 239 Формирование о тчета по поступлениям с итоговой информацией 241 Формирование в отчете двух таблиц 245 Drчет по даявкам 245 Drчет по даявкам 245 Drчет по оплаченным квитанциям 254 Orчет по квитанциям с группировкой 258 Orчет по квитанциям с группировкой <td>Правое внешнее соединение</td> <td></td>	Правое внешнее соединение	
Спючевое слово ВЫБОР 213 Зстроенные функции языка запросов 216 Функция ПОДСТРОКА 216 Функция ЛОДСТРОКА 216 Функция ЛОДСТРОКА 217 Функция РазностьДат 218 Функция РазностьДат 219 Оператор ПОДОБНО 219 Результат использования запросов 221 Глава 7. Отчеты и система компоновки данных 222 Отчет по остаткам товаров 224 Использование конструкторов настроек 235 Отчет в виде диаграммы 239 Рормирование отчета по поступлениям с итоговой информацией 241 Формирование в отчете двух таблиц 245 Отчет по документам Заявка 249 Отчет по оплаченным квитанциям 254 Отчет по оплаченным квитанциям 254 Отчет по квитанциям с группировкой 261 Отчеты по информационной базе главы 1 262	Полное внешнее соединение	
Зстроенные функции языка запросов 216 Функция ПОДСТРОКА 216 Функция ДобавитьКДате 217 Функция РазностьДат 218 Функция ПРЕДСТАВЛЕНИЕ 219 Оператор ПОДОБНО 219 Результат использования запросов 221 Глава 7. Отчеты и система компоновки данных 222 Отчет по остаткам товаров 224 Использование конструкторов настроек 232 Отчет в виде таблицы 235 Отчет в виде диаграммы 239 Рормирование отчете двух таблиц 245 Отчет по документам Заявка 245 Отчет по оплаченным квитанциям 245 Отчет по полеченным квитанциям 254 Отчет по квитанциям с группировкой 254 Отчет по квитанциям с группировкой 261 Отчеты по информационной базе главы 1 262	Ключевое слово ВЫБОР	
Функция ПОДСТРОКА	Встроенные функции языка запросов	
Функция ДобавитьКДате	Функция ПОДСТРОКА	
Функция РазностьДат	Функция ДобавитьКДате	
Функция ПРЕДСТАВЛЕНИЕ 219 Оператор ПОДОБНО 219 Результат использования запросов. 221 Глава 7. Отчеты и система компоновки данных 222 Отчет по остаткам товаров. 224 Использование конструкторов настроек. 232 Отчет в виде таблицы 235 Отчет в виде диаграммы. 239 Формирование отчета по поступлениям с итоговой информацией 241 Формирование в отчете двух таблиц. 245 Отчет по документам Заявка 249 Отчет по оплаченным квитанциям 254 Отчет по договорам 254 Отчет по квитанциям с группировкой 258 Отчет по информационной базе главы 1 261 Отчеть по информационной базе главы 1 262	Функция РазностьДат	
Оператор ПОДОБНО	Функция ПРЕДСТАВЛЕНИЕ	
Результат использования запросов	Оператор ПОДОБНО	
Глава 7. Отчеты и система компоновки данных	Результат использования запросов	
Этчет по остаткам товаров	Глава 7. Отчеты и система компоновки данных	222
Использование конструкторов настроек. 232 Отчет в виде таблицы 235 Этчет в виде диаграммы. 239 Формирование отчета по поступлениям с итоговой информацией. 241 Формирование в отчете двух таблиц. 245 Этчет по документам Заявка 245 Этчет по заявкам. 249 Этчеть по информационной базе главы 5 254 Отчет по договорам 254 Отчет по договорам 258 Отчет по квитанциям с группировкой. 261 Этчеть по информационной базе главы 1 261	Отчет по остаткам товаров	224
Отчет в виде таблицы 235 Отчет в виде диаграммы. 239 Рормирование отчета по поступлениям с итоговой информацией. 241 Рормирование в отчете двух таблиц. 245 Отчет по документам Заявка 245 Отчет по заявкам. 249 Отчет по информационной базе главы 5 254 Отчет по оплаченным квитанциям 254 Отчет по договорам 258 Отчет по квитанциям с группировкой. 261 Отчеть по информационной базе главы 1 261	Использование конструкторов настроек	
Отчет в виде диаграммы	Отчет в виде таблицы	
Формирование отчета по поступлениям с итоговой информацией 241 Формирование в отчете двух таблиц 245 Отчет по документам Заявка 245 Отчет по заявкам 249 Отчеть по информационной базе главы 5 254 Отчет по оплаченным квитанциям 254 Отчет по договорам 258 Отчет по квитанциям с группировкой 261 Отчеть по информационной базе главы 1 261	Отчет в виде диаграммы	
Формирование в отчете двух таблиц	Формирование отчета по поступлениям с итоговой информацией	
Отчет по документам Заявка 245 Отчет по заявкам 249 Отчеты по информационной базе главы 5 254 Отчет по оплаченным квитанциям 254 Отчет по договорам 258 Отчет по квитанциям с группировкой 261 Отчеты по информационной базе главы 1 262	Формирование в отчете двух таблиц	
Отчет по заявкам	Отчет по документам Заявка	
Отчеты по информационной базе главы 5 254 Отчет по оплаченным квитанциям 254 Отчет по договорам 258 Отчет по квитанциям с группировкой 261 Отчеты по информационной базе главы 1 262	Отчет по заявкам	
Отчет по оплаченным квитанциям	Отчеты по информационной базе главы 5	
Отчет по договорам	Отчет по оплаченным квитанциям	
Отчет по квитанциям с группировкой	Отчет по договорам	
Отчеты по информационной базе главы 1	Отчет по квитанциям с группировкой	
	Отчеты по информационной базе главы 1	
Отчет по учащимся	Отчет по учащимся	
Отчет по курсам 264	Отчет по курсам	

I лава о. Автоматизация работы с данными	
Учет начислений преподавателям	
Информация о тарифах	
Вспомогательные справочники	271
Документ Учет	273
Процедуры обработки событий на форме документа	274
Просмотр актуальных тарифов	277
Оборотный регистр накопления	280
Движения по регистру Начисление	280
Отчет по начислению	
Реализация множественного подбора	
Использование системы компоновки данных для создания сводной таблицы	
Формирование сводной таблицы с использованием конструктора	
Передача данных между приложениями	
Взгляд на технологии обмена данными	
Использование СОМ-технологии	300
Передача информации в Microsoft Excel с использованием технологии	302
Buttomation	
Интеграния трех офисиих приложений	
Интеграция прех офисных приложении	
Использование цветов в электронных формах	
Чередование цветов в формах справочников и документов	
Глава 9. Анализ данных	320
Создание новой конфигурации	321
Создание новой конфигурации	321 325
Создание новой конфигурации Общая статистика Поиск ассоциаций	321 325 329
Создание новой конфигурации Общая статистика Поиск ассоциаций Поиск последовательностей	321 325 329 331
Создание новой конфигурации Общая статистика Поиск ассоциаций Поиск последовательностей Статистика в учебном центре	321 325 329 331 333
Создание новой конфигурации Общая статистика Поиск ассоциаций Поиск последовательностей Статистика в учебном центре Статистические отчеты к главе 2	321 325 329 331 333 341
Создание новой конфигурации Общая статистика Поиск ассоциаций Поиск последовательностей Статистика в учебном центре Статистические отчеты к главе 2 Глава 10. Знакомимся с версией 1С:Предприятие 8.2	
Создание новой конфигурации Общая статистика	
Создание новой конфигурации	
Создание новой конфигурации Общая статистика	
Создание новой конфигурации	

Введение

В настоящее время информационные технологии широко используются в деятельности практически всех организаций разного уровня. Это приводит к повышению потребности в специалистах, которые профессионально владеют разнообразными современными компьютерными технологиями. При этом среди наиболее частых требований и пожеланий со стороны работодателей — навыки работы с программным продуктом 1С:Предприятие. Эта информационная система является самой популярной на территории нашей страны программой автоматизации различных участков офисной работы. Дело в том, что система 1С:Предприятие позволяет существенно повысить эффективность экономической и организационной деятельности предприятий.

Если посмотреть на спектр специалистов, работающих с рассматриваемым программным продуктом, то мы увидим, что он достаточно широк — менеджеры, бухгалтеры, программисты, разработчики, руководители подразделений и т. д. Всех их можно разделить на две условные категории — разработчиков и пользователей.

Первые создают прикладные разработки для дальнейшего использования пользователями в конкретной области деятельности (бухгалтерия, торговля, склад, управление персоналом и др.). Для этого в систему 1С:Предприятие заложены разнообразные ресурсы, основным из которых является *встроенный программный язык*. И построение любого прикладного решения на платформе 1С:Предприятие связано с написанием программного кода.

Другая, бо́льшая по численности категория специалистов, связанных с рассматриваемым программным продуктом, — пользователи. Это все те, кто работает с готовыми разработками (с тем, что уже создано разработчиками). И часто у пользователя, имеющего заметный опыт работы с одним из прикладных решений на платформе 1С:Предприятие 8, создается иллюзия, что он уже овладел всеми секретами мастерства. Однако для внесения даже небольших изменений (которые реально часто требуются в связи с тем, что в работе каждой организации есть свои специфические особенности) в готовое решение необходимы другие знания и практические навыки — представление об объектах, составляющих систему 1С:Предприятие, знание встроенного языка программирования, опыт разработчика и многое другое. Перед пользователями, овладевшими этими знаниями, открываются широкие возможности, как для внесения изменений в готовые решения, так и для создания собственных прикладных разработок. Именно для читателей, которые на практических примерах хотят быстрее овладеть секретами мастерства разработчика, написана эта книга.

Несколько лет назад фирмой "1С" была выпущена очередная версия программы — 1С:Предприятие 8.1. Один из первых вопросов, который сразу возникает что необходимо предварительно знать для успешного усвоения информации? Практические занятия с различными категориями учащихся показали, что освоение приведенных в книге разработок вполне по силам и тем, кто не имеет навыков в области программирования, и тем, кто до этого не сталкивался с системой 1С:Предприятие. Приведенные в книге примеры использовались в реальном учебном процессе. На этих занятиях присутствовали учащиеся как различных возрастных категорий, так и с различным уровнем начальной подготовки.

Вообще, практика показала, что приобрести навыки для самостоятельных разработок на платформе 1С:Предприятие 8.1 можно за относительно небольшой срок.

Читателям четвертого издания книги

Первое издание данной книги вышло под названием "1С:Предприятие 8. Учимся программировать на примерах". Оно имело много положительных отзывов со стороны читателей. И многие из них смогли научиться разрабатывать собственные прикладные решения для платформы 1С:Предприятие 8.

Выход новой версии программного продукта (1С:Предприятие 8.1) послужил поводом для выпуска второго издания книги. Она вышла под тем же названием, что и книга, которую вы держите в руках, и в нее было включено большинство примеров из первого издания, а также добавлены разделы, касающиеся новизны информационной платформы 1С:Предприятие 8.1.

В третье издание также был внесен ряд дополнений. Так появилась новая глава, касающаяся формирования статистических отчетов по имеющейся информации в базе данных. Дело в том, что в систему 1С:Предприятие 8.1 заложены программные объекты, которые позволяют легко получать сведения о статистических показателях и о наличии статистических закономерностей в имеющихся данных. Например, можно легко определить совместно покупаемые товары или оценить закономерность в последовательных заказах, сделанных нашими партнерами. Это в итоге можно использовать для повышения качества принятия управленческих решений в организациях.

В четвертое издание книги добавлена новая глава, в которой рассматривается развитие системы 1С:Предприятие 8 в плане использования новой платформы 8.2. Приведены сведения, касающиеся преимуществ новой платформы по сравнению с предыдущими. Кроме того, *глава 8* дополнена примерами и справочными сведениями по настройке цветов при оформлении табличных данных.

Общий взгляд на систему 1С:Предприятие

Программный продукт 1С:Предприятие позволяет существенно автоматизировать деятельность как определенного подразделения, так и всей организации в целом. Основной отличительной чертой рассматриваемой системы является ее способность подстройки под конкретную область деятельности. Это достигается за счет конфигурируемости, которая заложена в данный программный продукт. Вообще, функционирование системы разделяется на два процесса — конфигурирование (работа в режиме конфигуратора) и непосредственно пользовательскую работу с тем, что создано на этапе конфигурирования.

В режиме конфигуратора создаются необходимые для решения прикладной задачи *объекты* (точнее — объекты конфигурации). При этом определяются формы ввода информации, алгоритмы обработки данных, а также необходимые пользователю отчеты. В целом результатом конфигурирования является конкретная *конфигурация*, которая отражает модель предметной области определенного подразделения или целой организации.

Сама по себе деятельность разработчика в конфигураторе может быть разделена на две составляющие. Как и в любой другой современной инструментальной среде, в системе 1С:Предприятие 8.1 большое место занимают визуальные средства разработки. Они представляют собой работу с диалоговыми окнами, размещение необходимых элементов управления и задание их свойств. В этом случае практическая сторона фактически сводится к управлению объектами с помощью мыши, а также к формированию необходимой информации в полях ввода, флажках, переключателях и т. д. Вторую же (и наиболее трудоемкую) составляющую работы в конфигураторе представляет непосредственное программирование на встроенном языке.

При создании конфигурации, направленной на решение конкретной прикладной задачи, разработчик использует такие заложенные в систему объекты конфигурации, как справочник, документ, форма, регистр сведений, регистр накопления и многие другие. На базе указанных объектов разработчик для определенной прикладной области уже создает специфические объекты, составляющие разрабатываемую конфигурацию. Например, это могут быть справочники номенклатура или сотрудники, первый из которых предназначен для информации о товарах, а второй — для информации о лицах, работающих в организации.

Аналогично, используя объект конфигурации *документ*, можно разработать прикладной документ приходная накладная.

Стоит заметить, что в конфигураторе создаются не конкретные документы, справочники либо отчеты, а определяется лишь их вид, или, можно сказать, структура (еще лучше подходит слово шаблон). Например, в режиме конфигуратора можно определить, что в создаваемом документе должны присутствовать следующие поля:

- название склада;
- название фирмы-поставщика;
- список товаров с указанием количества.

После создания подобной структуры в работу может включаться пользователь, который в режиме 1С:Предприятие вносит данные в создаваемые системой документы, имеющие уже разработанную структуру. Таким образом, для сотрудников, не знакомых с конфигуратором и технологией создания объектов конфигурации, видна только пользовательская составляющая деятельности.

Важно отметить, что система 1С:Предприятие позволяет создавать программные объекты, которые являются очень близкой моделью реальных, используемых в конкретной сфере деятельности. В этом случае пользователи, впервые столкнувшиеся с этой информационной системой, отмечают удобство при переходе от бумажной документации к электронной.

Фактически для последующей работы пользователя в режиме 1С:Предприятие разработчиком в режиме конфигуратора создается необходимая схема (алгоритм действий). Результаты конфигурирования обеспечивают сотрудникам организации комфорт при работе с информационной базой, созданной системой на основании разработанной конфигурации.

В качестве технологических средств разработки ранее отмечен встроенный язык программирования, заложенный в рассматриваемую систему. Это наиболее важный ресурс для разработчика, но не единственный. Приведем еще несколько наиболее востребованных.

В системе 1С:Предприятие присутствует большое количество конструкторов инструментов, которые позволяют разработчику избежать выполнения разнообразных рутинных действий. В частности, конструкторы предоставляют возможность автоматически создавать необходимые формы для ввода данных, печатные формы, программный код и, что важно, все это происходит в результате несложного диалога.

Также в программный продукт 1С:Предприятие заложен механизм запросов, который в основном используется для построения разнообразных отчетов, являющихся непременным атрибутом автоматизации в любой предметной области. Механизм запросов основан на еще одном языке, заложенном в систему, — языке запросов, который обеспечивает выборку интересующей информации из базы данных.

Встроенный текстовый редактор используется для создания программных модулей на встроенном языке. Для удобства разработчика в этом редакторе имеется возможность контекстного выделения цветом синтаксических конструкций. Кроме того, при работе программист может пользоваться подсказками и шаблонами.

Для создания макетов выходных форм в системе 1С:Предприятие предусмотрен редактор табличных документов. Формат табличного документа является универсальным для документов, справочников, отчетов и других объектов.

Средством, которое существенно облегчает труд разработчика, является *отлад-чик*. Он позволяет отслеживать выполнение программных модулей, замерять время выполнения и просматривать содержимое переменных.

Этот перечень составляет только часть многочисленных технологических средств, заложенных в рассматриваемую информационную систему. Для получения исчерпывающей информации следует обратиться к справочной документации, прилагаемой к программному продукту 1С:Предприятие 8.1.

Краткое содержание книги

Книга состоит из десяти глав, которые наполнены разнообразными практическими примерами. Организация материала обусловлена желанием сделать восприятие информации удобным для читателя. Необходимые справочные сведения по используемым средствам и встроенному языку программирования приводятся по ходу изложения примеров. Наилучший эффект при чтении будет достигаться при параллельном выполнении приведенных разработок на компьютере. Именно для этого все технические шаги в книге описаны достаточно подробно.

В *главе 1* читатели познакомятся с такими основными объектами конфигурации, как *перечисление, справочник, документ* и *отчет*. На практических примерах будет рассмотрена технология разработки электронных форм, являющихся обязательным атрибутом интерфейса пользователя. Первые шаги в плане программирования будут выполнены с использованием конструкторов — замечательных средств для облегчения труда разработчика. После этого самостоятельное программирование (без использования конструкторов) начнется с небольших процедур, которые будут понятны читателям, не встречавшимся до этого с системой 1С:Предприятие 8.1, либо работавшим в ней только в качестве пользователей. Вы увидите, как легко с помощью несложного программного кода можно обеспечить необходимый сервис для последующей пользовательской работы сотрудников организации.

В целом стратегическая цель главы 1 заключается в том, чтобы заложить у читателя базовые навыки по работе с конфигуратором, который представляет собой один из режимов работы системы 1С:Предприятие. Результатом труда разработчика в этом режиме является конфигурация — ее использует пользователь при работе в режиме работы 1С:Предприятие (это другой важный режим работы данной системы).

В *главе 2* рассматриваются приемы программной работы с документами. Примеры касаются создания документов программными средствами, отбора документов и заполнения одних документов на основании других.

Глава 3 отводится организации и программированию регистров накопления. Здесь рассматривается программное извлечение информации, а также организация движения по регистрам.

В *главе* 4 демонстрируются возможности нового объекта системы 1С:Предприятие 8 — регистра сведений (в очень популярной версии 7.7 он отсутствовал).

Глава 5 демонстрирует разработку на платформе 1С:Предприятие 8.1, которая встретилась в реальной практике. А именно, рассматривается создание конфигурации, связанной с решением задачи, которая возникла в ходе управления деятельностью подразделением высшего учебного заведения. Разработка касается учета оплаты за обучение. Одним из ключевых моментов *славы 5* является демонстрация возможностей программирования форм ввода. В целом рассмотренный здесь пример конфигурации показывает, как легко с помощью разнообразных объектов системы, их свойств и методов можно создать удобные пользовательские решения для автоматизации офисной работы. Рассмотренная разработка позволит читателю в дальнейшем создать собственные прикладные решения для фирмы, где он работает.

Как показала практика, ценность подобного специалиста значительно увеличивается, и очень часто он воспринимается руководством не как обыкновенный пользователь, а как вполне приличный программист.

Глава 6 целиком отведена языку запросов, который позволяет обеспечить удобный доступ к имеющимся в системе данным. Причем содержание главы не представляет собой чисто справочную информацию: для каждой синтаксической конструкции языка запросов приводятся примеры, результаты, выполнения которых продемонстрированы иллюстрациями, отражающими работу в режиме 1С:Предприятие. Заметим, что в этой главе не разрабатывается новая информационная база, а все запросы используют ранее созданные разработки из предыдущих глав книги.

В *главе* 7 рассматривается *система компоновки данных*. Показано, каким образом без знания программного кода можно эффективно создавать разнообразные отчеты. Система компоновки данных — это новое средство, предназначенное для создания отчетов на основе декларативного описания. Механизм компоновки данных позволяет дать целостное описание отчета, используя только визуальные средства.

В главе 8 демонстрируются разнообразные возможности, заложенные в уже встречавшиеся до этого в тексте книги объекты конфигурации. Приводится пример создания конфигурации по учету денежных вознаграждений сотрудникам организации с учетом изменяющихся с течением времени различных тарифов. В настоящее время большое внимание уделяется интеграции различных приложений, что позволяет существенно автоматизировать работу офиса в целом и избежать многих неэффективных действий. В этом плане интересными представляются примеры организации взаимодействия системы 1С:Предприятие с такими офисными приложениями, как Microsoft Excel и Microsoft Word. Такие примеры приводятся в главах 5 и 8. В четвертое издание книги включен материал по настройке цветов при оформлении табличных данных.

В *главе* 9 рассматриваются ресурсы системы 1С:Предприятие 8.1, касающиеся формирования статистической информации. Практические примеры показывают, каким образом можно с минимальными усилиями рассчитывать статистические показатели и находить статистические закономерности в исходных данных.

Глава 10 появилась в четвертом издании книги. В ней рассматривается развитие системы 1С:Предприятие 8 в плане использования новой платформы 8.2. Приведены сведения, касающиеся преимуществ новой платформы по сравнению с предыдущими. Скоро в издательстве "БХВ-Петербург" планируется выпуск моей книги по программированию и разработке прикладных решений для новой платформы 1С:Предприятие 8.2.

Компакт-диск

Прилагающийся к книге компакт-диск содержит все разработки, приведенные в книге. Основная составляющая диска — это конфигурации, описание разработки которых и составляет содержательную часть издания. Кроме того, как уже говорилось ранее, часть рассмотренных примеров в книге связана с интеграцией работы

системы 1С:Предприятие 8.1 и популярных офисных приложений. Упоминаемые в издании автоматизированные книги Microsoft Excel и документы Microsoft Word также содержатся на компакт-диске. Все файлы распределены по тематическим папкам, а подробная инструкция работы с ними приводится в *приложении*.

От автора

Во-первых, мне бы хотелось выразить благодарность всем читателям, которые познакомились с моей книгой. По тематике, связанной с офисным программированием, она у меня не первая [3—6]. Первые три издания данной книги были отмечены положительными отзывами со стороны читателей. Фактически это явилось для меня поводом подготовки четвертого издания, которое, наряду с примерами из первых изданий, включает и ряд новых разделов.

При подготовке книги большую помощь мне оказали супруга и дочь. Это касается подготовки многочисленных представленных примеров, а также графических иллюстраций.

Большую благодарность хочу выразить издательству "БХВ-Петербург" за сотрудничество в плане подготовки к изданию моих книг на протяжении последних лет. Особенно хочу поблагодарить Рыбакова Евгения Евгеньевича за ценные советы и большое внимание. Также выражаю благодарность редактору первых изданий книги Якубовичу Юрию Владимировичу за полезные замечания и советы, которые я учел в этом (четвертом) издании книги.

Что касается тематики разработок, приведенных в книге, то все они связаны с Институтом дистанционного обучения, который входит в состав Нижегородского государственного лингвистического университета им. Н. А. Добролюбова. Директору Института дистанционного обучения Александру Николаевичу Сверчкову и заместителю директора Юрию Васильевичу Воронкову я хочу выразить большую благодарность. Фактически значительная часть материала появилась как результат их эффективного управления организацией учебного процесса. Поставленные ими задачи фактически явились фундаментом данного издания.

В заключение еще раз замечу, что наибольший эффект будет наблюдаться в том случае, если вы параллельно с чтением будете выполнять практические примеры непосредственно в системе 1С:Предприятие 8.1.

Я боюсь, что, несмотря на все старания, книга не лишена неточностей. Все их я отношу исключительно на свой счет и заранее приношу читателям за них свои извинения.

Связаться с автором книги можно по электронной почте mail@bhv.ru. Также вы можете обратиться по адресу www.bhv.ru, чтобы выразить свое отношение к книге.

глава 1



Представление об объектах системы 1С:Предприятие

В настоящее время популярность программного продукта 1С:Предприятие 8.1 стремительно растет и, соответственно, увеличивается потребность в специалистах, знакомых с этой информационной системой. Часто уровень квалификации в данной области является определяющим при подборе кандидатуры на то или иное вакантное место. При этом в большинстве ситуаций от потенциального сотрудника требуются, в основном, навыки уверенного пользователя при работе с одной из наиболее популярных на сегодняшний день конфигураций системы 1С:Предприятие 8.1. В этом случае специалист занимается профессиональной деятельностью в своей предметной области (например, бухгалтерия или сфера складского учета) и использует в работе современные технологии автоматизации. А большинство сложных задач, связанных с внесением изменений в имеющуюся конфигурацию (например, добавление нового документа или отчета), решает, как правило, системный программист.

Распространена и другая ситуация. Так, в небольших организациях пользователям и программистам (которые не сталкивались до этого с программными продуктами фирмы "1С") часто приходится самостоятельно решать многие практические вопросы, связанные с внесением изменений в имеющуюся конфигурацию. Именно для такой категории читателей мы и попробуем на примерах разобрать основные приемы, касающиеся программирования в среде 1С:Предприятие 8.1.

Будет правильно, если мы начнем с рассмотрения достаточно простых ситуаций, которые должны быть понятны читателям, только начинающим знакомиться со средствами разработки, имеющимися в программных продуктах фирмы "1С". Разумеется, сразу возникает вопрос: а что нужно предварительно знать для успешного восприятия изложенного в книге материала? Конечно, желательны навыки в области программирования, а также приветствуется любой опыт работы с информационной системой 1С:Предприятие. Все это придаст читателю больше уверенности при работе с книгой, увеличит скорость рассмотрения примеров и позволит ему быстрее перейти к решению собственных задач. Однако можно обойтись и без этого.

Главное, что необходимо, — это наличие на своем компьютере установленной программы 1С:Предприятие 8.1. Для той категории читателей, которая не располагает данной программой, хочу рекомендовать учебную версию 1С:Предприятие 8.1. Она существенно дешевле рабочего варианта, но при этом поддерживает практически все функциональные возможности.

При наличии программного продукта оптимальная технология работы с книгой на начальном этапе заключается в параллельном выполнении рассматриваемых разработок на своем компьютере. Понятно, что объем справочной информации по программированию в среде 1С:Предприятие 8.1, просто огромен, и уже в середине этой главы нам потребуется познакомиться с рядом справочных сведений. По ходу всей книги я буду приводить подробную информацию по мере необходимости, что весьма удобно для читателей. Поэтому рассмотрение примеров не предполагает использования дополнительных источников информации — в ходе описания примеров необходимые справочные данные будут представлены.

Для работы нам не потребуется наличие стандартных конфигураций, разработанных для системы 1С:Предприятие 8.1 — мы будем создавать свои собственные и на их примере рассматривать различные технические приемы, необходимые для выполнения прикладных разработок. Конечно, эти конфигурации не будут такими сложными, как разработки специалистов фирмы "1С", однако итогом рассмотрения примеров для читателей станут навыки, которые они смогут использовать в практической работе с рассматриваемой системой. Другие читатели смогут после рассмотрения приводимых конфигураций создать и собственные разработки, которые автоматизируют деятельность организаций, где они работают.

Архитектура системы 1С:Предприятие

В системе 1С:Предприятие можно выделить две основные составляющие:

- технологическую платформу;
- прикладные решения автоматизации различных участков деятельности, которые создаются с помощью технологической платформы.

Подобная организация системы обеспечивает достаточную гибкость прикладных решений. В этом случае программист, даже не участвовавший в создании конкретной разработки, располагает всеми необходимыми ресурсами для внесения в нее изменений. Следует заметить, что в настоящее время имеется большое количество стандартных и узкоспециализированных конфигураций, которые постоянно корректируются и совершенствуются. Такая открытость системы для модификации делает ее очень удобной для программистов, и они могут легко дорабатывать и развивать существующие типовые разработки.

Ничто не мешает каждой организации, располагающей небольшим составом программистов (даже одним), настроить большинство стандартных прикладных решений под себя. В этом случае не требуется использовать какие-либо отдельные программные продукты — все средства разработки входят в состав технологической платформы.

В технологической платформе выделяются две составляющие:

- среда исполнения;
- среда разработки.

Среда исполнения называется режимом работы 1С:Предприятие, а среда разработки открывается пользователю при запуске системы в режиме конфигуратора.

В среде разработки используется технология метаданных, которые представляют собой структуру объектов, описывающую конкретное прикладное решение. Среда исполнения выполняет (проигрывает) метаданные. Здесь можно провести некую аналогию с любой системой, выполняющей программный код.

При работе с метаданными доступно широкое использование визуальных средств разработки. В этом случае программисту не требуется писать программный код для добавления нового объекта метаданных. Все выполняется с помощью щелчков мыши и установки необходимых параметров в списках, переключателях и других элементах управления. Результат же этих действий автоматически трансформируется системой в программный код.

Работа программиста в среде разработки приводит к построению конкретной конфигурации. И такая конфигурация реализуется (проигрывается) в среде выполнения (в режиме 1С:Предприятие). Важно заметить, что платформа 1С:Предприятие содержит фиксированный набор прототипов (шаблонов) объектов конфигурации. Например, имеются: шаблоны справочника, документа, регистра сведений, отчета и еще ряд других. Когда в среде разработки создается новый объект конфигурации, то он наследует используемый при его построении шаблон.

Таким образом, несмотря на то, что каждая прикладная конфигурация обладает собственной индивидуальностью, она создана на основании базовых объектов конфигурации.

Встроенный язык программирования 1С:Предприятие, а также язык запросов используются для описания специфических алгоритмов прикладного решения. Встроенный язык 1С:Предприятие имеет много общих черт с другими языками программирования, такими как Visual Basic, JavaScript и др. Для построения запросов в систему 1С:Предприятие включен еще один язык — язык запросов, который похож на известный (по работе с базами данных) язык построения запросов SQL. Важной особенностью языка запросов в системе 1С:Предприятие является то, что он представляет собой доступ к данным только на чтение. Для записи же информации используются различные методы объектов системы 1С:Предприятие.

При создании прикладных решений применяются как визуальные средства разработки, так и программирование на встроенном языке. Любое новое прикладное решение — это совокупность метаданных. Однако когда вы начинаете новую разработку, то начинаете ее не с нуля. Платформа при создании новой конфигурации автоматически создает некую базовую структуру метаданных. Она уже сама по себе представляет собой разработку, которая может быть выполнена на платформе 1С. Конечно, какой-то практической пользы от этого нет, т. к. необходимой функциональности в нее еще не заложено. Задача разработчика как раз и заключается в том, чтобы эту функциональность создать путем доработки структуры метаданных. В дальнейшем после такой доработки при переходе в пользовательский режим работы система увидит эти изменения в структуре метаданных и реализует необходимую функциональность. Набор объектов метаданных ограничен (справочник, документ, регистр накопления и еще ряд других) и разработчик не может создать свой, принципиально новый, объект конфигурации. Он может только добавлять в разрабатываемое прикладное решение новый объект конфигурации, который базируется на одном из прототипов, поддерживаемых системой.

Разумеется, практические задачи требуют разработки различных алгоритмов. Эти алгоритмы специфичны для каждой решаемой задачи и для их разработки предназначен встроенный язык системы 1С:Предприятие. Для размещения текстов программ на встроенном языке предназначены модули (далее мы встретимся с модулями форм, модулями объектов и общими модулями, но в системе есть и другие). Большинство программных модулей вызываются в определенные моменты работы прикладного решения.

Взгляд на встроенный язык программирования

В этом разделе мы дадим первые пояснения по встроенному языку программирования, который будет использоваться для прикладных разработок на протяжении всех глав книги. Встроенный язык представляет собой предметно-ориентированный язык программирования, специально разработанный с учетом возможности его применения не только профессиональными программистами. Данный язык является двуязычным — все операторы языка имеют как русское, так и англоязычное написание.

В текстах программных процедур и функций операторы между собой обязательно следует разделять точкой с запятой (;). Конец строки не является признаком конца оператора, и в строке можно располагать произвольное число операторов, разделяя их точкой с запятой.

Именем переменной (процедуры или функции) может быть любая последовательность букв, цифр и знаков подчеркивания. Важно, что распознавание имен переменных ведется без учета регистров букв. Начинаться имя должно с буквы или знака подчеркивания. Назначаемые пользователем имена не должны совпадать с зарезервированными словами встроенного языка.

Отметим некоторые специальные символы, которые можно использовать в программных модулях и которые встретятся далее в книге:

- ♦ // комментарий, который располагается после двух наклонных черт;
- нертикальная черта в начале строки используется только в строковых константах и означает, что данная строка является продолжением предыдущей (организуется перенос строки);
- в круглые скобки заключается список параметров методов, процедур, функций и конструкторов (также они используются в выражениях встроенного языка);
- "" в двойные кавычки заключаются строковые литералы;
- ♦ '' одинарные кавычки используются для литералов даты.

Во встроенном языке системы 1С:Предприятие поддерживается набор примитивных типов данных. Наиболее часто в разработках используется тип Строка. Значения данного типа содержат строку символов в формате Unicode произвольной длины. Литералы строкового типа представляют собой набор символов, заключенных в кавычки. Если же необходимо задать в строке непосредственно символ " (кавычка), то следует записать две кавычки подряд. Можно задавать и многострочные строковые константы. Для этого удобнее всего использовать символ продолжения строки (+). Подобные константы нам встретятся при рассмотрении запросов.

Тип данных число позволяет представить в процедурах любое десятичное число. Над данными числового типа определены основные арифметические операции: сложение, вычитание, умножение и деление.

Null представляет собой тип данных, который используется исключительно для определения отсутствующего значения.

Значения типа Булево могут быть двух видов — истина и ложь. В программных конструкциях значения данного типа возвращаются в качестве результата вычисления логических выражений.

Значения типа дата содержат дату от Рождества Христова (с 1 января 0001 года) и время с точностью до секунды.

Рассмотрим теперь ряд наиболее часто используемых операторов и синтаксических конструкций. Оператор цикла для предназначен для циклического повторения других операторов. Синтаксис этого оператора цикла выглядит следующим образом:

Для <Имя переменной> = <Выражение 1> По <Выражение 2> Цикл // Операторы

КонецЦикла;

Данный оператор позволяет циклически повторять операторы, находящиеся внутри конструкции цикл...конеццикла. Перед началом выполнения цикла значение «Выражение 1> присваивается переменной с именем «Имя переменной». Значение «Имя переменной» автоматически увеличивается при каждом проходе цикла. Величина приращения счетчика при каждом выполнении цикла равна 1. Цикл выполняется, пока значение переменной «Имя переменной» остается меньшим или равным значению «Выражение 2». Условие выполнения цикла всегда проверяется в начале, перед выполнением цикла.

Оператор цикла для каждого предназначен для циклического обхода коллекций значений. При каждой итерации цикла возвращается очередной элемент коллекции. Обход осуществляется до тех пор, пока не будут перебраны все элементы коллекции. Синтаксис данного оператора выглядит следующим образом:

```
Для каждого <Имя переменной 1> Из <Имя переменной 2> Цикл
// Операторы
КонецЦикла;
```

Здесь «Имя переменной 1» — переменная, которой при каждом повторении цикла присваивается значение очередного элемента коллекции. Другая переменная («Имя переменной 2») представляет саму коллекцию. Элементам этой коллекции бу-

дут периодически присваиваться переменной <имя переменной 1>. Операторы, следующие за ключевым словом цикл, выполняются до тех пор, пока не будут перебраны все элементы коллекции. Существует возможность завершить любой цикл досрочно. Для этого следует воспользоваться оператором прервать.

Оператор Если управляет выполнением программы, основываясь на результате одного или более логических выражений. Этот оператор может содержать любое количество групп, начинаемых с конструкции иначеЕсли. Синтаксис рассматриваемого оператора выглядит так:

```
Если <Логическое выражение> Тогда

// Операторы

ИначеЕсли <Логическое выражение> Тогда

// Операторы

Иначе

// Операторы

КонецЕсли;
```

Операторы, следующие за первым ключевым словом тогда, выполняются, если результатом логического выражения, расположенного после ключевого слова Если, является истина. Логическое выражение, следующее за ключевым словом иначеЕсли, вычисляется только тогда, когда условия, указанные в Если и всех предыдущих иначеЕсли, оказались равны ложь. Операторы, следующие за конструкцией иначеЕсли...Тогда, выполняются, если результат логического выражения в данном иначе-Если равен истина. Операторы, следующие за ключевым словом иначе, выполняются, если результаты логических выражений в конструкции Если и всех предшествующих конструкциях иначеЕсли оказались равны ложь.

Оператор новый позволяет создать значение указанного типа. Он допустим только при использовании тех типов, для которых разрешено создание новых значений. Синтаксис этого оператора выглядит так:

Новый <Имя типа>

Оператор цикла пока предназначен для циклического повторения операторов, находящихся внутри конструкции цикл...Конеццикла. Цикл выполняется, пока логическое выражение равно истина. Условие выполнения цикла проверяется в начале, перед выполнением цикла. Синтаксис оператора выглядит так:

Пока Логическое выражение Цикл // Операторы КонецЦикла

Операторы, следующие за ключевым словом цикл, выполняются, пока результат логического выражения равен истина.

Таким образом, в этой вводной части главы сформировано начальное представление о конструкциях языка. Дальнейшие сведения на данную тему будут приводиться по ходу рассмотрения примеров книги.

Постановка задачи

В этой главе мы создадим несколько прикладных объектов конфигурации для решения несложной практической задачи, после чего поработаем в качестве пользователей с информационной базой, созданной системой на основании разработанной конфигурации. Все технические моменты, связанные с работой в конфигураторе, будут подробно комментироваться, и, таким образом, читатель легко сможет повторить описанные действия на своем компьютере. При желании можно обратиться к компакт-диску, прилагаемому к книге, где приведена разработанная в главе конфигурация (как, впрочем, и все другие конфигурации, описываемые в книге).

Условимся считать, что наша организация предлагает платные услуги, связанные с учебно-преподавательской деятельностью. Это может быть одно из подразделений в высшем учебном заведении или просто учебный центр, занимающийся коммерческой деятельностью. Мы, как разработчики, попробуем помочь нашим сотрудникам в автоматизации некоторых участков работы с учащимися (слушателями курсов), для обозначения которых будем использовать более общее понятие — "клиенты". Каждый клиент может записаться на тот или иной платный курс, после чего внести плату за обучение (сразу либо частично), а затем начать посещать занятия. Понятно, что руководство учебного центра должно максимально эффективно контролировать данный процесс.

В целом наша задача выглядит так — используя возможности платформы 1С:Предприятие 8.1 построить информационную систему для внесения, хранения и извлечения информации, которая позволит автоматизировать работу персонала учебного центра. Для этого нам потребуется создать объекты конфигурации, которые будут отражать реальную ситуацию в организации учебного процесса. Фактически мы создадим модель предметной области, которую затем реализуем на практике, используя возможности системы 1С:Предприятие 8.1.

Все действия будут выполняться в конфигураторе, который, как уже говорилось, является одним из режимов работы информационной системы 1С:Предприятие. По сути, в конфигураторе и были созданы все известные прикладные разработки фирмы "1С".

Окно запуска

Итак, после выбора раздела 1С Предприятие 8.1 в меню Пуск и далее пункта Конфигуратор (либо 1С Предприятие) перед нами открывается стартовое окно информационной системы 1С:Предприятие 8.1 (рис. 1.1).

В центральной части окна перечислены уже имеющиеся информационные базы, каждая из которых представляет собой конкретную разработку, направленную на автоматизацию участка работы либо определенного подразделения, либо всей организации в целом. В нашей ситуации мы начинаем решение новой задачи, поэтому от нас потребуется создание новой информационной базы. Эта база будет построена на основе разрабатываемых далее объектов конфигурации.



Рис. 1.1. Стартовое окно информационной системы 1С:Предприятие 8.1

Примечание

В системе 1С:Предприятие возможны два режима работы — режим конфигуратора и режим 1С:Предприятие. *Режим 1С:Предприятие* для пользователей является основным, и в нем пользователи вводят информацию в систему, получают необходимые отчеты и т. д. *Режим конфигуратора* используется разработчиками и программистами. В этом режиме производится создание новых прикладных решений и изменение уже имеющихся.

Создание новой информационной базы

Технические действия для создания новой информационной базы достаточно просты. Сначала следует щелкнуть по кнопке Добавить (см. рис. 1.1), что приведет к появлению на экране следующего окна (рис. 1.2), в котором необходимо выбрать один из двух сценариев дальнейшего развития событий. Мы пойдем по пути разработки новой информационной базы и, нажав кнопку Далее, перейдем к окну, в котором следует задать способ создания будущей разработки (рис. 1.3). Здесь перед нами открывается путь создания новой информационной базы на основе одного из имеющихся шаблонов, однако мы поступим по-другому и создадим пустую информационную базу.

Очередное окно данного диалога показано на рис. 1.4. Здесь необходимо дать базе имя, и в нашем случае вместо варианта, предложенного системой по умолчанию, лучше установить другое название, например глава 1. Если этот вариант не нравится, то можно выбрать любое другое словосочетание. Кроме того, от нас потребуется указать каталог, где созданная разработка будет располагаться. Для этого необходимо создать пустую папку в любом месте на компьютере либо в сети, а в диалоговом окне указать ее месторасположение. В этой папке и будет храниться наша конфигурация, которую мы разработаем в *данной главе*.

После выбора названия информационной базы на экране повторно появляется окно, которое мы уже видели (см. рис. 1.1), но уже с новой строкой, содержащей

Добавление информационной базы/группы	×
 Добавление информационной базы в список: ○ Создание новой информационной базы Cоздание информационной базы ило поставляемой конфигурации, поставляемой демонстрационной базы или создание пустой информационной базы без конфигурации Добавление в список существующей информационной базы Включение в список существующей информационной базы, расположенной на данном компьютере, в локальной сети или на сервере 1С:Предприятия 	
< Назад Далее > Отмена	

Рис. 1.2. Окно для добавления новой (существующей) информационной базы

Добавление информационной базы/группы	×
 Создание информационной базы из шаблона Выберите поставляемую конфигурацию для начала работы или демонстрационный пример для ознакомления 	
 ПС:Бухгалтерия предприятия ПС:Зарплата и Управление Персоналом Профессиональная разработка в системе 1С:Предприятие Профессиональная розработка в системе 1С:Предприятие 	
 Создание информационной базы без конфигурации для разработки ново конфигурации или загрузки выгруженной ранее информационной базы 	ЭЙ
< Назад Далее > Отмена	

Рис. 1.3. Окно выбора способа создания информационной базы



Рис. 1.4. Окно для указания наименования создаваемой информационной базы

название только что созданной информационной базы. Теперь с помощью кнопки **Конфигуратор** следует перейти в режим разработки объектов конфигурации, на основе которых системой будет построена база данных.

Итак, все предварительные шаги выполнены, и мы можем перейти к основной работе по созданию конфигурации. В дальнейшем (при последующих сеансах работы на компьютере), когда вы решите продолжить созданную разработку, от вас потребуется лишь выбрать название информационной базы в списке (см. рис. 1.1) и нажать кнопку Конфигуратор или 1С:Предприятие. Вторая кнопка вызывает режим, предназначенный для непосредственной работы по внесению данных в информационную базу.

В настоящий момент мы остановились на том, что с помощью кнопки **Конфи**гуратор открыли окно режима с аналогичным названием (рис. 1.5). Это и есть инструментальная среда, в которой мы будем создавать объекты конфигурации. Все прикладные решения для автоматизации различных участков деятельности созданы разработчиками именно с использованием конфигуратора.

Следующее, что от нас потребуется, — это выбор в меню **Конфигурация** пункта **Открыть конфигурацию**. В результате на экране появится окно с *деревом объектов конфигурации*, которое показано в левой части рис. 1.5. Сейчас в нем присутствуют только заголовки, а в дальнейшем создаваемые нами объекты конфигурации будут автоматически отображаться в этом дереве.



Рис. 1.5. Окно системы 1С:Предприятие 8.1 в режиме конфигуратора

Примечание

Каждая ветвь древовидной структуры описывает определенную тематическую группу конфигурации. Например, при создании новых *справочников* они будут группироваться в ветви **Справочники**, а в ветви **Документы** найдут свое место объекты конфигурации типа *документ*. Технология работы с деревом конфигурации аналогична работе с подобными древовидными представлениями в операционной системе Microsoft Windows. Для открытия той или иной ветви (например, для обзора имеющихся в конфигурации справочников) достаточно щелкнуть по обозначению соответствующего знака "плюс".

Перечисления

Среди большого количества объектов информационной системы 1С:Предприятие 8.1 наиболее простыми являются *перечисления*, представляющие собой статические списки данных, которые создаются на этапе разработки и в дальнейшем просто используются (без возможности их изменения или дополнения при работе в режиме 1С:Предприятие). Перечисления в основном используются не самостоятельно, а в совокупности с другими типами данных. Так, эти объекты применяются для ввода значений реквизитов справочников и документов. Основная цель использования перечислений — исключить неоднозначность вводимой информации.

И первым объектом, который мы создадим в конфигураторе, будет перечисление с именем Статус, отражающее статус конкретного учащегося. Дело в том, что статус не подразумевает какого-то большого разнообразия возможных значений. Кроме того, данный параметр продолжительное время не изменяется (если происходит добавление какого-то нового значения для статуса в силу изменения методики работы нашего учебного центра, то это случается очень редко). Фактически для подобных ситуаций объект типа *перечисление* и был разработан. В нашей ситуации будем считать значениями статуса учащегося следующие:

- ♦ Зачислен;
- ♦ Отчислен;
- ♦ Сертификат.

Перейдем к техническим действиям, позволяющим реализовать сказанное на практике. В режиме конфигуратора для создания нового перечисления необходимо в дереве объектов конфигурации (см. рис. 1.5) сначала щелкнуть правой кнопкой мыши по строке **Перечисления**. В результате на экране откроется контекстное меню, состоящее из единственного пункта — Добавить, который мы и выберем, что позволит перейти к работе в окне редактирования объекта конфигурации (рис. 1.6).

Примечание

Для выполнения определенного технического действия в системе 1С:Предприятие, как правило, существует несколько способов. Это весьма характерно для современных программных продуктов. В документации, входящей в комплект программного продукта 1С:Предприятие, а также в издании [7] указываются различные способы для достижения целей, которые возникают в процессе разработок прикладных решений. Здесь же (и далее в книге) для выполнения определенного действия мы будем ограничиваться одним из способов (который, по мнению автора, наиболее удобен).

Окно редактирования объекта конфигурации содержит несколько тематических закладок, что обеспечивает удобство работы разработчика. Сейчас от нас требуется на закладке Основные внести слово Статус в поле Имя. Поля Синоним и Комментарий мы заполнять не будем, а назначение этих полей поясняется в следующем примечании.

Примечание

В системе 1С:Предприятие для редактирования свойств основных объектов конфигурации (а также для настройки взаимодействия между объектами) предназначено окно

редактирования объекта конфигурации. Редактируемые свойства располагаются на нескольких закладках, и каждая закладка предназначена для настройки свойств конкретной категории. Сами закладки (их название и содержание) меняются в зависимости от типа объекта конфигурации (так, закладки, используемые для редактирования перечислений и, скажем, справочников, отличаются). Все основные объекты конфигурации, с которыми мы встретимся в книге, содержат закладку Основные. Ключевое поле здесь — Имя, значение которого представляет собой идентификатор объекта конфигурации. Поле Синоним используется в различных ситуациях в режиме работы 1С:Предприятие как название объекта, понятное для пользователя. В частности, поле Синоним используется в качестве названия раздела меню, связанного с открытием конкретного справочника, документа, отчета и др. Дело в том, что на имя любого объекта накладывается ряд ограничений, например, в нем не должно быть пробелов. В этом случае при использовании сочетаний из нескольких слов имя объекта (например, документ ПриходнаяНакладная) смотрится непривычно для пользователей. И для того чтобы при работе в режиме 1С:Предприятие объект ассоциировался с названием в более традиционной форме, следует использовать раздел Синоним для замены имени в меню. На синоним подобных ограничений не накладывается. Еще одно поле Комментарий носит вспомогательный характер и представляет собой произвольную строку. Смысл этого поля заключается в расшифровке и пояснении предназначения данного объекта. В последующих примерах каким-то образом заполнять поле комментария мы не будем. Однако в сложных разработках это имеет смысл делать, поскольку даже автор конфигурации может через какое-то время забыть собственную логику, используемую при разработке.

📩 Конфигуратор (учебная вер	осия) - Конфигурац	ия		
🛯 <u>Ф</u> айл <u>П</u> равка Конфигурация	Отладка Администр	ирование <u>С</u> ервис	<u>О</u> кна Сп <u>р</u> авка	
I 🗋 🔟 🖬 🐰 🖪 🚳 🧉	🖻 5 C 🌆	Ph	• A A 5	🦹 🚧 🌷
i 🔄 📭 🕋 🔝 🗸				
Конфигурация 🛛 🕂 🗙	🔛 Перечисление	Статус		_ 🗆 ×
Действия - Заказана - Спарти	 Основные Данные Формы Макеты Подсистемы 	Имя: Синоним: Комментарий: Назад Дале	Статус Статус 	Травка
Перечисление Статус	- [1			
для получения подсказки нажмит	eri		LAP NUM	10 V

Рис. 1.6. Окно редактирования перечисления Статус

Таким образом, несложные действия привели к тому, что мы легко создали первый объект нашей конфигурации, который, правда, пока пуст. Для того чтобы внести в перечисление статус возможные значения, следует перейти на закладку Данные и добавить необходимую информацию (рис. 1.7). Для внесения значений следует воспользоваться пиктографическим меню в верхней части окна редактирования объекта конфигурации. С помощью этих кнопок легко не только добавлять, но и корректировать и удалять элементы. Вернемся к перечислениям. У читателя наверняка возникает вопрос: как перечисления, и в частности перечисление *статус*, использовать? В этой главе мы создадим еще один объект — справочник клиенты, в котором введем реквизит с именем *статус*. Он будет принимать варианты только из набора значений перечисления *статус*, что позволит пользователю в этом случае избежать ввода других данных, кроме трех уже указанных.

Примечание

В практической работе, кроме исключения ошибок, использование заранее определенных списков данных заметно ускоряет работу пользователя в режиме 1С:Предприятие. Наряду с перечислениями для этого часто используются еще и справочники. Заметим, что справочники — это существенно более сложные (в плане программирования и функциональных возможностей) объекты по сравнению с перечислениями, и в следующем разделе мы с ними познакомимся.



Рис. 1.7. Внесение значений перечисления Статус

Справочники

Упрощенно справочник можно считать совокупностью строк, причем в каждой строке присутствует несколько колонок (количество колонок одинаково для каждой строки). Строки есть не что иное, как имеющиеся в справочнике *записи* (или, точнее говоря, элементы справочника), а колонки определяют структуру этих записей. В отличие от перечислений, в справочники пользователь при работе в режиме 1С:Предприятие может вносить информацию. Можно добавлять новые элементы справочника, удалять ненужные и редактировать существующие.

Как уже говорилось, справочники используются в ситуациях, когда необходимо исключить неоднозначный ввод информации. Аналогом справочника является спи-

сок данных, который может корректироваться и дополняться в процессе работы. Например, это могут быть списки номенклатуры, организаций или сотрудников.

Система 1С:Предприятие 8.1 позволяет создавать практически любое количество справочников. При этом в конфигураторе формируется структура любого справочника, а данными он заполняется уже в режиме 1С:Предприятие.

Примечание

Справочники в системе 1С:Предприятие могут быть иерархическими. Так, в случае наиболее часто используемой *иерархии групп* все элементы справочника делятся на два вида — простые элементы и группы. Например, в справочнике товары могут существовать группы Продукты, Одежда, Обувь и др. А внутри каждой из этих групп уже располагаются товары с указанием их стоимости и прочих свойств. Другой вариант иерархии — *иерархия элементов*. В этом случае роль групп играют сами элементы.

Справочник Должности

Продолжим практическую работу в конфигураторе созданием простого справочника должности, в котором будут содержаться названия должностей специалистов (системный администратор, преподаватель и др.). Вместо неформального слова "колонки" (отражающего представление справочника в виде таблицы) в системе 1С:Предприятие используется более общий термин — *реквизиты* (колонок столько, сколько реквизитов у справочника). При этом в любом справочнике всегда есть два обязательных реквизита (присутствующих в любом справочнике независимо от наших действий), которыми являются код и наименование. Другими словами, в каждой строке справочника всегда есть поле кода и поле наименования.

Начальные технические действия при разработке справочников аналогичны тем, которые мы рассматривали ранее при создании перечислений. Для того чтобы создать новый справочник, необходимо щелкнуть правой кнопкой мыши на пункте Справочники в дереве объектов конфигурации (см. рис. 1.5) и в появившемся контекстном меню выбрать пункт Добавить. В результате на экране откроется окно редактирования объекта конфигурации (рис. 1.8). Здесь на закладке Основные следует ввести имя, которое мы собираемся дать этому справочнику, — Должности.

Таким образом, не выполнив особо сложных действий, мы создали новый объект конфигурации — справочник должности, с которым уже можно работать, поскольку никаких дополнительных реквизитов в справочнике должностей не требуется. Следующий новый технический момент — заполнение созданного справочника в режиме работы 1С:Предприятие.

Для этого в меню Отладка (см. рис. 1.8) следует выбрать пункт Начать отладку, что позволит нам перейти в режим 1С:Предприятие или, можно сказать, в рабочую среду 1С:Предприятие (рис. 1.9). Здесь уже можно работать с информационной базой, которая построена системой на основании созданной конфигурации. В частности, в разделе Операции (меню Справочники) доступен пункт, соответствующий названию только что созданного справочника должности.

Примечание

На самом деле реальная ситуация выглядит чуть сложнее, чем просто два режима работы — режим конфигуратора и режим 1С:Предприятие. Платформа 1С:Предприя-

тие 8.1 на основе объектов конфигурации создает базу данных, включающую в себя информационные структуры, в которых будут храниться данные, см. [8]. В дальнейшем для исключения лишней путаницы в терминологии объекты конфигурации и соответствующие им информационные структуры мы часто не будем разделять.

📩 Конфигуратор (учеб	ная версия) - Кон	фигурация	
<u> Ф</u> айл <u>П</u> равка Конфигу	рация Отладка Ад	дминистрирование <u>С</u> ервис	<u>О</u> кна
Справка			
I 🗋 🛋 🖬 🐰 🖻 🕯	1 6 0 5 c	2 A A	» •
II 🗐 🔒 🚺 🚺 ,			
菌 Справочник Должно	ости		_ 🗗 🗙
• Основные	Ma and	D	
Иерархия	имя:	Должности	
Владельцы	Синоним:	Должности	
Данные	Комментарий:		
Нумерация			
Действия 👻	<Назад Дал	пее> Закрыть	Справка
😓 Перечисление Статус 📄 Справочник Должности			
Для получения подсказки	нажмите F1	CAP NUM	4 ru ▼ //

Рис. 1.8. Окно редактирования справочника Должности



Рис. 1.9. Окно информационной системы в режиме 1С:Предприятие

Теперь в режиме 1С:Предприятие заполним только что созданный справочник названиями ряда должностей (рис. 1.10). Это, с одной стороны, даст некоторый практический навык, а с другой — заполненные информацией справочники потребуются для дальнейшей работы.

При вводе данных в справочник следует обратить внимание на один важный момент. В процессе работы в режиме 1С:Предприятие пользователи просматривают и вводят информацию в справочники с помощью электронных форм (это справедливо и для других объектов). При этом, если мы в конфигураторе не совершаем никаких действий для создания формы справочника или какого-либо другого объекта, то система автоматически генерирует стандартные формы (это происходит в

момент перехода в режим 1С:Предприятие). В частности, экранная форма списка справочника, которую мы использовали на рис. 1.10, генерируется системой по умолчанию.



Таким образом, в режиме конфигуратора (в окне редактирования объекта конфигурации) для любого справочника можно ввести только имя на закладке **Основные** (см. рис. 1.8), и после этого перейти в режим 1С:Предприятие и начать работу по внесению информации в базу данных. Эта возможность позволяет существенно ускорить процесс разработки, когда не требуются включения дополнительных сервисных функций помимо тех, которые имеются в формах, генерируемых системой по умолчанию.

Примечание

Форма — это вспомогательный объект системы 1С:Предприятие, который предназначен для ввода и просмотра информации. Форма может быть эффективно использована для организации работы с данными. При открытии определенной формы справочника или документа система формирует на ней необходимую информацию. В дальнейшем при вводе данных в элементы, расположенные на форме, они сохраняются в соответствующих основных объектах (тех же справочниках, документах, регистрах и т. д.). Программный продукт 1С:Предприятие 8.1 предоставляет пользователю возможность при желании отказаться от самостоятельного создания форм, а использовать те. которые автоматически генерируются системой в режиме 1С:Предприятие. В этом случае от разработчика требуется меньше действий при создании прикладных решений.

Справочник Специалисты

После создания в какой-то степени символического справочника должности (он будет использоваться в дальнейшем для подстановки названий должностей) начнем разработку другого, более важного для рассматриваемой задачи справочника Специалисты.

Заполнив в окне редактирования объекта конфигурации традиционную для любого объекта закладку Основные, перейдем к следующей — Иерархия. В связи с тем, что наших специалистов мы собираемся классифицировать по группам (категориям или, можно сказать, по профилю специализации), то на закладке Иерархия следует отметить флажок Иерархический справочник (рис. 1.11). Это позволит в режиме 1С:Предприятие создать категории специалистов, скажем, технического и гуманитарного направления. В рассматриваемой ситуации можно ограничить число уровней иерархии двумя. Следует отметить, что в режиме 1С:Предприятие доступны удобные возможности по переносу элементов справочника из одной группы в другую.

Примечание

Количество уровней иерархии в справочнике можно ограничить, что производится с помощью установки флажка **Ограничение количества уровней иерархии** и выбора необходимого значения в поле **Количество уровней иерархии** (рис. 1.11). Если же данный флажок не установлен, то максимальное количество вложенности справочника реально не ограничено.

🛃 Конфигуратор (учеб	ная версия) - Конфигурация	
🛛 <u>Ф</u> айл <u>П</u> равка Конфигу	рация Отладка Администрирование Сервис Окна	Справка
I 🗋 🛋 🖬 🛛 🔏 🖻	a o	
菌 Справочник Специа	листы	_ 🗗 🗙
Основные • Иерархия Владельны	Иерархический справочник) Вид иерархии:	~
Данные Нумерация	Иерархия групп и элементов Размещать группы сверху	▼
Формы Макеты	Ограничение количества уровней иерархии Количество уровней иерархии	2
Подсистемы Ввод на основании		
Действия	• <Назад Далее> Закрыть	Справка
🚞 Справочник Специалист	ы	
Для получения подсказки	нажмите F1 CAP NUM	d ru▼ //

Рис. 1.11. Закладка Иерархия окна редактирования справочника Специалисты

Обратимся теперь к закладке Данные (рис. 1.12), где определяются реквизиты создаваемого справочника. Перед нами открывается управляющая форма, которая содержит много информации. В правой верхней части этой формы указаны размеры полей, отводимые для кода и наименования. Эти значения при создании нового справочника задаются по умолчанию, но при необходимости их можно изменить. Что касается длины кода, то достаточно оставить значение, предложенное по умолчанию, а для наименования (фактически для фамилии, имени и отчества специалиста) можно увеличить число символов до 50.

Примечание

Поле Код используется для идентификации элементов справочника (записей справочника). Содержание этого поля должно быть уникально для каждого элемента справочника. В принципе пользователю можно не задумываться о заполнении поля Код — система при работе в режиме 1С:Предприятие сама обеспечивает автоматическое внесение уникальных кодов при создании новых элементов в справочнике. Однако при желании можно самостоятельно заполнять поле кода, исходя из определенных соображений. Например, при использовании некоторой систематизации содержание поля кода помогает пользователю быстро идентифицировать записи (в поле кода при желании можно заложить необходимую информацию об элементах справочника). Поле Наименование отражает основную информацию об элементе справочника (например, если рассматривается справочник городов, то поле Наименование предназначено для названия города). Для него можно отвести различное число символов, однако существуют рекомендации фирмы "1С" (или, можно даже сказать, норматив) для числа символов в наименовании [7].

🛃 Конфигуратор (учеб	ная версия) - Конфигурация				
📱 <u>Ф</u> айл <u>П</u> равка Конфигу	рация Отладка Администрирование Сервис Окна Справка				
I 🗋 🛋 🖬 🛛 🔏 🛱	I 🗋 🖬 🔏 🍯 🚆 🛞 🐥 I				
🖹 Справочник Специа	листы _ 🗗 🗙				
Основные	Длина кода 5 🗘 🔿				
Иерархия					
Владельцы	— Тип кола — Основное представление — —				
Данные	Число В виде кода				
Нумерация					
Формы	О строка				
Макеты	包装准全正式				
Подсистемы					
Ввод на основании					
Права					
Интерфейсы	<u> </u>				
Обмен данными					
Прочее					
	V				
Действия - <Назад Далее> Закрыть Справка					
😸 Справочник Специалисты					
Для получения подсказки	нажмите F1 САР NUM ги т				

Рис. 1.12. Закладка Данные окна редактирования справочника Специалисты

В центре окна редактирования справочника (см. рис. 1.12) находится поле **Реквизиты**, в котором с помощью самой левой кнопки расположенного выше пиктографического меню можно добавлять дополнительные реквизиты в справочник. В рассматриваемом нами примере создадим дополнительный реквизит телефоны (рис. 1.13) для хранения информации о контактных телефонах, по которым можно связаться с преподавателем.

Для каждого реквизита должен быть определен тип данных или, иначе говоря, тип значений, которые можно заносить в реквизит в режиме 1С:Предприятие. Этот параметр (а также другие) следует установить в окне свойств, которое открывается при двойном щелчке мышью по имени реквизита. В качестве типа данных для поля телефоны можно указать — строка не более 50 символов (см. рис. 1.13), поскольку номеров телефонов может быть несколько.

🚵 Конфигуратор (учеб	ная версия) - Конфигурация			. 🗆 🗵
🛯 <u>Ф</u> айл <u>П</u> равка Конфигу	рация Отладка Администрирова	ние <u>С</u> ервис <u>О</u> н	кна Справка	
I 🗋 🛋 🖬 🐰 🖻 🦷	i 🍯 💽 🖕 🔭 📜 🎽 🕻			
🖹 Справочник Специа	листы		-	đΧ
Основные	Длина кода			ΨX
Иерархия	Плина наименования	21 🛐 🍽	X V	
Владельцы	— Тип кола — — — —	•Основные:		^
🕨 Данные	ОЧисло	Имя Т	елефоны	
Нумерация		Синоним Т	елефоны	_
Формы	• строка	Комментар		
Макеты	열 등 책 🕇 🖡 튼			_
Подсистемы	Телефоны	Использов Д	1ля элемента	-
Ввод на основании	– Рейтинг	Индексиро Н	łе индексировать	-
Права		Полнотекса 🗸	1спользовать	-
Интерфейсы	論名表演作其書	▼Тип данных	<u>.</u>	
Обмен данными	III Табличные части	Тип С	Строка	▼
Прочее		Длина	50 🗘	_
Де	йствия 🖌 🤇 Кназад 🛛 Да	Имя объекта м	1етаданных	
Для получения подсказки	нажмите F1			

Рис. 1.13. Создание реквизита Телефоны

После этого добавим еще один реквизит — Рейтинг (тип данных — число) для характеристики квалификации специалиста. Таким способом при необходимости можно добавлять разнообразные реквизиты в создаваемые справочники, делая их более информативными.

Примечание

С помощью кнопок панели инструментов, расположенной над перечнем реквизитов (они фактически являются подчиненными объектами), производится их добавление, удаление и упорядочивание.

Следует заметить, что в отличие от системы 1С:Предприятие 7.7, программные продукты 1С:Предприятие 8.0 и 8.1 позволяют создавать для справочников табличные части.

Примечание

Табличные части также являются объектами конфигурации, только относятся к кате-гории вспомогательных, а не основных.

Так, для каждого элемента справочника можно создать одну или несколько подчиненных табличных частей (таблиц). Здесь мы сразу воспользуемся этой возможностью и добавим в справочник Специалисты табличную часть с именем МестаРаботы (рис. 1.14).

Дело в том, что преподаватели (особенно высокой квалификации) часто одновременно работают в нескольких местах. При этом у одного специалиста дополнительных мест трудовой деятельности может быть три, а у другого даже пять. Табличная часть справочника для хранения информации подобного плана в этом случае идеально подходит. В противном случае мы должны были бы создавать несколько реквизитов вне табличной части, причем их количество заранее точно предусмотреть достаточно сложно.

Далее уже в табличной части МестаРаботы создадим (что выполняется с помощью второй слева кнопки пиктографического меню, относящегося к табличным частям) реквизит Организация, в качестве типа данных которого следует указать — строка длиной не более 50 символов, а также добавим реквизит должность (тип данных — СправочникСсылка.Должности). В этом случае, при работе в режиме 1С:Предприятие при заполнении поля должность будет автоматически открываться форма справочника Должности.



Рис. 1.14. Структура справочника Специалисты

Таким образом, мы разработали структуру нового объекта конфигурации — справочника специалисты, и теперь следует позаботиться о том, как пользователь

будет с ним работать. Для этого в информационной системе существуют электронные формы справочников, а инструментальные средства позволяют пользователю создавать удобные формы для работы в режиме 1С:Предприятие.

Как уже упоминалось, для ускорения процесса разработки в системе 1С:Преприятие 8.1 автоматически генерируются формы по умолчанию. Ранее при создании справочника должности мы уже воспользовались этой возможностью. Данный способ очень удобен, когда структура объекта конфигурации (справочника или какого-либо другого объекта) проста, а после его создания необходимо как можно быстрее начать работу по внесению данных в информационную базу (либо продолжить работу в конфигураторе по созданию других объектов). В дальнейшем (уже в этой главе) мы будем разрабатывать свои пользовательские формы, а сейчас еще раз воспользуемся двумя формами, генерируемыми системой по умолчанию.

Вернемся к нашей разработке и на закладке **Формы** в группе **Редактирование** сделаем установку — **Обоими способами** (рис. 1.15). Этим мы указываем системе, что в режиме 1С:Предприятие будем использовать две стандартные формы — списка справочника и элемента справочника (данная форма имеет и другое название — форма диалога). Их система 1С:Предприятие автоматически создает сама при переключении в пользовательский режим.

📩 Конфигуратор (учебная версия) - Конфигурация				
📱 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка				
I D 🔟 🖬 🔏 🚳 🖉 🙄 🕄 🖓 🐥 I				
📄 Справочник Специалисты 💶 🗗 🗙				
Основные Иерархия Владельцы Данные Нумерация ▶ Формы Макеты Подсистемы Ввод на основании Права Интерфейсы Обмен данными Прочее	Редактирование В списке В диалоге Обоими способат Звод по строке Основные формы Списка Выбора [Выбора группы Злемента Группы Сруппы	Наименован	Зыбор Из Формы Быстрый выб Обоими спос ие, Код	бор обами х х х х х х х х х х
Действия - <Назад Далее> Закрыть Справка				
и теравочник Специалисты				
Для получения подсказки нажмите F1 САР NUM ги -				

Рис. 1.15. Закладка Формы окна редактирования объекта конфигурации

В форме списка информация представлена в виде таблицы, где на каждую запись отводится одна строка (это обычный вид любого справочника). Основным достоинством подобной формы является то, что в этом случае на экране видно сразу несколько записей (элементов) справочника.

С другой стороны, недостаток формы списка обнаруживается, когда в справочнике присутствует много колонок (большое количество реквизитов), и хотелось бы, чтобы все они одновременно были видны пользователю. При практической работе в этом случае форму списка часто приходится прокручивать, чтобы увидеть необходимую информацию. Заметим, что при желании пользователь может самостоятельно настроить отображаемые колонки прямо в режиме работы 1С:Предприятие, хотя существует и другая возможность.

Эта возможность заключается в использовании еще одной стандартной формы — формы элемента, которая позволяет отображать на экране информацию только об одном элементе справочника. При этом, даже если реквизитов в справочнике много, все они видны на экране (точнее, могут быть видны, т. к. показ любого визуального элемента можно отключить).

У формы элемента имеется еще одно существенное достоинство. Если в справочнике присутствует табличная часть (как в только что созданном справочнике специалисты), то в форме списка она не отображается, а в форме элемента, наоборот, видна.

Посмотрим теперь, как воспользоваться возможностью редактирования справочника обоими способами в режиме 1С:Предприятие. Для этого перейдем в этот режим (через меню Отладка) и откроем справочник Специалисты.

Ранее при его разработке мы указали, что справочник Специалисты будет иерархическим, и сейчас создадим две группы — Техническое направление и Гуманитарное направление (рис. 1.16). Для этого воспользуемся пиктограммой Создание новой группы из пиктографического меню в верхней части окна формы справочника.



Рис. 1.16. Создание групп в справочнике Специалисты
Примечание

Для перехода в режим 1С:Предприятие из режима конфигуратора можно воспользоваться меню **Сервис**, где следует выбрать пункт **1С:Предприятие**. В процессе разработки конфигурации удобнее пользоваться пунктом **Начать отладку** меню **Отладка**.

Теперь нужно определиться, какую именно экранную форму справочника специалисты мы будем использовать при редактировании (она же будет использоваться и при создании новой строки в справочнике). Для этого у нас есть форма списка и форма элемента. Один из этих двух вариантов следует установить с помощью пункта **Редактировать в диалоге**, расположенного в меню **Действия** (см. рис. 1.16). При установке опции **Редактировать в диалоге** в случае попытки создания нового элемента (либо редактирования уже имеющегося) будет автоматически открываться форма элемента. Соответственно, если отменить режим **Редактировать в диалоге**, будет автоматически открываться форма списка.

Примечание

В реальной работе, как правило, используется большое количество разнообразных форм. Объект конфигурации *"форма"* ориентирован исключительно на пользователя — он служит для обеспечения пользователю удобного интерфейса при работе с основными объектами конфигурации.

На рис. 1.17 показан пример заполнения элемента справочника для специалиста, который входит в группу техническое направление. Видно, что в этом случае при редактировании элемента нам становится доступна для работы и табличная часть справочника.

🏪 1С:Предприя	ти <mark>е (учебная версия)</mark> - Ко	онфигурация			
<u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка					
i 🗋 📫 🖬 🕽	(🖻 🛍 🧉 🙋 🖕 🚆	. 📰 🛅 🧟	M M+	м- 🎇 🖕	
菌 Специалист	ы: Петров Иван Алексан	црович		_ 🗆 ×	
Действия 🕶 🛃	2 4				
Код:				3	
Наименование:	Петров Иван Александрови	14			
Родитель:	Техническое направление			x	
Телефоны:	12 34 56, 8-903 605 34 78				
Рейтинг:				7	
엄청동작	E GRANT I I S A E A				
N Орган	изация	Должность			
1 MTКИ		Преподавател	ь		
2 AO Ce	зязьИнвест	Программист			
3 HMC/	1	Преподавател	ь		
ОК Записать Закрыть					
🗐 Справочник Специалисты 📄 🚞: Петров Иван Александр					
Для получения по	дсказки нажмите F1		CAP	NUM //	

Рис. 1.17. Заполнение формы элемента справочника Специалисты

Примечание

Число строк табличной части не может быть более 100 000. В то же время каждый справочник может иметь практически неограниченное количество табличных частей.

На рис. 1.18 приведена форма списка рассматриваемого справочника, где в группу техническое направление внесено несколько элементов. Однако информации из табличной части в этом случае на экране не видно.



Рис. 1.18. Форма списка справочника Специалисты

Таким образом, мы познакомились с технологией разработки структуры справочников (объектов конфигурации) в режиме конфигуратора, а также узнали, каким образом заполнять справочники данными с помощью электронных форм, создаваемых системой по умолчанию. На этапе разработки и отладки прикладных решений такой вариант используется достаточно часто.

Следующая практическая задача заключается в создании справочника, который будет хранить информацию об учебных курсах, предлагаемых учащимся нашего учебного центра. В этом же справочнике в дополнительных реквизитах мы поместим информацию о стоимости курсов и их продолжительности. В функциональном плане новый справочник будет несколько сложнее предыдущего. Так, мы предусмотрим контроль корректности вводимых данных со стороны системы.

Примечание

Все разработки, относящиеся к каждой главе, объединены общей задачей, поэтому при параллельном выполнении описанных в книге действий на компьютере желательно соблюдать точность в названиях. Это связано с использованием приводимых обозначений в программных процедурах.

Справочник Курсы

Вернемся в конфигуратор и приступим к созданию еще одного справочника. Техническая работа во многом будет аналогичной ранее рассмотренным действиям,

и поэтому подчеркнем лишь существенную информацию для выполнения практических действий в конфигураторе.

Создаваемый справочник назовем курсы, а кроме обязательных реквизитов код и наименование добавим еще четыре:

- датаначала для записи даты начала занятий по данному курсу (очевидно, что в качестве типа данных следует выбрать дата);
- Продолжительность для определения продолжительности занятий в неделях (тип данных — число);
- Стоимость для указания стоимости обучения (тип данных также число);
- Преподаватель для определения специалиста, который будет эту дисциплину вести (тип данных — СправочникСсылка.Специалисты).

Примечание

При создании справочников (как, впрочем, и других объектов) системой автоматически генерируются новые типы данных. Так, когда мы создали справочник Специалисты, в системе появился новый тип данных СправочникСсылка.Специалисты. Это позволяет нам хранить в указанном выше поле Преподаватель ссылку на конкретный элемент справочника (он также является объектом) Специалисты. Подобные типы данных будут часто встречаться в книге при разработке объектов конфигурации.

Продолжим работу и теперь перейдем на закладку **Формы** (рис. 1.19), где создадим пользовательскую форму списка. С другими справочниками мы использовали формы, генерируемые системой по умолчанию, но сейчас это нас не устроит. Для создания формы щелкнем в окне редактирования объекта конфигурации по изображению лупы, расположенной напротив пункта Списка в группе Основные формы.

В результате мы перейдем к работе с конструктором экранной формы справочника и в первом диалоговом окне установим значения опций в соответствии с рис. 1.20 (если вы параллельно выполняете описанные действия на компьютере, то эти параметры, скорее всего, уже установлены конструктором по умолчанию).

Примечание

На рис. 1.20 группа элементов управления **Выберите тип формы** предназначена для выбора типа формы. Состав этих элементов определяется видом объекта конфигурации, для которого мы создаем форму. Скажем, для справочников эти объекты одни, а для документов — другие. Число форм каждого типа неограниченно, а при наличии у объекта нескольких форм одного типа одну из них можно назначить в качестве основной. Для указания основной формы при разработке следует установить флажок **Назначить форму основной**.

После этого в следующем окне (рис. 1.21) необходимо будет включить все только что введенные нами реквизиты в создаваемую форму справочника.

Нажав кнопку **Готово**, мы увидим на экране структуру формы списка справочника курсы (рис. 1.22). Форма практически готова, следует только немного изменить ширину колонок в соответствии с подписями в первой строке. Это легко выполнить с помощью мыши (аналогичным образом подбирается ширина колонок для рабочих листов в приложении Microsoft Excel).

🛃 Конфигуратор (учеб	ная версия) - Конфигу	рация		<u>_ </u>
🛛 <u>Ф</u> айл <u>П</u> равка Конфиг <u>у</u>	урация Отладка Админи	стрирова	ние <u>С</u> ервис <u>О</u> кн	а Справка
🗋 🛁 🔛 🕺 🛍 🕯	👌 🍊 🗋 🦊 🕌 🕌	1		· •
菌 Справочник Курсы				_ 🗗 🗙
Основные	– Редактирование —		– Выбор – – – – – – – – – – – – – – – – – – –	
Иерархия	🔘 В списке		🔘 Из формы	
Владельцы	• В диалоге		🔘 Быстрый вы	бор
Данные	🔘 Обоими способа	ми	💿 Обоими спо	собами
Нумерация	Beag po otpore		Land Kan	
🕨 Формы	 ввод по строке Основные формы 	наимено	вание, код	×
Макеты				
Подсистемы				
Ввод на основании	выоора			
Права	Выбора группы			× Q
Интерфейсы	Элемента			× Q
Обмен данными	Группы			x Q
Прочее	「営業営士」	ŧ.		
	Формы			
Действия	• <Назад Д	алее>	Закрыты	Справка
Для получения подсказки	нажмите F1		CAP N	UM ru▼ //

Рис. 1.19. Закладка Формы окна редактирования справочника Курсы

Конструктор фор	рмы справочника	×			
— Выберите тип ф	Рормы:	7			
💿 Форма спис	жа справочника				
🔷 Форма выбо	ора справочника				
🔷 Форма выбо	ора группы справочника				
🔘 Форма элем	иента справочника				
🔘 Форма груп	пы справочника				
🔷 Произвольн	ая форма				
Назначить ф Основная ф Основная ф	о рму основной орма списка и выбора орма элемента и группы	_			
Имя:	ФормаСписка				
Синоним:	Форма списка	1			
Комментарий:		1			
 Командная панель формы сверху Командная панель формы снизу Казад Далее > Готово Отмена Справка 					

Рис. 1.20. Окно конструктора формы справочника Курсы

Конструктор формы справочника	1	×
Располагать элементы управления в	з 1 🗘 колонку(и)	
Реквизиты	Элементы формы	A
⊡. ✓ Список	Табличное поле	<u>.</u>
🦳 🗸 Картинка		Cr.
… 🗸 Код	аы Поле ввода	
— Иаименование	аы Поле ввода	
ПометкаУдаления	аы Поле ввода	
Предопределенный	ы Поле ввода	
— 🗸 ДатаНачала	аы Поле ввода	
	аы Поле ввода	
🗸 Стоимость	аы Поле ввода	
🛄 🗸 Преподаватель	аы Поле ввода	
< Назад Далее >	Готово Отмена Спра	вка

Рис. 1.21. Установка отображения элементов на форме справочника Курсы

📩 Конфигуратор (учебная версия) - Конфигурация	
🛛 🙅 айл Правка Форма Конфигурация Отладка Администрирование Сервис Окна Справка	
I 🗋 💶 X 🖻 🛍 🥔 🖻 🖆 🕅 א 📃 🐨 🖓 🔚 🔭 🖕	
🚟 Справочник Курсы: ФормаСписка	_ 🗆 ×
Справочник Курсы	<u> </u>
Действия - 🔄 🔄 🔩 🍇 🏹 🎦 🗃 - 🌾 🔯 🕐 💭	
Код Наименование Дата начала Продолжительность Стоимость Преподаватель	
аы <аы <Поле ввода> аы <Поле ввода>	
	~
	>
Диалог Модуль Реквизиты	
Для получения подсказки нажмите F1 САР NUM ru 🕇 📩 8, 33 🕫 682 x 75	5 //

Рис. 1.22. Структура формы списка справочника Курсы

Если скептически взглянуть на только что выполненную работу, то можно сказать — наши действия привели к созданию формы, на первый взгляд аналогичной той, с которой мы уже встречались (форма списка, генерируемая по умолчанию). Однако основные плюсы выполненной работы вы скоро увидите. А пока перейдем в режим 1С:Предприятие и заполним справочник курсы информацией с помощью созданной формы (рис. 1.23).

🏪 1С:Предприятие (учебная веј	рсия) - Конфигура	ция			
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервио	с <u>О</u> кна Сп <u>р</u> авка				
I 🗋 🔟 🔛 🐰 🖻 🛍 🧉 🖉) 5 C M		▼ <i>M M</i>	🐥 🖩 🎹 📄 🧝 M M+	м- 🎇 🖕
🗐 Справочник Курсы					
Действия 🕶 🔄 式 😽 🔜	🛛 🗗 🖉 • 🕅	2			
Код Наименование 🔻	Дата начала	Продолжительность	Стоимость	Преподаватель	^
🗕 🔜 PowerPoint	11.08.2007	3	1 200	Петров Иван Александрович	
🗕 😑 Базы данных	07.08.2007	5	1 800	Петров Иван Александрович	
💻 😑 1 Информатика (Excel)	11.08.2007	5	2 200	Петров Иван Александрович	
	·				>
🗐 Справочник Курсы					
Для получения подсказки нажмите Р	-1			CAP	NUM //

Рис. 1.23. Заполнение справочника Курсы в режиме 1С:Предприятие

В принципе, это же мы могли бы сделать и с помощью автоматически генерируемой формы. Плюсы пользовательской формы заключаются в возможности дополнения функциональности расположенных на ней элементов управления, а также просто для внесения в эти элементы небольших изменений с помощью визуальных средств разработки. Так, разработчик конфигурации может написать программную процедуру, выполняемую при определенном событии на форме (при нажатии кнопки или при выборе значения из списка). Кроме того, на подобной форме можно размещать дополнительные объекты.

Примечание

Форма списка в окне редактирования объекта конфигурации (см. рис. 1.22) представлена тремя закладками — Диалог, Модуль и Реквизиты. Закладка Диалог отображает прямоугольную область экрана, которая предназначена для размещения разнообразных элементов управления (кнопок, надписей, полей ввода и др.). Ее содержание определяет интерфейс пользователя в режиме 1С:Предприятие при работе с формой. Закладка Модуль предназначена для размещения процедур на встроенном языке программирования. Эти процедуры обеспечивают обработку действий со стороны пользователя, а также системных событий (например, открытия формы). Закладка Реквизиты отображает совокупность объектов разных типов, принадлежащих форме.

Продемонстрируем теперь на простом примере добавление дополнительных функциональных возможностей в форму справочника. Наша первая задача заключается в небольшой доработке созданной формы. Для этого потребуются только визуальные средства конструктора форм.

Примечание

В систему 1С:Предприятие заложены разнообразные технические средства разработки. Это, прежде всего, визуальные конструкторские возможности, которые позволяют с помощью движений мыши и заполнения определенных полей создавать необходимый пользовательский интерфейс. Также в системе имеется свой встроенный язык программирования, с которым буквально через несколько страниц мы начнем знакомиться. Кроме того, существуют разнообразные конструкторы (или мастера), позволяющие генерировать программный код с помощью несложного диалога с разработчиком. На рис. 1.23 в столбце **Наименование** отражаются названия курсов. Наша задача заключается в том, чтобы заменить это стереотипное название более осмысленным — **Названия курсов**. Для этого необходимо в окне редактирования объекта конфигурации двойным щелчком на надписи **Наименование** открыть окно свойств этого элемента управления и поменять содержание в поле **ТекстШапки** (рис. 1.24). В результате проделанных действий изменяется только внешний вид формы подпись над одной из колонок, а само имя реквизита наименование остается прежним.

📩 Конфигуратор (учебная версия) - Конфигурация		Y
🛛 <u>Ф</u> айл Правка Форма Конфигурация Отладка Админ	: СправочникСписок.Наим + АІ Щ № × ∨ хОсновные:	×
🚟 Справочник Курсы: ФормаСписка	Имя Наименование	
	Данные Наименование 🔻	
	ЭлементУпр Поле ввода 🔹 🔻	
	ДанныеФла 👻	
Код Названия курсов Дата начала П	Доступность 🗸	
аы <аы <Поле ввода> аы <Поле ввода> а	видимость 🔽	
	ТолькоПрос 🗌	
<	ТекстШапки Названия курсов	
K	ТекстПолеа	\geq
📃 🗏 Диалог 📃 Модуль 📑 Реквизиты	Имя, по которому будет выполняться	
Для получения подсказки нажмите F1 САР NUM	алгоритме модуля	~

Рис. 1.24. Изменение текста шапки в форме списка справочника Курсы

Примечание

Окном свойств в дальнейшем мы будем пользоваться часто, поэтому стоит обратить на него внимание.

🏪 1С:Предприятие (учебная вер	сия) - Конфигура	ция			_ 🗆 🗵
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис	<u>О</u> кна Сп <u>р</u> авка				
🗋 🔟 🔛 🔏 🍙 🍙 🖾	1 5 C M		▼ 18° 18	🕒 📴 🗸 🛛 📰 🔁 🦧	м м+ м- 🎇 🗸
🗐 Справочник Курсы					_ 🗆 ×
Действия 🗸 🔄 🛬 🔩 🄜	🕅 🗗 🔟 - 🕅	0 ?			
Код Названия курсов 🔹	Дата начала	Продолжительность	Стоимость	Преподаватель	^
🗕 😑 3 PowerPoint	11.08.2007	3	1 200	Петров Иван Александрович	
🗕 😑 Базы данных	07.08.2007	5	1 800	Петров Иван Александрович	
🗕 💶 Информатика (Excel)	11.08.2007	5	2 200	Петров Иван Александрович	
					×
<					>
🗐 Справочник Специалисты 🗐	Справочник Курсы				
Для получения подсказки нажмите F	1				CAP NUM

Рис. 1.25. Результат изменений в форме списка справочника Курсы

Теперь, если перейти в режим 1С:Предприятие и открыть справочник курсы, можно увидеть результат — надпись в колонке названий курсов изменилась (рис. 1.25). В форме, генерируемой системой 1С:Предприятие по умолчанию, даже такое простое изменение было для нас невозможно.

Выполним еще одну несложную доработку формы — для продолжительности курсов установим ограничение по интервалу — от двух до семи недель. Это позволит пользователю в процессе заполнения справочника избежать случайных ошибок. Для реализации подобной возможности достаточно всего лишь в окне свойств установить значения параметров МинимальноеЗначение и МаксимальноеЗначение (рис. 1.26). В нашей ситуации, как сказано ранее, продолжительность курсов должна лежать в интервале от двух до семи недель.

🛃 Конфигуратор (учебная версия) - Конфигурация	
🛛 <u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование <u>С</u> ервис	<u>О</u> кна Сп <u>р</u> авка
	💌 🖗 🛍 🔭 🐅 🖪 唑 🖕
🚟 Справочник Курсы: ФормаСписка	: СправочникСписок Продолжительность 🕈 🗙
Справочник Курсы	
Действия - 😫 🚔 🔫 📇 🔛 🗿 🗇 🕅 - 🏹 🔘 🕐 💭	• Основные:
	ТипЗначения Число 💌
Код Названия курсов Дата начала Продолжительность	Длина 10 🗘
ab < ab ab <! ab ab</th <td>Точность 0</td>	Точность 0
	Неотрицательный 🗌
	Доступность
	Формат
	МинимальноеЗначение 2
	МаксимальноеЗначение 7
Действия 👻	•Использование:
Справочник Кирсы	Тип значения, используемого для хранения данных,
	ТипЗначения, ValueType
Для получения подсказки нажмите н	

Рис. 1.26. Окно свойств поля ввода продолжительности курсов

Теперь при работе в режиме 1С:Предприятие в случае неправильно указанной продолжительности курсов система сама будет сообщать нам об ошибке (рис. 1.27).

1С:Предприятие (учебная версия) 🛛 🗙			
В поле введены некорректные данные. Отменить ввод?			
Нажмите "Да" для отмены ввода или "Нет" для продолжения редактирования			
Да Нет			

Рис. 1.27. Реакция системы 1С:Предприятие на ошибочный ввод данных

Автоматический контроль возможных ошибок дает на практике существенное преимущество при использовании самостоятельно разработанной формы по сравнению с формой, генерируемой системой по умолчанию.

После разработки ряда справочников (в том числе справочника курсы) можно сказать, что знакомство с конфигуратором системы 1С:Предприятие 8.1 произошло, и теперь наступила очередь познакомиться со встроенным языком программирования. При этом мы начнем не с самостоятельного написания программ, а пока, в качестве начальных действий, воспользуемся возможностями конструктора, а именно конструктора печати. Это одно из средств, которое существенно облегчает труд разработчика. В этом случае мы фактически продолжаем работу в режиме визуальной разработки, а программный код генерируется конструктором автоматически как результат наших действий. Тем самым программные конструкции становятся доступными для изменения и изучения (что весьма помогает на начальном этапе знакомства со встроенным языком программирования).

Такая технология (получение программного кода с помощью конструктора и его изменение программистом при необходимости) позволяет существенно ускорить процесс разработки и уменьшить количество ошибок и неточностей. Именно этой возможностью мы и воспользуемся в следующем разделе.

Печатная форма справочника

Практически всегда кроме электронных форм справочников при работе пользователям требуются и их печатные варианты (это же распространяется и на другие объекты, с которыми мы столкнемся в дальнейшем). В этом разделе мы узнаем, каким образом можно автоматизировать процесс создания печатных форм для объектов системы 1С:Предприятие 8.1. Приводимый далее пример будет связан с уже известным справочником курсы, и поэтому начнем с того, что в режиме конфигуратора откроем окно редактирования этого справочника.

Теперь для дальнейших действий (создания печатной формы) необходимо перейти на закладку Макеты (рис. 1.28), где в нижней части окна расположена

🛃 Конфигуратор (учебная	і версия) - Конфигурация 📃 🔲 🗙
🛛 <u>Ф</u> айл <u>П</u> равка Конфигурац	ия Отладка Администрирование <u>С</u> ервис <u>О</u> кна
Справка	
I 🗋 🛋 🖬 🕹 🛍	🎒 🔄 🖢 d' 🕅 🙀 🐥
菌 Справочник Курсы	_ @ ×
Основные	
Иерархия	
Владельцы	Макеты
Данные	
Нумерация	
Формы	
🕨 Макеты	
	Конструкторы 👻
Действия 👻 <На	азад Далее> Закрыть Справка
Для получения подсказки наж	кмите F1 CAP NUM ru т

Рис. 1.28. Закладка Макеты окна редактирования справочника Курсы

кнопка Конструкторы. Щелкнув по ней, воспользуемся еще одним из инструментов разработки системы 1С:Предприятие, который называется Конструктор печати.

Сам по себе диалог несложен. В первом открывшемся окне потребуется сделать установку для создания процедуры в модуле формы (рис. 1.29). Это приведет к тому, что автоматически созданный конструктором печати текст процедуры будет располагаться в окне редактирования объекта формаСписка на закладке **Модуль** (см. рис. 1.26). Именно там данную процедуру можно найти по завершении работы конструктора, а в дальнейшем мы увидим, как результат работы конструктора можно изменить.

Конструктор печати сп	равочника	×
∟⊙ Создать новую проц	едуру ————	
🔘 в модуле:	• Печать элемента справочника 📀 Печать списка справочника	
💿 в модуле формы:	ФормаСписка	•
Имя процедуры:	Печать	×
🖵 🔿 Использовать найде	енную процедуру	
Местоположение	Процедуры	^
		~
	<Назад Далее> ОК Отмена Спр	равка

Рис. 1.29. Окно конструктора печати

А пока вернемся к техническим действиям. Следующее окно (рис. 1.30) позволяет определить, какие реквизиты шапки мы хотим расположить на печатной форме. В данном случае следует с помощью соответствующей пиктограммы перенести все реквизиты, кроме поля код. Если поле код присутствует в списке реквизитов шапки, то удалить его можно с помощью двойного щелчка мыши.

После нажатия кнопки Далее открывается завершающее окно конструктора, где осталось лишь сделать установку (рис. 1.31) для создания новой кнопки, при нажатии на которую в режиме 1С:Предприятие будет вызываться процедура формирования печатной формы справочника. При этом можно выбрать другое имя кнопки, если слово печать нас не устраивает.

Таким образом, действия, выполняемые в диалоговом окне конструктора печати, достаточно просты, а итог, как мы увидим далее, весьма эффективен. Один из ре-

зультатов работы конструктора печати появляется сразу после нажатия кнопки **ОК** (см. рис. 1.31).

Конструктор печати справочника		×
🕇 👃 🧏 Сброс		
Реквизиты шапки		Реквизиты справочника
🗕 Код		🕳 Код
🗕 Наименование		🗕 Наименование
🗕 ДатаНачала		🗕 ДатаНачала
🗕 Продолжительность]	- Продолжительность
🗕 Стоимость		- Стоимость
💻 Преподаватель		🗕 Преподаватель
	-	
	44	
<Назад Далее	e>	ОК Отмена Справка

Рис. 1.30. Формирование реквизитов шапки для печатной формы справочника Курсы

На рис. 1.32 показан так называемый *макет печатной формы*, представляющий собой заготовку (шаблон) для печатного документа, который будет создаваться системой при работе в режиме 1С:Предприятие. Макет печатной формы доступен для разработчика — его всегда можно открыть и посмотреть (а при необходимости и изменить) на закладке **Макеты** (см. рис. 1.28). После завершения работы конструктора на закладке вы увидите пиктограмму нового объекта конфигурации — макета Печать.

Примечание

Для проектирования макета используется редактор табличного документа, входящий в систему 1С:Предприятие. В процессе создания макета в ячейки вводится текст (это может быть выполнено не только в результате работы конструктора, но и непосредственно разработчиком). При этом можно задавать параметры форматирования для ячеек. Печатные формы, как правило, содержат несколько различных областей — заголовочную область, таблицу, место для итоговых показателей и др. Это связано с тем, что информация, выводимая в отчеты, обычно неоднородна — есть табличные данные, заголовок, область размещения подписей. Макет представляет собой совокупность прямоугольных областей, каждая из которых служит для формирования какой-то части печатной формы. Заметим, что табличный документ (его визуальное представление и есть печатная форма) и макет — это различные объекты. И, в общем, они не связаны между собой. Макет представляет собой некий эталон для табличного документа, показывающий, как необходимо создавать табличный документ. Можно сказать, что в конфигураторе мы определяем вид печатной формы, которую хотим увидеть затем при работе в режиме 1С:Предприятие.

K	онструктор печати справочн	ика	x
	— Вызов процедуры — не вставлять вызов процед	уры	
	• новая кнопка	Печать	
	🔘 выбрать кнопку	•	
	Печатать без предваритель	ного просмотра	
	— Таблица "только для чтения	e.	
	🗌 Защита таблицы		
=			_
	<has< td=""><td>ад Далее> UK Отмена Справка</td><td></td></has<>	ад Далее> UK Отмена Справка	

Рис. 1.31. Завершающее окно конструктора печати

🚵 Конфигур	оатор	(уче	бная версия) - Конфигур	рация						
<u> Ф</u> айл <u>П</u> ра	🙅 айл <u>П</u> равка Таблица Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка									
🗋 🔟 🚽 🔏 📾 🖾 😟 🖕 🖉 🛤 📃 🔍 🐨 🐨 🕼 🎘 🖓 💭 🗸										
	×	.	E 🛯 🖡 👫 🖉 🖕							
R1C1			▼ -⊨	=						
🛓 Справо	чник І	Курсь	я: Печать				_ 🗆 ×			
		1	2	3	4	5	6 ^			
	1									
Заголовок	2		Курсы							
	3									
Шапка	4									
	5		Наименование	Дата начала	Продолжительность	Стоимость	Преподаватель			
Группа	6		<haumenobanue></haumenobanue>							
Элемент	7		<Наименование>	≺ДатаНачала≻	<Продолжительность>	<Стоимость>	<Преподаватель>			
	8									
	9						~			
<							>			
Для получени	кя подо	казкі	и нажмите F1				CAP NUM ru -			

Рис. 1.32. Макет печатной формы

Другим результатом работы конструктора печати является процедура печать (), которая выполняется при работе в режиме 1С:Предприятие и взаимодействует с макетом. Результатом этого взаимодействия является табличный документ, отражающий информацию из справочника (документа или другого объекта). Что касается названия процедуры, то мы его выбрали ранее сами в процессе диалога с конструктором (см. рис. 1.29).

Возникает вопрос — как эту процедуру найти, или где она находится? Для этого в окне конструктора формы списка справочника следует перейти на закладку **Мо**дуль, где и располагается ее текст (он приведен в листинге 1.1).

Кроме того, на самой форме списка справочника конструктором печати создана кнопка **Печать** (она располагается в правой нижней части на рис. 1.33), щелчком по которой в режиме 1С:Предприятие эту процедуру можно выполнить.

📩 Конфигуратор (учебная	🍌 Конфигуратор (учебная версия) - Конфигурация 📃 🗆 🗙								
Файл Правка Форма Конфигурация Отладка Администрирование Сервис Окна									
🛛 Справка									
🗋 🛋 🖬 X 🖻 🚳	la 🗟 🖕 🕹 🖉	34 94		▼					
🚟 Справочник Курсы: Ф	ормаСписка			o x					
				^					
🗏 🖳 🕄 🔛									
Продолжительность	Стоимость	Преподаватель							
а> 🚺 <Поле ввода>	аы <Поле ввода>	<mark>аы</mark> <Поле ввода≻							
			>						
			Печать	_					
<				>					
🗏 Диалог 🗏 Модуль	📒 Реквизиты								
Для получения подсказки нах	K CAP NUM I	u 🕶 📑 🔤 0, 0	📫 719 x 167						

Рис. 1.33. Размещение кнопки Печать на форме списка справочника

Листинг 1.1. Процедура печати справочника Курсы

```
Процедура Печать (Элемент)
//{ { КОНСТРУКТОР ПЕЧАТИ СПИСОК (Печать)
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора внесенные вручную
// изменения будут утеряны!!!
ТабДок = Новый ТабличныйДокумент;
Макет = Справочники.Курсы.ПолучитьМакет ("Печать");
// Заголовок
Область = Макет.ПолучитьОбласть ("Заголовок");
ТабДок.Вывести (Область);
// Шапка
Область = Макет.ПолучитьОбласть ("Шапка");
ТабДок.Вывести (Область);
Выборка = Справочники.Курсы.Выбрать ();
Элемент = Макет.ПолучитьОбласть ("Элемент");
Пока Выборка.Следующий () Цикл
    Элемент.Параметры.Заполнить (Выборка);
```

```
ТабДок.Вывести (Элемент);
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Ложь;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать();
//}}_КОНСТРУКТОР_ПЕЧАТИ_СПИСОК
КонецПроцедуры
```

Первая и последняя строки листинга 1.1 ограничивают содержание процедуры. Ключевое слово процедура (вместе с ее именем справа) определяет начало процедуры, а другое — конецпроцедуры говорит о завершении ее текста.

При рассмотрении листинга 1.1 бросается в глаза, что несколько строк начинаются с двух наклонных черт. Это комментарий, который не выполняется, а предназначен лишь для наглядности текста программы. В первой выполняемой строке создается новый объект типа табличныйдокумент, а переменной табдок присваивается ссылка на него.

Примечание

Термин "ссылка" уже упоминался ранее, но здесь попробуем пояснить его более точно. В процессе работы программы в памяти компьютера присутствует много объектов, и для работы с ними ключевой является информация об их местонахождении в памяти. Фактически эта информация и содержится в ссылке (ссылке на конкретный объект). После выполнения первой строки процедуры переменная Taбдок содержит ссылку на новый табличный документ, что позволяет нам в дальнейшем работать со свойствами и методами этого объекта.

Упрощенно объект типа табличный документ представляет собой таблицу, но включает в себя разнообразные свойства и методы работы с ней. В строке Макет = Справочники.Курсы.Получить Макет ("Печать");

используется метод получить Макет (), в качестве входного параметра которого указывается имя макета печатной формы, который был создан непосредственно конструктором.

Примечание

Если сформулировать более точно, то табличный документ является объектом, который предназначен при разработке конфигураций для формирования печатных форм основных объектов (справочников, документов, регистров и др.), а также для форм, в которых информация представляется в табличном виде. Используя свойства и методы этого объекта, можно обеспечить необходимый пользователю вид таблицы в режиме 1С:Предприятие.

Автоматически созданный конструктором макет с именем печать имеет несколько областей, одна из которых содержит заголовочную информацию. Она так и называется — Заголовок (см. рис. 1.32). В следующих двух строках процедуры в табличный документ выводится эта первая секция:

```
Область = Макет.ПолучитьОбласть("Заголовок");
ТабДок.Вывести(Область);
```

Таким способом происходит заполнение табличного документа на основании макета. Содержание областей макета и их названий, сформированное по умолчанию, мы в принципе можем изменить, однако в рассматриваемом примере этого делать не будем. Для тех, кто любит заглядывать вперед, скажем, что в конце этой главы приведен пример создания макета и процедуры печати, где все технические действия выполняются самостоятельно без помощи конструктора печати. Созданная здесь с помощью конструктора печати процедура поможет нам написать программный код для заполнения табличного документа.

Учитывая вид макета (см. рис. 1.32), можно сказать, что печатная форма справочника курсы представляет собой совокупность одинаковых по структуре строк, подписи для которых располагаются в секции Шапка. Метод Выбрать (), применяемый к справочнику курсы, позволяет подготовить дальнейшую последовательную выборку строк справочника, которая непосредственно осуществляется с помощью другого метода — Следующий (). Этот метод последовательно извлекает все имеющиеся элементы выборки (в данном случае в выборку входят все строки, но может быть и по-другому), и при каждом проходе цикла, организованного с помощью строки

Пока Выборка.Следующий() Цикл

мы получаем очередной элемент справочника. Элемент справочника также представляет собой один из объектов системы 1С:Предприятие. В этом цикле перебора строк справочника производится заполнение секции элемент информацией, содержащейся в реквизитах. Для этого используется свойство параметры, позволяющее получить доступ к значениям конкретных параметров, расположенных в области макета. В нашем примере производится заполнение параметров информацией об очередном элементе справочника:

Элемент.Параметры.Заполнить (Выборка);

Примечание

Свойство Параметры содержит коллекцию параметров макета табличного документа.

После установки значений параметров с помощью метода вывести() область табличного документа заполняется на основании макета. И результат выполнения цикла (перебора элементов справочника) заключается в том, что в табличном документе создается несколько строк, и их количество совпадает с числом элементов в справочнике Курсы.

Завершающая часть процедуры связана с настройкой табличного документа — режимом вывода заголовков, возможности редактирования полученного документа и т. д.

Так, в конструкции

ТабДок.ОтображатьСетку = Ложь;

указано, что не надо выводить сетку. В качестве альтернативы для свойства табличного документа отображатьСетку можно указать значение истина. Тогда для всего табличного документа (в режиме 1С:Предприятие) будет установлена сетка. Свойство защита позволяет при установке для него значения истина защитить табличный документ от редактирования и копирования. В нашем примере мы этого не делали:

ТабДок.Защита = Ложь;

Однако если для данного свойства установить значение истина, то в режиме 1С:Предприятие скопировать табличный документ в буфер обмена не удастся. Свойство толькопросмотр позволяет обеспечить два варианта работы с табличным документом:

- только просмотра информации (при значении истина), в этом случае будут недоступны для редактирования ячейки и рисунки, у которых установлено свойство Защита (установка производится в окне свойств);
- возможность изменения информации (при значении ложь) в табличном документе при работе в режиме 1С:Предприятие.

Свойство ОтображатьЗаголовки содержит признак отображения заголовков колонок и строк таблицы. Если для данного свойства установлено значение Истина, то в табличном документе будут выведены номера строк и колонок в левой и верхней частях таблицы.

Обратим внимание на последнюю строку процедуры листинга 1.1:

ТабДок.Показать();

в которой используется метод Показать () для отображения табличного документа на экране.

Теперь осталось перейти в режим работы 1С:Предприятие, открыть форму списка справочника курсы, внести в справочник несколько названий курсов и щелкнуть по кнопке **Печать**. На рис. 1.34 показана заполненная печатная форма (вместе с электронной формой списка) в режиме 1С:Предприятие.

🏪 1С:Предпр	риятие (у	чебная версия	а) - Конфигу	рация						
Daŭn Πραει	va Tafiqu	ца Операции (Гереис Пина	. Справка						
	ка табли	ца операции у	<u>Бервис О</u> кна	а справка				-	(Tiles	
I 🗋 🚢 📕	- <mark>Ж</mark> 🔁	🖷 🔿 🔍	5 C M			▼ [8] [8]	🕒 😁 🖵	🎹 📃 🎇 М	M+ M- 🎇 🗕	
🗐 Справочник Кирсы 🛛 🗸 🗆 🗸										
	<u>e</u> +.	X 🗆 🛛								
деиствия •	35.	 		^{PK} <u>11</u>						
Код Наз	звания кур	осов 🝷 Дат	а начала	Продолжит	ельность	Стоимость	Преподаватель	,		^
🚽 3 Pov	werPoint	11.0	08.2007		3	1 200	Петров Иван А	лександрович		
🗕 2 Баз	зы данных	07.0	08.2007		5	1 800	Петров Иван А	лександрович		
💻 1 Инс	форматик	a (Excel) 11.0	08.2007		5 2 200 Петров Иван Александрович					
										\sim
<										>
🔓 Таблица										. 🗆 🗙
	<i>.</i>	_								~
r r	курсь	əl 🛛								
_										
	Код	Наименова	ние	Дата начала		Іродолжите	льность	Стоимость	Преподавател	ь
	3	PowerPoint	11.0	8.2007 0:00:00			3	1 200	Петров Иван Алекс	андров
	2	Базы данных	07.0	8.2007 0:00:00		5		1 800	Петров Иван Алекс	андров
	1	Информатика (Excel) 11.0	8.2007 0:00:00			5	2 200	Петров Иван Алекс	андро
	Kunni	N T - 4								
Справочник	. Курсы	A lao	лица							
Для получения	подскази	и нажмите F1							CAP 1	NUM //

Рис. 1.34. Форма списка и печатная форма справочника Курсы

При желании печатную форму можно скорректировать (изменить содержание ячеек), а также сохранить как рабочую книгу Microsoft Excel (для этого следует в меню **Файл** при сохранении воспользоваться вариантом рабочей книги Microsoft Excel).

Созданные программные фрагменты были сгенерированы системой самостоятельно, однако сделанными разработками можно воспользоваться для внесения дополнений. Именно так мы и поступим в следующем разделе.

Внесение изменений в печатную форму

Попробуем самостоятельно немного изменить вид табличного документа. Из рис. 1.34 видно, что вместе с датой начала курсов выводится еще и время, что нам не требуется, и теперь наша задача этот недостаток устранить.

В конфигураторе откроем уже знакомый макет печать, который отражает все поля, выводимые в табличный документ, и щелкнем правой кнопкой на параметре датаначала (рис. 1.35). В результате перед нами откроется контекстное меню, в котором следует выбрать раздел **Свойства**, где мы установим необходимый формат даты. Если после сделанных изменений перейти в режим 1С:Предприятие и открыть печатную форму справочника, то вы увидите только дату в требуемом формате без информации о времени (рис. 1.36).

Как и в большинстве инструментальных сред разработки, в режиме конфигуратора системы 1С:Предприятие имеется возможность размещения на экранных формах разнообразных элементов управления.

Примечание

Редактор форм позволяет размещать на форме различные элементы управления, задавать их размеры, выравнивать границы, определять механизм поведения при различных действиях пользователя. Основное назначение элементов управления заключается в обеспечении функциональности при работе с формой.

Сделаем в рассматриваемой форме списка справочника Курсы еще одну доработку — обеспечим возможность печати только коротких курсов, т. е. тех курсов, которые продолжаются только две недели. Для этого на экранную форму списка справочника следует добавить элемент управления типа флажок, с помощью которого можно будет сформировать либо полный печатный вариант справочника, либо только перечень коротких (двухнедельных) курсов. Интерфейс формы списка справочника в этом случае необходимо дополнить элементом управления (рис. 1.37) и обеспечить его функциональность.

Примечание

Флажок — это элемент управления, который предназначен для выбора одного значения (из двух либо трех возможных). Тип значения, связанный с флажком, определяет количество его состояний. Если у флажка установлен тип значения Булево, то он имеет два состояния (установлен либо снят). А если выбран тип значения Число, то флажок может иметь три состояния (установлен, снят либо неопределенное значение).

В режиме конфигуратора при открытой форме (в данном случае мы работаем с формой списка) меню в верхней части окна изменяется — в нем появляется пункт

A	(онфигурат	ор (уч	ебная вер	сия) - Конф	ригурация			
<u>φ</u>	айл <u>П</u> равка	Табл	ица Конфиг	урация От	ладка Администрирова	ние <u>С</u> ервис <u>О</u> кна Сп <u>р</u>	Свойства: Ячейки	# X
) 🛋 🖬 🖡	X 🖻	🖻 🔿 [a 🖕 d	23 23	▼ 98 * 99	21 🕅 🍽 🗙 ∨	
Ячейки								-
		• •				ГиперСсылк 🗌	^	
R7C	.4			▼ -	= ДатаНачала		▼Макет:	
	Справочник	к Курс	ы				Заполнение Параметр	-
	🔓 Справо	чник І	Курсы: Печ	Параметр ДатаНачала				
			1	2	3	4	ПараметрРа	
		1					Использова Ячейка	-
	Заголовок	2		Курсі	ы		• Значения:	
-		3					Содержит Зн 📃	
	Шапка	4					ТипЗначения	▼ ×
		5		Код	Наименование	Дата начала	ЭлементУпр	-
	Группа	6		<kad></kad>	<haumenobanue></haumenobanue>		Change and the state of the sta	
	Элемент	7		≺Код≻	<Наименование>	<ДатаНачала>	Формат дФ=оо/мм/уу	ч 🗸
		8					Формат содержимого ячейки	
		9					Формат, Format	
Для	получения п	одсказ	ки нажмите	F1				

Рис. 1.35. Окно свойств ячейки макета

🏪 1С:Пред	цприятие (у	чебная версия) - Кон	фигурация						
<u>∎ Ф</u> айл <u>П</u> р	<u>Ф</u> айл <u>П</u> равка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка								
I 🗋 🛋 🖬	I 🗋 🖬 🔏 🖓 ち 🖒 🛤 🔜 🔍 🐄								
🛓 Таблиц	ιa				_ 🗆 ×				
	Курсь	ы			<u>^</u>				
	Код	Наименование	Дата начала	Продолжительность	Стоимость				
	3	PowerPoint	11/08/07	3	1 200				
	2	Базы данных	07/08/07	5	1 800				
	1	Информатика (Excel)	11/08/07	5	2 200				
<									
🗐 Справочн	ник Курсы	🚠 Таблица							
Для получен	ния подсказн	ки нажмите F1			CAP NUM				

Рис. 1.36. Исправленная печатная форма справочника Курсы

	IC:Предприятие (учебная вер	осия) - Конфигура	щия			<u>_ 0 ×</u>
! ⊈	айл <u>П</u> равка Операции <u>С</u> ервио	: <u>О</u> кна Сп <u>р</u> авка				
) 🛋 🖬 X 🖻 📽 🍯 🖸	1 5 C M		▼ 36 '86	🕒 📴 🗸	M M+ M- 🎇 🗸
a	Справочник Курсы					_ 🗆 ×
Де	ействия 🗸 😫 🔄 式 🍇 🔜	1 11 To 11 - K	2 ?			
	Код Названия курсов 🔹	Дата начала	Продолжительность	Стоимость	Преподаватель	^
	PowerPoint	11.08.2007	2	1 200	Петров Иван Александрович	
	🗕 2 Базы данных	07.08.2007	5	1 800	Петров Иван Александрович	
	 1 Информатика (Excel) 	11.08.2007	5	2 200	Петров Иван Александрович	
			1	1		×
						>
	📃 Короткие кур	сы				
						Печать
J	Справочник Курсы					
Для	я получения подсказки нажмите Р	71				CAP NUM

Рис. 1.37. Добавление флажка в форму списка справочника Курсы

Форма. В рассматриваемой ситуации от нас требуется в этом пункте меню выбрать пункт Вставить элемент управления, а затем в списке элементов щелкнуть на строке Флажок. Для флажка установим имя короткие Курсы, а аналогичный заголовок (подпись на форме рядом с флажком) среда разработки сформирует автоматически.

Теперь от нас требуется немного подкорректировать процедуру печати, ранее созданную конструктором. Изменения отражены в тексте листинга 1.2. Это будет нашим первым шагом самостоятельного написания программного кода. Вообще такой стиль программирования достаточно популярен — часть работы выполняется с помощью средств автоматизации, а затем ряд корректирующих действий производится вручную.

Листинг 1.2. Процедура печати справочника Курсы (с отбором двухнедельных курсов)

Процедура Печать (Элемент) //{{ КОНСТРУКТОР ПЕЧАТИ СПИСОК (Печать) // Данный фрагмент построен конструктором. // При повторном использовании конструктора внесенные вручную // изменения будут утеряны!!! ТабДок = Новый ТабличныйДокумент; Макет = Справочники. Курсы. Получить Макет ("Печать"); // Заголовок Область = Макет.ПолучитьОбласть ("Заголовок"); ТабДок.Вывести (Область); // Шапка Область = Макет.ПолучитьОбласть ("Шапка"); ТабЛок.Вывести (Область); Если (ЭлементыФормы.КороткиеКурсы.Значение=Истина) Тогда СтруктураОтбора = Новый Структура; СтруктураОтбора.Вставить ("Продолжительность", 2); Выборка = Справочники. Курсы. Выбрать (,, СтруктураОтбора); Иначе Выборка = Справочники. Курсы. Выбрать (); КонецЕсли; Элемент = Макет.ПолучитьОбласть ("Элемент"); Пока Выборка.Следующий() Цикл Элемент.Параметры.Заполнить (Выборка); ТабДок.Вывести (Элемент); КонецЦикла; ТабДок.ОтображатьСетку = Ложь; ТабДок.Защита = Ложь; ТабДок.ТолькоПросмотр = Ложь; ТабДок.ОтображатьЗаголовки = Ложь; ТабДок.Показать (); //}} КОНСТРУКТОР ПЕЧАТИ СПИСОК КонецПроцедуры

Перед тем как прокомментировать строки листинга 1.2 с новыми программными конструкциями, учтем, что многие сразу попробуют посмотреть результат выполнения данной процедуры в режиме 1С:Предприятие. В связи с этим сделаем важное замечание: необходимо вернуться в окно редактирования объекта конфигурации и у реквизита продолжительность установить значение Индексировать для свойства Индексировать (рис. 1.38). Зачем это надо сделать, поясним далее, а пока перейдем в режим 1С:Предприятие и посмотрим, как выполненные изменения сказываются на формировании табличного документа.

Теперь уже можно внимательнее рассмотреть текст выполненных в процедуре изменений. Так, в программный код мы добавили оператор условия

Если (ЭлементыФормы.КороткиеКурсы.Значение=Истина) Тогда

который позволяет выполнить последующий (до ключевого слова иначе) фрагмент программы только при выполнении указанного условия.

🛃 Конфигуратор (учебі	ная версия) - Конфигурация			
🛛 <u>Ф</u> айл <u>П</u> равка Конфигу	рация Отладка Администрирование (Свойства: Реквизит		
🗋 🛋 🖬 🐰 🖻 🖷	1 🗟 🗋 5 C 🕅 M	21 🖾 🎦 × ∨		
🗎 Справочник Курсы		• Основные:		^
Основные Иерархия Владельцы ▶ Данные Нумерация Формы Макеты Подсистемы Ввод на основании Права	Длина кода Длина наименования Тип кода • Число • Строка • Строка • ДатаНачала • Стоимость • Стоимость	Имя Продолз Синоним Продолз Комментарий Полотование Использование Индексировать Полнотекстовый поиска Тип данных: Тип Данных: Тип Дисло Длина Почность Точность О Имя объекта метаданны 1000000000000000000000000000000000000	жительность жительность Для элемента Индексировать Использовать 10 ↓ 1 ↓	V V V
Для получения подсказки і	нажмите F1			

Рис. 1.38. Окно свойств реквизита Продолжительность

Примечание

Оператор Если управляет выполнением программы, основываясь на результате логического выражения. Операторы, следующие за ключевым словом Тогда, выполняются, если результатом логического выражения является значение Истина. Операторы, следующие за ключевым словом Иначе, выполняются (до оператора КонецЕсли), если результат логического выражения в конструкции Если оказался равен значению Ложь. Заметим, что конструкция Если имеет еще расширенный вариант синтаксиса, информацию о котором можно легко найти с помощью системы электронной помощи, включенной в программный продукт 1С:Предприятие 8.1.

Программно состояние флажка короткиеКурсы определяется с помощью свойства Значение этого объекта, и если флажок установлен, то Значение=Истина.

Примечание

Слова Истина и Ложь являются ключевыми во встроенном языке программирования системы 1С:Предприятие, и они часто используются в подобных программных конструкциях. Если переменные имеют тип данных Булево, то они могут принимать только значения Истина и Ложь.

Таким образом, если пользователем рассматриваемый флажок на форме установлен, то выполняется фрагмент процедуры, позволяющий сформировать условия отбора для метода Выбрать ():

```
СтруктураОтбора = Новый Структура;
СтруктураОтбора.Вставить("Продолжительность",2);
Выборка = Справочники.Курсы.Выбрать(,,СтруктураОтбора);
```

Объект Структура представляет собой коллекцию пар. Каждая пара состоит из имени переменной (ключа) и значения переменной, которое нас устраивает при дальнейшем отборе (это значение ключа). В данном случае в структуре отбора размещается переменная продолжительность (реквизит справочника), а для ее значения устанавливается цифра 2 (две недели). После этого для справочника применяется метод Выбрать () с использованием параметра СтруктураОтбора, и этому методу стоит дать небольшой комментарий.

У метода выбрать () четыре параметра, но все они являются необязательными. Мы здесь используем только третий параметр, который задает поле и значение отбора для выборки. Поэтому в программной конструкции место первых двух необязательных параметров необходимо отметить запятыми.

Примечание

В качестве реквизитов для отбора могут использоваться только поля Код, Наименование и реквизиты справочника, для которых в конфигураторе признак индексирования установлен в значение Индексировать или в значение Индексировать с доп. упорядочиванием.

В связи с этим стоит обратить внимание на то, чтобы для реквизита продолжительность признак индексирования был установлен в значение индексировать (см. рис. 1.38). После этого метод выбрать () позволяет сформировать выборку элементов справочника по заданным условиям.

Если в режиме работы 1С:Предприятие флаг короткие Курсы на форме не установлен, то метод выбрать () в тексте процедуры вызывается без указания отбора и позволяет получить в выборке все элементы справочника курсы.

Доступ к созданным на форме элементам обеспечивается с помощью объекта элементыформы. Далее правила синтаксиса требуют указать (через точку) имя элемента управления, который нам необходим, а затем, также через точку, его свойство. Таким образом, установка флажка на форме проверяется с помощью следующей программной конструкции:

ЭлементыФормы.КороткиеКурсы.Значение = Истина

В результате добавленного программного кода при снятии флажка на форме табличный документ будет заполняться всеми элементами, внесенными в справоч-

ник курсы, а при его установке в печатную форму попадут только те курсы, продолжительность которых равна двум неделям.

Примечание

Объект ЭлементыФормы является коллекцией размещенных на форме элементов управления.

Для тестирования дополненной разработки в режиме работы 1С:Предприятие внесем в справочник двухнедельный курс, а затем при установленном флажке заполним печатную форму. Один из вариантов подобной ситуации показан на рис. 1.39. В данном случае условие отбора достаточно простое и рассматриваемого программного отбора вполне достаточно. Однако при усложнении задачи удобнее воспользоваться языком запросов (см. главу б).

🏪 1С:Пре,	дприятие	(учебная ве	рсия) - Кон	фигура	ция									- D ×
<u>Ф</u> айл <u>П</u>	равка Таб	лица Операці	ии <u>С</u> ервис	<u>О</u> кна С	Справка									
I 🗋 🛋	- 🗋 🖬 😹 📾 📾 🖸 🖕 d 🛤 🔜 🔽 🐨 🐨 🖓 🖬 🛅 🖉 🗶 🖬 🖬 🎘 🖕													
🗐 Спран	🗐 Справочник Курсы 💶 🗸 🖂													
Действия	• 1 🗄 🖆	: 🛃 🍓 🖬	1 1	() - T _k	2?									
Код	Названия н	курсов 🔻	Дата начал	1a I	Продолжит	ельность	Стоим	ость	Препода	авател	ь			^
- 3	PowerPoint		11.08.2007			2		1 200	Петров	Иван А	Александрович			
- 2	Базы данн	ых	07.08.2007			5		1 800	Петров	Иван А	Александрович			
= 1	Информаті	ика (Excel)	11.08.2007			5		2 200	Петров	Иван А	Александрович			
														~
<														>
	v	Короткие кур	осы										Печа	эть
🛓 Табл	ица												_	. 🗆 X
	Kynci	_ 1												^
	турс													
	Код	Наимено	ование	Дата	а начала	Π	одоля	итель	ность		Стоимость	Пре	подавател	њ
	3	PowerPoint		11/08/07						2	1 20) Петров I	Иван Алекс	сандро
														\sim
	_		_			_	_	_	_	_		_		>
Б Справоч		A	Таблица											
Для получе	ения подска	ізки нажмите I	F1										CAP N	IUM //

Рис. 1.39. Печатная форма справочника с отражением только коротких курсов

Справочник Клиенты

Добавим в нашу информационную базу еще один справочник, который назовем клиенты. В нем будут отражены учащиеся или слушатели наших курсов. Для продолжения практических действий вернемся в конфигуратор и начнем уже известным способом создание нового справочника.

Во-первых, на закладке Данные необходимо внести информацию о реквизитах в соответствии с рис. 1.40. Для реквизитов справочника Адрес и телефоны в качестве типа данных укажем — строка длиной не более 50 символов.



Рис. 1.40. Окно редактирования справочника Клиенты

Во-вторых, с помощью кнопки пиктографического меню в нижней части окна редактирования объекта конфигурации создадим табличную часть с именем Изучаемые Курсы.

В-третьих, в созданной табличной части разместим два реквизита (рис. 1.41) — Курс (тип данных — СправочникСсылка.Курсы) и СтатусУчащегося (тип данных — перечислениеСсылка.Статус). В этом случае при заполнении данных полей в режиме работы 1С:Предприятие будут открываться соответственно форма списка справочника курсы и форма списка перечисления Статус.

Для просмотра и заполнения справочника в режиме работы 1С:Предприятие можно воспользоваться электронными формами ввода, генерируемыми системой по умолчанию. Однако далее в этом разделе нам понадобится разместить на форме кнопку **Печать**, и для того чтобы это сделать, необходима пользовательская форма.

Начнем конструкторские действия. Первое, что требуется, — перейти на закладку **Формы** (рис. 1.42) и сделать установку на редактирование — **В** диалоге. Иначе (при редактировании в форме списка) табличная часть справочника будет недоступна. Можно было также воспользоваться и вариантом **Обоими способами**. Тогда в режиме 1С:Предприятие пользователю предоставляется возможность выбора или формы списка, или формы элемента. Для этого следует использовать кнопку **Действия** (она видна в левом верхнем углу на рис. 1.39).

Примечание

На практике вариант редактирования данных обоими способами при работе со справочниками используется очень часто в связи с его мобильностью — легко перейти от

🚴 Конфигуратор (учеб	ная версия) - Конфигурац	
Файд Правка Конфиги	рация Отдалка Алминистр	ирование Сервис Окна Справка
		· · · · · · · · · · · · · · · · · · ·
🗐 Справочник Клиент	ы	_ 🗗 ×
Основные	Длина кода	: Реквизит табличной ч 🕈 🗙
Иерархия		21 🖾 🍽 🗙 ∨
Владельцы	Длина наименования — Тип кода	▼ Основные:
🕨 Данные	Пипкода	Имя Курс
Нумерация	O Granue	Синоним Курс
Формы	О Строка	Комментари
Макеты	包装准全工具	
Подсистемы		
Ввод на основании	Адрес	
Права	Телефоны	Тип Справочник Ссылка Кирсы
Интерфейсы		справочникосвыха. Курсы 1 ш
Обмен данными	🏦 💁 🔜 🖉 🕇 🔳	
Прочее	— — — — — — — — — — — — — — — — — — —	
	⊡∘Ⅲ ИзучаемыеК	
	Курс	Имя объекта метаданных
	і СтатисУч	
Действ	ия 👻 <Назад Д.	
, Для получения подсказки	нажмите F1	CAP NUM ru-

Рис. 1.41. Добавление реквизитов в табличную часть справочника Клиенты

📩 Конфигуратор (учебн	ная версия) - Конфигура	ция	×
<u>Ф</u> айл <u>П</u> равка Конфигу; Справка	рация Отладка Администр	рирование <u>С</u> ервис <u>О</u> кна	
🗋 🛋 🖬 🐰 🗞 🛱	i	-	-
菌 Справочник Клиенть	k	- 8	×
Основные	Редактирование —	— Выбор — — — — — — — — — — — — — — — — — — —	^
Иерархия	🔘 В списке	🔘 Из формы	
Владельцы	• В диалоге	🔘 Быстрый выбор	
Данные	🔘 Обоими способами	 Обоими способами 	
Нумерация			
🕨 Формы	Псновные формы	аименование, код	
Макеты	Списка		
Подсистемы	Purforea		
Ввод на основании	выоора		
Права	выбора группы		
Интерфейсы	Элемента	X Q	
Обмен данными	Группы	× Q	
Прочее	9 2 9 4 1 1		
	чермы	8	~
Действия 👻	<Назад Далее>	Закрыты Справка	
Для получения подсказки н	нажмите F1	CAP NUM ru-	///

Рис. 1.42. Закладка Формы окна редактирования справочника Клиенты

одной формы к другой. Форма списка концентрированно отражает информацию о многих элементах справочника, но при этом, если реквизитов много, то вид представления информации может быть не очень удобен для просмотра конкретного элемента. Эту проблему снимает форма элемента, которая предоставляет удобную возможность для просмотра, анализа и изменения элементов справочника.

Вернемся к нашей разработке в конфигураторе и щелкнем по изображению лупы в группе **Основные формы** напротив надписи — Элемента, после чего сделаем необходимые установки в следующем окне (рис. 1.43).

Конструктор фо	рмы справочника	×
Выберите тип о	рормы:	7
🔷 Форма спис	ска справочника	
🔷 Форма выб	ора справочника	
💿 Форма выб	ора группы справочника	
• Форма элег	мента справочника	
🔘 Форма груп	пы справочника	
🔘 Произвольн	ная форма	
💌 Назначить 🤄	рорму основной	
Основная ф	орма списка и выбора	
Основная ф	орма з лемента и группы	
Имя:	ФормаЭлемента	
Синоним:	Форма элемента]
Комментарий:		1
🗹 Командная	панель формы сверху	
🗹 Командная	панель формы снизу	
< Ha	азад Далее > Готово Отмена Справка	

Рис. 1.43. Окно конструктора формы справочника

Примечание

Для интерактивной работы пользователя предназначен ряд объектов, из которых наиболее важным является форма. Этот объект используется для доступа к свойствам, методам и событиям формы в целом, а также к коллекции ее элементов управления.

Теперь после щелчка по кнопке Далее перед нами открывается окно, позволяющее скрыть либо отобразить отдельные реквизиты справочника на форме (рис. 1.44). В данном случае следует оставить все созданные нами реквизиты.

После этого осталось нажать кнопку Готово, и в итоге мы увидим на экране сформированную форму элемента справочника клиенты (рис. 1.45).

Теперь, когда форма, необходимая для работы со справочником, готова, можно перейти в режим 1С:Предприятие (с помощью знакомой команды **Начать отладку**) и внести информацию об учащихся в созданный справочник. На рис. 1.46 показан пример заполнения формы элемента справочника информацией об одном из учащихся.

Конструктор формы справочника		×
Располагать элементы управления в	: 1 🗘 колонку(и)	
Реквизиты	Элементы формы	7
г✔ Код	аы Поле ввода	5
— Иаименование	💵 Поле ввода	צי
ПометкаУдаления	🗹 Флажок	
Предопределенный	🗹 Флажок	
✓ Адрес	аы Поле ввода	
🛛 🗸 Телефоны	аы Поле ввода	
🖃 🗸 ИзучаемыеКурсы	Табличное поле	
… ✓ НомерСтроки		
✓ Курс	аы Поле ввода	
	аы Поле ввода	
< Назад Далее >	Готово Отмена Справка	a

Рис. 1.44. Окно для включения реквизитов справочника в форму

🛃 Конфигуратор (учебная версия) - Конфигурация 📃 🗆 🗙
Файл Правка Форма Конфигурация Отладка Администрирование Сервис Окна Справка
I 🗋 🖬 🐰 🖻 🙄 🛞 💥 📕 🔍 🔍
📟 Справочник Клиенты: ФормаЭлемента 📃 🗖 🗙
Клиенты
Действия - 🛃 🐼 😵 🕐 💭
Код:
Наименование
Адрес:
Телефоны:
N Курс Статус учащегося
аь <Поле ввода> аь <Поле ввода>
💭 ОК Записать Закрыть
🗏 Диалог 🗏 Модуль 🚦 Реквизиты
Для получения под САР NUM ги + 1 👘 8, 157 🛛 🕂 384 x 102 //

62

Рис. 1.45. Форма элемента справочника Клиенты в режиме конфигуратора



Рис. 1.46. Форма справочника Клиенты в режиме работы 1С:Предприятие

Перейдем к созданию печатного варианта формы элемента справочника (табличного документа, отражающего информацию, расположенную на форме). Для этого опять воспользуемся возможностями конструктора печати. Мы уже знакомы с техническими действиями для его вызова — в окне редактирования объекта конфигурации (рис. 1.47) требуется перейти на закладку **Макеты**, где следует воспользоваться кнопкой **Конструкторы**.

Примечание

В системе имеются различные конструкторы, облегчающие труд разработчика. Это конструкторы форм констант, справочников, документов, журналов документов, отчетов и других объектов, конструкторы печатных форм, конструкторы движений регистров и др.

В процессе диалога с конструктором укажем, что создаваемая процедура должна располагаться в модуле формы (в данном случае это будет формаЭлемента). В реквизиты шапки перенесем наименование, Адрес и Телефоны, а в реквизиты табличной части — курс и СтатусУчащегося. Подвал табличного документа нам не потребуется. В завершение диалога укажем на необходимость создания кнопки печать на форме. По итогам диалога с конструктором на закладке Макеты мы увидим макет Печать (рис. 1.48).

Как мы уже знаем, параллельно с формированием макета конструктор печати на закладке **Модуль** формы элемента справочника расположил программную процедуру, позволяющую создать табличный документ на основании структуры макета печать и информации, содержащейся в справочнике (листинг 1.3).

🛃 Конфигуратор (учебная	а версия) - Конфигурация 📃 🛛 🗙
<u>Ф</u> айл <u>П</u> равка Конфигурац	ция Отладка Администрирование <u>С</u> ервис
<u>О</u> кна Сп <u>р</u> авка	
1 🗋 🚢 🔛 💥 🙄 🖡	♥ ♥
菌 Справочник Клиенты	_ 🗗 ×
Основные	9
Иерархия	
Владельцы	Макеты
Данные	
Нумерация	
Формы	
🕨 Макеты	
	Конструкторы 👻
Действия 👻 <Назад	Далее> Закрыть Справка
<	>
Для получения подсказки наж	кмите F1 CAP NUM ru т

Рис. 1.47. Закладка Макеты окна редактирования справочника Клиенты

📩 Конфигуратор (учебная версия) - Конфигурация								
<u>Файл Правка</u> Таблица Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна								
Справка								
I 🗋 🛋 🖬	1 X	B B 6	s 🝳 🖕	d [M M		·	•
🛓 Справо	чник І	(лиенты: П	ечать				_ 1	⊐ × □
		1	2		3		4	^
	1							
Заголовок	2		Клиен	ιты				
	3							
Шапка	4							
	5		Наименован	ние	≺Наименование>			
	6		Адрес		<Адрес>			
	7		Телефоны		<Телефоны>		_	
	8							
	9							
Изучаемые	10						-	
	11		Курс		Статус учащегос.	я		
Изучаемые	12		<Курс>	<Стат	усУчащегося≻			
<	40						1	>
Для получен	ия пода	жазки нажм	илте F1			CAP	NUM ru	• //

Рис. 1.48. Макет табличного документа

Листинг 1.3. Процедура печати элемента справочника Клиенты

Процедура Печать (Элемент)

- //{{_КОНСТРУКТОР_ПЕЧАТИ_ЭЛЕМЕНТ(Печать)
- // Данный фрагмент построен конструктором.
- // При повторном использовании конструктора
- // внесенные вручную изменения будут утеряны!!!

```
ТабДок = Новый ТабличныйДокумент;
Макет = Справочники. Клиенты. Получить Макет ("Печать");
// Заголовок
Область = Макет.ПолучитьОбласть ("Заголовок");
ТабДок.Вывести (Область);
// Шапка
Шапка = Макет.ПолучитьОбласть ("Шапка");
Шапка.Параметры.Заполнить (ЭтотОбъект);
ТабДок.Вывести (Шапка);
// ИзучаемыеКурсы
Область = Макет.ПолучитьОбласть ("ИзучаемыеКурсыШапка");
ТабДок.Вывести (Область);
ОбластьИзучаемыеКурсы = Макет.ПолучитьОбласть ("ИзучаемыеКурсы");
Для Каждого ТекСтрокаИзучаемыеКурсы Из ИзучаемыеКурсы Цикл
    ОбластьИзучаемыеКурсы.Параметры.Заполнить (ТекСтрокаИзучаемыеКурсы);
    ТабДок.Вывести (ОбластьИзучаемыеКурсы);
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Ложь;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать ();
//}} КОНСТРУКТОР ПЕЧАТИ ЭЛЕМЕНТ
КонецПроцедуры
```

Здесь новым моментом является использование цикла перебора строк табличной части элемента справочника:

Для Каждого ТекСтрокаИзучаемыеКурсы Из ИзучаемыеКурсы Цикл

Такой цикл позволяет заполнить табличную часть печатной формы элемента справочника курсы. В остальном же строки созданной конструктором процедуры уже знакомы нам по листингу 1.2.

Примечание

Оператор цикла Для Каждого предназначен для циклического обхода коллекции значений. При каждой итерации цикла возвращается новый (очередной) элемент коллекции. Цикл продолжается до окончания перебора всех элементов.

На рис. 1.49 показан пример формирования печатной формы элемента справочника курсы в режиме 1С:Предприятие.

Теперь уже можно подвести небольшой промежуточный итог и отметить, что этот раздел позволил нам получить первые практические навыки по работе со справочниками — объектами конфигурации, которые очень часто используются на практике для хранения достаточно статичной информации.

В большинстве ситуаций требуется фиксирование информации о событиях, происходящих в подразделении либо в организации в целом. Для этого в системе 1С:Предприятие 8.1 существуют другие объекты, среди которых в первую очередь следует отметить документы. В следующем разделе мы с ними начнем знакомиться.

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🗖 🗙
📱 <u>Ф</u> айл <u>П</u> равка Таблица Операции <u>О</u>	ервис <u>О</u> кна Сп <u>р</u> авка
🗋 🛁 🖌 🍡 🛍 🖾 🔍 🐥	I 🎹 🛅 🧝 M M+ M- 🎇 🖕
🛕 Таблица	_ 🗆 ×
Клиенты	^
Наименование Сидоров Адрес Н.Новгор Телефоны 123-45-6	Евгений Петрович юд, Минина 101-7 7
Курс	Статус учащегося
PowerPoint	Зачислен
Базы данных	Зачислен
Информатика (Excel)	Зачислен
<[<]	×
🎒 Справочник Клиенты 📄: Сидоров	Евгений 🚂 Таблица
Для получения подсказки нажмите F1	CAP NUM

Рис. 1.49. Печатная форма элемента справочника Клиенты

Документы

Сразу отметим, что документы относятся к наиболее востребованным разработчиками объектам конфигурации. Как уже говорилось, они предназначены для фиксирования информации о происходящих событиях в подразделении или в организации в целом.

Документ Внесение оплаты

Наша задача в этом разделе — познакомиться с организацией документов и некоторыми техническими приемами их разработки в конфигураторе. Здесь мы создадим документ, который будет фиксировать информацию о текущей оплате за обучение. И первое, уже знакомое действие на этом пути заключается в щелчке правой кнопкой мыши на строке Документы в окне Конфигурация и выборе пункта Добавить в открывшемся контекстном меню, после чего начинается несложная работа с конструктором документа. В качестве имени создаваемого объекта конфигурации введем ВнесениеОплаты (рис. 1.50). Этот документ будет предназначаться для фиксации внесенной оплаты за обучение конкретным учащимся, определяемой набором курсов, которые он посещает.

Примечание

Документ — одно из основных понятий системы 1С:Предприятие [8]. Каждый объект данного типа содержит информацию о конкретной хозяйственной операции и характе-

ризуется своим номером, датой и временем. Дата и время создания документа позволяют установить строгую временную последовательность совершения операций. В конфигураторе создается только структура документа, а конкретные экземпляры документов вводятся в информационную базу при работе в режиме 1С:Предприятие пользователем. В дальнейшем для краткости вместо словосочетания "структура документа" будет использоваться просто слово "документ".

📩 Конфигуратор (учебная	я версия) - Конфи	гурация		
<u>Ф</u> айл <u>П</u> равка Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна				
Справка				
I 🗋 🛋 🖬 X 🖻 🛍	5 C 🖸 🖻	24 24	» •	
🐱 Документ ВнесениеО	платы		_ 🗆 ×	
• Основные		-		
Данные	Имя:	ВнесениеОплаты		
Нумерация	Синоним:	Внесение оплаты		
Движения	Комментарий:			
Последовательности				
Журналы				
Формы				
Макеты				
Действия 🗸 (Назад	Далее>	Закрыть Сг	равка	
Для получения подсказки нах	кмите F1	CAP NUM	ru • //	

Рис. 1.50. Окно редактирования документа ВнесениеОплаты

Заполнение закладки Данные

На рис. 1.51 показана закладка Данные окна редактирования объекта конфигурации, на которой необходимо определить поля создаваемого документа. Заметим, что любой документ имеет шапку — общую или заголовочную часть, а также одну или несколько табличных частей. В шапке размещается информация, общая для всего документа в целом.

Для создаваемого нами документа ВнесениеОплаты в области шапки разместим два реквизита — клиент и телефоны. В качестве типа данных первого из них установим СправочникСсылка.Клиенты, а для другого выберем строковый тип данных — строка не более 50 символов.

С помощью приведенной далее процедуры (см. листинг 1.4) программным способом обеспечивается автоматическое заполнение поля телефоны, когда пользователь в режиме работы 1С:Предприятие вносит информацию в поле клиент (т. е. выбирает необходимого клиента из справочника).

Перейдем теперь к разработке табличной части документа. Следует заметить, что большинство реальных документов содержит один или несколько перечней каких-либо данных (строк или элементов). Это может быть список сотрудников, список номенклатуры или что-то другое. Для отображения подобной информации в объект конфигурации "*документ*" заложена возможность создания табличных частей.



Рис. 1.51. Структура документа ВнесениеОплаты

Примечание

Табличные части — это также объекты конфигурации, но они являются подчиненными основным объектам. С ними мы уже встречались при работе со справочниками. И в том, и в этом случае табличные части подчинены основным объектам (элементу справочника или документу). В плане программирования для объекта *"табличная часть"* существует многообразие свойств и методов работы с ними.

На практике для документов с несложной структурой используется одна табличная часть, но в типовых конфигурациях, созданных программистами фирмы "1С", вы найдете немало документов с включением нескольких таких частей.

После такого комментария вернемся к практике — в разрабатываемом документе внесениеОплаты создадим табличную часть, которую назовем ОплатаКурсов, и разместим в ней два реквизита:

- ♦ Курс (ТИП Данных СправочникСсылка.Курсы);
- ВнесеннаяСумма (ТИП данных Число).

Форма документа

После определения структуры документа перейдем к разработке интерфейса пользователя для работы с документом — к созданию его экранной формы. Для

этого в окне редактирования объекта конфигурации (рис. 1.52) выберем закладку Формы и приступим к созданию формы документа. В данной ситуации пользоваться формой, генерируемой системой 1С:Предприятие 8.1 по умолчанию, мы не будем.

В дальнейшем нам придется добавлять в разработку собственные программные процедуры, поэтому и необходимо создать пользовательскую форму. Для этого щелкнем по изображению лупы в группе **Основные формы** напротив надписи Документа (рис. 1.52) и перейдем к работе с конструктором документа.

📩 Конфигуратор (учебная в	ерсия) - Конфигурация 📃 🗵 🗙
🛛 <u>Ф</u> айл <u>П</u> равка Конфигурация	Отладка Администрирование <u>С</u> ервис <u>О</u> кна
Справка	
I D 🛋 🖬 🐰 🖻 🛍 🖉	i 🗋 5 c 🗿 A 🚬
🐱 Документ ВнесениеОпл	аты _ 🗆 🗙
Основные	_ Основные формы
Данные	Списка
Нумерация	Buttona
Движения	
Последовательности	Документа
Журналы	
▶ Формы	ввод по строке Номер х
Макеты	'열롱'열忄부름
Подсистемы	е Формы
Ввод на основании	
Права	
A 🔻	
Действия 🔹 <Назад	Далее> Закрыть Справка
Для получения подсказки нажмі	urre F1 CAP NUM ru▼ //

Рис. 1.52. Закладка Формы окна редактирования документа

В первом появившемся окне диалога следует сделать установки (реально они должны быть выполнены по умолчанию) в соответствии с рис. 1.53. После этого, нажав кнопку Далее, мы увидим следующее окно конструктора (рис. 1.54), позволяющее отобразить (либо исключить) на форме определенные реквизиты документа.

Работа в режиме диалога с конструктором формы документа завершается нажатием кнопки **Готово**, что приводит к отображению на экране формы документа с автоматически созданными на ней элементами управления (рис. 1.55).

На этом работа с конструктором документа в конфигураторе завершена, и теперь требуется перейти в режим 1С:Предприятие, где и следует заполнить несколько документов созданного вида. Один из вариантов внесения информации в документ показан на рис. 1.56.

После заполнения полей документа его следует сохранить и закрыть (все это выполняется нажатием кнопки **OK**). В дальнейшем данный документ (как и все

Конструктор фор	эмы документа	×
Выберите тип ф Форма спис Форма выбо Форма доку Произвольна Иазначить ф Основная фо	юрмы: ка документа ра документа мента ая форма орму основной орму основной	
Имя:	ФормаДокумента	
Синоним:	Форма документа	1
Комментарий:		
Командная п Командная п	анель формы сверху анель формы снизу ізад Далее > Готово Отмена Справка	

Рис. 1.53. Окно конструктора формы документа

Конструктор формы документа		×
Располагать элементы управления в	а 🚺 🗘 колонку(и)	
Реквизиты	Элементы формы	A
г ✓ Номер	аы Поле ввода	- -
	аы Поле ввода	
Проведен	🗹 Флажок	
ПометкаУдаления	🗹 Флажок	
✓ Клиент	аы Поле ввода	
— Гелефоны	аы Поле ввода	
🖻 🔽 ОплатаКурсов	Табличное поле	
НомерСтроки		
	аы Поле ввода	
У ВнесеннаяСумма	аы Поле ввода	
< Назад Далее >	Готово Отмена Справ	ка

Рис. 1.54. Определение отображаемых на экранной форме реквизитов

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🗙
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование
<u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
D 📣 🖬 🔏 🕼 🖆 🖒 🕐 м 🛛 🎽 🎇
📓 Документ ВнесениеОплаты: ФормаДокумента 💦 💶 🗙
Внесение оплаты
Действия 🗸 🛃 🔯 🏂 👔
Номер:
Дата::Е
Клиент:
Телефоны:
╘╘ ぇ Ҷ晶↑↓₿₿₿₩
N Курс Внесенная сумма
аы <Поле ввода> аы <Поле ввода>
индуль стехвизиты
Для получения подс САР NUM ги т +3, 157 🕂 384 x 102 //

Рис. 1.55. Форма документа ВнесениеОплаты в режиме конфигуратора

🏪 1С:Предп	риятие (учебная версия) - Конфигурация 📃 🔍
<u>∎</u> айл <u>П</u> рав	зка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
i 🗋 📫 冒	X 🖻 🛍 🍜 🖻 🎽 🎹 🛅 🦧 M M+ M- 🎇 🧉
🐱 Внесені	ие оплаты: Внесение оплаты (Новый) * 💦 💶 🗙
Действия 🕶	🛃 🔤 📑 🛃 🤶
Номер:	00000001
Дата:	31.08.2008 0:00:00
Клиент:	Сидоров Евгений Петрович 🗙
Телефоны:	
e te a	🕻 🔄 🖦 🕇 ↓ 戡
N Курс	с Внесенная сумма
1 Баз	ы данных
	ОК Записать Закрыть
📇 Документь	ы Внесение опла 📴: Внесение оплаты (Новы
Для получени	я подсказки нажмите F1 САР NUM

Рис. 1.56. Почти заполненная форма документа ВнесениеОплаты в режиме 1С:Предприятие

другие, созданные таким же образом) можно всегда найти (при работе в режиме 1С:Предприятие) в списке, который открывается при выборе в разделе Документы названия документа — Внесение оплаты.

Программирование формы документа

Теперь обеспечим небольшую автоматизацию работы, о которой мы уже упоминали ранее. В созданном документе два поля, Клиент и телефоны, связаны между собой. А именно, при заполнении поля клиент значением из справочника клиенты хотелось бы, чтобы соответствующий телефон автоматически фиксировался в поле телефоны. Наличие телефона в шапке документа удобно, т. к. в этом случае уже по документу видно, как с человеком связаться, а автоматизация заполнения этого поля приведет к ускорению работы и уменьшению ошибок ввода. Для такой автоматизации мы и напишем короткую процедуру.

Итак, в окне редактирования формы документа (режим конфигуратора) необходимо двойным щелчком по полю клиент (рис. 1.57) открыть окно его свойств. Затем напротив имени события приизменении следует щелкнуть по пиктограмме с изображением лупы, в результате чего мы попадем в процедуру, автоматически выполняемую при выборе клиента в поле клиент документа. Это фактически пока еще только заготовка нашей будущей процедуры, которую мы напишем. В дальнейшем ее всегда можно найти на закладке **Модуль**, где размещаются все программные процедуры, работающие с экранной формой документа.

🛃 Конфигуратор (учебная версия) - Конфигурация	
📱 🕂 дайл Правка Форма Конфигурация Отладка Администрирование Сервис Окна Справка	
I 🗋 🖬 🔏 📾 📾 🖻 🖕 🖬 🕷 🔍 🔻 🕷 🐥 🐥	
🧮 🚟 Документ ВнесениеОплаты: Форма	Свойства: Поле ввода 🛛 🕂 🗙
Внесение оплаты	2∔ 🙀 🍽 🗙 ∨
Действия - 🛃 🔃 🏂 🚹 ? 🤅	Имя Клиент
Номер:	Данные Клиент
Дата:	ТипЗначения СправочникСсылка.Клиенть 🔻 …
Клиент:	ИзменяетДанні
Телефоны:	ПропускатьПри
열 철 🗟 철 🖬 🕇 🖊 🛱 🔛	ПервыйВГруппа Видимость 🔽
N Курс Внесенная	Доступность 🔽
аы «Поле ввода» аы «Поле	АвтоКонтекстн
	КонтекстноеМе Отсутствует 📃
К Диалог Модуль Реквизиты	Имя, по которому будет выполняться обращение к элементу управления в алгоритме модуля
Для получения подсказки нажмите F1 CAF	Имя, Name

Рис. 1.57. Окно свойств поля Клиент

Сам текст процедуры, которую необходимо написать для решения поставленной задачи, приведен в листинге 1.4. В результате выполнения программного кода
в режиме 1С:Предприятие после выбора клиента поле телефоны в шапке документа заполняется данными о телефонах из справочника клиенты, в чем можно убедиться.

Листинг 1.4. Процедура, выполняемая при заполнении поля Клиент

Процедура КлиентПриИзменении(Элемент) Телефоны = Клиент.Телефоны; КонецПроцедуры

Продолжим работу по усовершенствованию нашего документа. Следующая цель — обеспечить автоматическое заполнение его табличной части информацией о курсах после выбора клиента. Для этого необходимо внести в уже написанную процедуру клиентПриИзменении новые программные конструкции, и в листинге 1.5 показана измененная процедура.

Листинг 1.5. Процедура заполнения табличной части документа ВнесениеОплаты

```
Процедура КлиентПриИзменении(Элемент)

Телефоны = Клиент.Телефоны;

ОплатаКурсов.Очистить();

// Перебор строк табличной части документа

Для Каждого ТекущаяСтрока Из Клиент.ИзучаемыеКурсы Цикл

НоваяСтрока = ОплатаКурсов.Добавить();

НоваяСтрока.Курс = ТекущаяСтрока.Курс;

КонецЦикла;

КонецПроцедуры
```

На рис. 1.58 показан результат выполнения процедуры в режиме 1С:Предприятие — после выбора учащегося в поле клиент его телефон перенесен в шапку документа, а табличная часть документа автоматически заполнена названиями посещаемых им курсов.

Разберем теперь непосредственно текст листинга 1.5. В начале процедуры после заполнения поля телефоны к табличной части документа (ОплатаКурсов) применяется метод Очистить (). Если этого не сделать, то табличная часть будет дополняться названиями курсов при последовательных щелчках по полю клиент. Таким образом, в процедуре из листинга 1.5 она сначала очищается, а затем заполняется вновь. В цикле, начинающемся со строки

```
Для Каждого ТекущаяСтрока Из Клиент.ИзучаемыеКурсы Цикл
```

последовательно перебираются все строки табличной части справочника клиенты, и при каждом проходе цикла в табличную часть документа добавляется один из курсов, на которые записался учащийся. Для этого используется метод добавить (). В результате пользователь, выбрав в форме документа конкретного учащегося, автоматически получает заполненную названиями посещаемых им курсов табличную часть документа, и остается только внести данные об оплаченных суммах (рис. 1.59).

🏪 1С:Предп	риятие (учебная версия)	- Конфигур	ация				
<u>∎ Ф</u> айл <u>П</u> рав	зка Операции <u>С</u> ервис <u>О</u> кн	а Сп <u>р</u> авка					
I 🗋 🚄 冒	🔏 🖻 🖻 🧉 💐 !	III 🗎 🏅	M M+	м- 🎇 🖕			
🐱 Внесені	ие оплаты: Внесение опла	аты (Новыі	a) *	_ 🗆 ×			
Действия 🕶	📴 🔯 🔹 😤)					
Номер:	000000001						
Дата:	31.08.2008 0:00:00						
Клиент:	Сидоров Евгений Петрович			x			
Телефоны:	123-45-67						
엄험공	1 1 1 1						
N Курс		Внесенная	сумма				
1 Pow	erPoint						
2 basi	ы данных						
ЗИНФ	орматика (Ехсеі)						
		OK	Записать	Закрыты			
📇 Документы Внесение опла 🐱: Внесение оплаты (Новы							
Для получени	я подсказки нажмите F1		CAP	NUM //			

Рис. 1.58. Автоматическое заполнение телефона и табличной части документа после выбора клиента

🏪 1С:Предп	риятие (учебная версия)	Конфигур	ация	<u>- 🗆 ×</u>				
<u>∎ Ф</u> айл <u>П</u> рав	зка Операции <u>С</u> ервис <u>О</u> кн	а Справка						
I 🗋 🚢 📟	X 🖻 🖻 🧉 🖉 🖡 I	III 🗎 🏭	М м+	м- 🎇 🖕				
🐱 Внесени	е оплаты: Внесение опла	аты (Новый	a) *	_ 🗆 ×				
Действия 🕶	🛃 🗷 💺 🚰 🛃 🤶)						
Номер:	000000001							
Дата:	31.08.2008 0:00:00							
Клиент:	Сидоров Евгений Петрович			x				
Телефоны:	123-45-67							
엄벌글	N 🔄 🔜 🌓 🗍 🛃 🛃							
N Курс	;	Внесенная	сумма					
1 Pow	erPoint			500				
2 5ası	ы данных			900				
3 Инф	орматика (Excel)			500				
4 VBA				870				
		UK	Записать	Закрыты				
🎦 Документы Внесение опла 🐱: Внесение оплаты (Новы								
Для получени	я подсказки нажмите F1		CAP	NUM //				

Рис. 1.59. Заполненный документ ВнесениеОплаты в режиме 1С:Предприятие

Примечание

При необходимости в автоматически заполненную табличную часть документа ВнесениеОплаты можно добавить дополнительно еще одну или несколько строчек. На рис. 1.59 показана ситуация, когда в табличную часть внесена еще одна строка с помощью стандартной кнопки пиктографического меню.

Создание печатной формы документа ВнесениеОплаты

Печатные формы справочников создавались с помощью конструктора печати, и его же мы используем при разработке печатной формы документа ВнесениеОплаты. В листинге 1.6 показана созданная конструктором программная процедура, реализующая данную функцию.

Листинг 1.6. Процедура печати документа ВнесениеОплаты

```
Процедура Печать (Элемент)
//{ { КОНСТРУКТОР ПЕЧАТИ ЭЛЕМЕНТ (Печать)
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора внесенные вручную
// изменения будут утеряны!!!
ТабДок = Новый ТабличныйДокумент;
Макет = Документы.ВнесениеОплаты.ПолучитьМакет ("Печать");
// Заголовок
Область = Макет.ПолучитьОбласть ("Заголовок");
ТабДок.Вывести (Область);
// Шапка
Шапка = Макет.ПолучитьОбласть ("Шапка");
Шапка.Параметры.Заполнить (ЭтотОбъект);
ТабДок.Вывести (Шапка);
// ОплатаКурсов
Область = Макет.ПолучитьОбласть ("ОплатаКурсовШапка");
ТабДок.Вывести (Область);
ОбластьОплатаКурсов = Макет.ПолучитьОбласть ("ОплатаКурсов");
Для Каждого ТекСтрокаОплатаКурсов Из ОплатаКурсов Цикл
    ОбластьОплатаКурсов.Параметры.Заполнить (ТекСтрокаОплатаКурсов);
    ТабДок.Вывести (ОбластьОплатаКурсов);
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Ложь;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать ();
//}} КОНСТРУКТОР ПЕЧАТИ ЭЛЕМЕНТ
КонецПроцедуры
```

Большинство программных конструкций нам уже знакомо, поэтому какого-то дополнительного комментария не требуется. В табл. 1.1 отражены основные свойства и методы табличного документа.

Название свойства (метода)	Описание
ОтображатьСетку	Позволяет отключить отображение сетки в табличном документе (при значении ложь) либо, наоборот, его включить (при значении Истина)
Защита	Позволяет допустить копирование либо ручное изменение таб- личного документа (при значении дожь) либо, наоборот, предот- вратить эти действия (при значении Истина)
ТолькоПросмотр	Дает возможность установить только режим просмотра (при значении Истина). В этом случае в табличном документе будут недоступны для редактирования ячейки, у которых установлено свойство Защита
ОтображатьЗаголовки	Включает режим отображения (при значении Истина) заголовков колонок и строк таблицы. Заголовки содержат номера столбцов и строк и выводятся вверху и слева от таблицы
Показать ()	Метод, открывающий окно показа и редактирования табличного документа

Таблица 1.1. Свойства и методы табличного документ	ıa
--	----

Список документов

Следует упомянуть еще об одном важном моменте. Мы все время говорили о текущем документе, с которым в настоящий момент работаем.

📙 1 С:Предприятие (учебная версия) - Конфигурация 💶 🗵 🗙									
<u>Файл Правка Операции Сервис Окна</u> Справка									
I 🗋 🛁 Х 🖻 🖷 🕌 🖄 🗢 🕐 📱 🎆 🧮 🧝 и м. м+ м- 🎘 🖕									
🚢 Документы Внесен	ие оплаты		_ 🗆 ×						
Действия 🕶 🔄 芸 ⊰	ka (⇔)	11 🚡 🖷 - 🏹 🖸 ?)						
Дата 🔻	Номер	Клиент	Телефоны 🛆						
31.08.2008 11:48:15	000000001	Сидоров Евгений Петрович	123-45-67						
31.08.2008 11:48:25	000000002	Сидоров Евгений Петрович	123-45-67						
☑ 31.08.2008 11:48:32	000000003	Сидоров Евгений Петрович	123-45-67						
<			>						
— Документы Внесение опла									
Для получения подсказки н	ажмите F1		CAP NUM						

Рис. 1.60. Список введенных документов ВнесениеОплаты

Может возникнуть вопрос: как его потом найти? Все созданные документы никуда не пропадают — они остаются в информационной базе. И если в режиме 1С:Предприятие обратиться к разделу Документы, то после выбора названия документа Внесение оплаты перед нами откроется список всех созданных ранее документов данного вида (рис. 1.60).

Отчеты

Выполненные до сих пор действия по разработке и заполнению документов и справочников позволяют в основном зафиксировать информацию, однако быстро ответить на естественные вопросы — кто записался на тот или иной курс, какая сумма внесена учащимися за конкретный курс, кто и сколько заплатил за обучение — мы пока не можем. Точнее, если мы самостоятельно аккуратно проанализируем справочники и уже созданные документы, то ответы на подобные вопросы мы, конечно, получим. Однако и время для этого может потребоваться весьма значительное.

Можно сказать, что получение информации в удобной форме и является одним из основных достоинств системы 1С:Предприятие. При сборе и накоплении данных важнейшую роль играют объекты под названием *"регистры"*. В последующих главах мы разберем технические приемы работы с регистрами сведений и с регистрами накопления. Фактически они предназначены для фиксирования и аккумулирования информации о разнообразных фактах и событиях.

Здесь же мы обойдемся без них и рассмотрим построение отчета по информации непосредственно из введенных документов внесениеОплаты. Однако когда количество обрабатываемых документов становится большим, то использование регистров значительно более эффективно по времени получения результата. При составлении отчета извлечение уже подготовленной информации из регистров сведений выполняется существенно быстрее по сравнению с перебором всех исходных документов. Далее (в *славах 3* и 4) мы поговорим и о других преимуществах регистров в плане представления отчетной информации.

Среди объектов конфигурации существует *отчет*, и этим объектом мы воспользуемся для получения ответа на вопрос, какая сумма внесена учащимися за конкретный курс. Первый уже знакомый шаг — в окне **Конфигурация** необходимо щелкнуть правой кнопкой по строке **Отчеты** и начать процесс создания нового отчета — учетоплаты (рис. 1.61).

Ранее мы пользовались готовыми макетами, которые создавал конструктор печати, а теперь создадим макет сами.

Примечание

Отчет — это объект конфигурации, который используется для получения выходной (сводной) информации на основании введенных данных. Данный объект конфигурации включает в себя алгоритм получения и обработки информации, форму и макет. С помощью форм организуется необходимый интерфейс, а макеты позволяют оформить полученную информацию в виде табличного документа желаемого вида.

📩 Конфигуратор (учебная версия) - Конфигурация 💶 🖂									
<u> Ф</u> айл <u>П</u> равка Ко	Файл Правка Конфигурация Отладка Администрирование Сервис Окна								
Справка									
I D → U X Pa @ → C M +									
🖤 Отчет УчетОпл	аты		_ 🗆 ×						
• Основные	14								
Данные	Имя:	9четОплаты							
Формы	Синоним:	Учет оплаты							
Макеты	Комментарий:								
Подсистемы									
Права									
Интерфейсы									
Прочее	Поновная схема								
	o chobildy excisit								
		Открыть схему компоновки д	анных						
Действия 👻	<Назад	Далее> Закрыть (Справка						
Для получения подсказки нажмите F1 [CAP] NUM ги -									

Рис. 1.61. Окно редактирования отчета УчетОплаты

Разработка макета

Для этого следует перейти на закладку **Макеты**, а затем с помощью пиктограммы **Добавить** (из пиктографического меню) начать работу с конструктором макета. Сам макет, как мы уже знаем, необходим для формирования табличного документа.

Единственный этап диалога, в котором следует сделать установки, показан на рис. 1.62. После этого перед нами открывается окно (рис. 1.63) для разработки структуры табличного документа. Здесь мы можем самостоятельно создать макет для печатной формы. Дальнейшие наши действия в этом разделе будут связаны с ручным заполнением данного макета — без участия конструктора печати.

Для вывода в табличный документ строк различного вида в макетах предусмотрен механизм секций. В этом случае табличный документ формируется не сразу целиком, а поэтапно — по мере заполнения секций. Секция является компонентой (составляющей табличного документа). Сами секции могут быть вертикальными и горизонтальными, но в рассматриваемом здесь примере мы будем использовать только горизонтальные секции. В реальной работе они применяются значительно чаще. Горизонтальная секция представляет собой совокупность нескольких строк и состоит как минимум из одной строки.

Создадим первую секцию, для чего следует замаркировать мышью названия заголовков первых трех строк. После этого в меню **Таблица** требуется выбрать пункт **Имена**, а затем — **Назначить имя**. Первой секции дадим осмысленное имя — Заголовок, и в одну из ячеек введем текст "Отчет по оплатам за курсы" (рис. 1.64).

Конструктор ма	кета	×						
Имя:	Имя: Макет							
Синоним:	Синоним: Макет							
Комментарий:								
Комментарии.								
_ Выберите тип №	акета:							
💿 Табличный д	цокумент							
🔘 Текстовый д	цокумент							
🔘 Двоичные да	анные							
 Active docum 	nent							
🔵 HTML докум	ент							
🔘 Географичес	ская схема							
🔵 Графическая	я схема							
🔿 Схема компо	рновки данных							
🔘 Макет офорг	мления компоновки данных							
Загрузить из фа	йла:							
	Готово Отмена Справка							

Рис. 1.62. Окно конструктора макета

🛃 Конфигурат	гор (учебная і	зерсия) - Ко	нфигураци	ទេ							
<u>Ф</u> айл <u>П</u> равка Таблица Конфигурация Отладка Администрирование											
Сервис Окна Справка											
I D → 🖬 🔏 📾 🖾 👌 👌 🖓 🛤 🛛 🐥											
🛓 Отчет Уче	етОплаты: Ма	кет			_ 🗆 ×						
1	2	3	4	5	6 ^						
2											
3											
4											
5											
6											
7											
8											
9											
10											
					`						
<					>						
Для получения г	подсказки нажм	ите F1		CAP N	UM ru - //						

Рис. 1.63. Окно структуры табличного документа

Для увеличения (или в общем случае — изменения) размера шрифта текста следует щелчком правой кнопкой мыши открыть окно свойств ячейки с текстом (в данном случае на пересечении второй строки и второго столбца). Это позволит нам, используя раздел **Оформление**, изменить и вид шрифта, и размер букв (рис. 1.65).

🛃 Конфигур	атор	(учебная в	ерсия) - Ко	нфигураци	я	_		
🛛 🛨 айл Правка Таблица Конфигурация Отладка Администрирование Сервис								
<u>О</u> кна Сп <u>р</u> а	вка							
I 🗋 🚢 🖬	I 🗋 🔟 🖬 🖄 🖴 🙆 🖕 🖕 🙀 🛤 🛛 🐥							
🛓 Отчет У	ІчетОг	ілаты: Мак	ет			_	□ × □	
		1	2	3	4	5	^	
Заголовок	1							
	2		Отчет	по оп	атам :	за курс	ы	
	3							
	4							
	5							
	6						_	
Для получения подсказки нажмите F1 CAP NUM ru ▼								

Рис. 1.64. Создание секции Заголовок

Основное содержание отчета будет представлять собой таблицу, но сначала мы должны вывести в табличный документ подписи для столбцов таблицы. Для этого потребуется создать в четвертой строке секцию шапка с содержимым, показанным на рис. 1.66.

Добавим в секцию Шапка небольшое оформление — выполним обводку линиями ячейки, где написано курс, что производится с помощью опций окна свойств ячейки (см. рис. 1.66). Аналогичную обводку следует сделать и для ячейки с текстом Оплаченная сумма.

Теперь можно перейти к созданию третьей, последней в рассматриваемом макете секции Строка (рис. 1.67). От нас также потребуется ввести в ячейки — курс и оплаченнаяСумма, однако для них в разделе Заполнение окна Свойства следует сделать установку — параметр. В этом случае в ячейке будет напечатан не текст, а значение переменной, которая либо явно определена, либо неявно известна в программной процедуре заполнения табличного документа по шаблону макета.

🛃 Конфигу	ратор	(учебная в	ерсия) - Ко	нфигурация	
<u> Ф</u> айл <u>П</u> ра	авка Та	аблица Кон	фигурация () ^т Свойства: Ячейки	₽ X
<u>О</u> кна Сп <u>р</u>	авка			21 © 1 ⊨ × v	
1 🗋 🛋 🖬	X	Pa (a) d	á 💁 🖕	d Ячейки	-
	lluoz0.	nasru Mar		ПоВыделеннымКолонка	м 🗌 ്
	94610	1. 1	2	Вертикальн Низ	-
Заголовок	1			Ориентация	0
	2		Отчет	Отступ	0
	3			АвтоОтступ	0 🗘 🗖
	4			• Оформление:	
	5			Шрифт Arial, 16	×
	6				v
<	1 7	1		Шрифт текста Шрифт Font	
Для получен	ия подс	казки нажм	ите F1	aprint () ton	

Рис. 1.65. Изменение оформления текста в ячейке макета

	🛵 Конфигур	оатор	(учеб	бная версия) - Конфигур	ация			_ 0	×
	<u>Ф</u> айл <u>П</u> рав	эка Та	аблиц	а Конфигурация Отладка	Администриров	ание <u>С</u> ервис <u>О</u>	кна Сп <u>р</u> авка		
	🛓 Отчет У	ІчетОі	ілать	ы: Макет				_ 🗆	×
l			1	2	3				
I	Заголовок	1		_		91 🖬 🏹 🕽	< √		
I		2		Отчет по опла	там за ку				
I		3				лчеики	-		
I	Шапка	4		Курс	Оплаченная суг	ЦветУзора	Авто	X	
I		5				Узор	Нет заливки	-	
I		6							
I		7				ГраницаСлева	— Сплошная	▼	
I		8				ГраницаСверу	— Сплошная	T	
I		9					Сплошпая		
I		10				І раницаСпрає	— Сплошная	▼	
1		11				ГраницаСнизу	— Сплошная	•	
I		12				Обвести	Нет линии	•	
I		13							\sim
		14				Текст в ячейке			
I		15				Текст, Text			
	Для получени	я подс	казки	и нажмите F1					

Рис. 1.66. Добавление секции Шапка

Тем самым фактически мы разделили макет (заготовку для печатной формы) на три части — заголовок, шапку и строку.

Примечание

Создание в шаблоне трех секций является типичным при формировании несложных отчетов. Когда требуется бо́льшая информативность, то количество секций увеличивается. Например, на практике часто встречается так называемая секция *Подвал*, отражающая нижнюю часть табличного документа — в ней подводятся итоги по отчету.

📩 Конфигуратор (учебная версия) - Конфигурация 💶 🗖 🗙									
<u>Ф</u> айл <u>П</u> равка Таблица Конфигурация Отладка Администрирование <u>С</u> ервис									
Окна Справка									
🛯 🗋 🖬 🔏 🖾 🖕 🖕 🛤 👋 🙄									
🛓 Отч	нет У	ІчетОі	плат	гы: Макет		_	□ × □		
			1	2	3	4	(^		
Заголо	BOK	1							
		2		Отчет по с	оплатам за к	урсы			
		3							
Шапка		4		Курс	Оплаченная сумма				
Строка	1	5		<Курс>	<ОплаченнаяСумма>				
		6							
		7							
		8					~		
Для пол	учени	ія подс	казі	ки нажмите F1	C	AP NUM	ru ▼ //,		

Рис. 1.67. Структура макета для табличного документа

Процедура заполнения отчета

Теперь на очереди разработка программной процедуры, которая заполняет табличный документ в соответствии с данным макетом. Для этого нас не устроит форма по умолчанию, и мы создадим свою.

Поэтому перейдем с закладки **Макеты** на закладку **Формы** (рис. 1.68) и щелчком по кнопке с изображением лупы начнем диалог с конструктором формы отчета (рис. 1.69).

🛃 Конфигуратор (у	чебная версия) - Конфигурация	
<u>Ф</u> айл <u>П</u> равка Кон	фигурация Отладка Администрирование	<u>С</u> ервис <u>О</u> кна
Справка		
I 🗋 📫 🖬 🐰 🖣	ы 🛍 🥌 🙆 🖕 с 🚺 м 👘 👘	» •
🕸 Отчет УчетОпл	аты	_ 🗆 ×
Основные	Основная форма отчета:	
Данные		
🕨 Формы		
Макеты	Основная форма настроек:	
Подсистемы		x Q
Права	包装汽车工具	
Интерфейсы	Рормы	
Прочее		
Действия 👻 <i< td=""><td>Назад Далее> Закрыть</td><td>Справка</td></i<>	Назад Далее> Закрыть	Справка
Для получения подска	азки нажмите F1 САР М	NUM ru 🕶 🎵

Рис. 1.68. Закладка Формы окна редактирования отчета

Следует проконтролировать установки, показанные на рис. 1.69, после чего нажать кнопку **Готово**, что позволит увидеть на экране следующее окно (рис. 1.70). Здесь отображена экранная форма отчета с элементами управления, созданными на ней конструктором.

Теперь нужно на закладке **Модуль** открыть окно редактора для написания программных процедур, в котором уже имеется заготовка для процедуры, обслуживающей нажатие на кнопку **Сформировать**. На этот раз мы самостоятельно напишем ее текст, генерирующий отчет (листинг 1.7).

Листинг 1.7. Процедура формирования отчета

```
Процедура КнопкаСформироватьНажатие (Элемент)
```

```
Запрос = Новый Запрос;
```

```
Запрос.Текст = "ВЫБРАТЬ
```

```
| ВнесениеОплатыЗаКурсы.Курс КАК Курс,
```

```
| СУММА (ВнесениеОплатыЗаКурсы.ВнесеннаяСумма)
```

| КАК ОплаченнаяСумма

```
I N3
| Документ.ВнесениеОплаты.ОплатаКурсов
| КАК ВнесениеОплатыЗаКурсы
| СГРУППИРОВАТЬ ПО
| ВнесениеОплатыЗаКурсы.Курс";
Результат = Запрос.Выполнить ().Выбрать ();
ТабДок = Новый ТабличныйДокумент;
Макет = Отчеты. УчетОплаты. Получить Макет ("Макет");
// Заголовок
Область = Макет.ПолучитьОбласть ("Заголовок");
ТабДок.Вывести (Область);
// Шапка
Область = Макет.ПолучитьОбласть ("Шапка");
ТабДок.Вывести (Область);
// ОплатаКурсов
Пока Результат.Следующий() Цикл
    Область = Макет.ПолучитьОбласть ("Строка");
    Область.Параметры.Курс = Результат.Курс;
    Область.Параметры.ОплаченнаяСумма = Результат.ОплаченнаяСумма;
    ТабДок.Вывести (Область);
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Ложь;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать ();
КонецПроцедуры
```

Конструктор формы отчета 🛛 🗙				
Выберите тип формы: ● Форма отчета ● Форма настроек отчета ● Произвольная форма ✓ Назначить форму основной ■ Использовать систему компоновку данных				
Имя: ФормаОтчета				
Синоним: Форма отчета				
Комментарий:				
 Командная панель формы сверху Командная панель формы снизу 	:			

Рис. 1.69. Окно конструктора формы отчета

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🔲
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование
<u>Сервис Окна</u> Сп <u>р</u> авка
I 🗋 🛁 🖌 🖻 🛍 🥌 🔯 🖕 🖬 🛤
🗮 Отчет УчетОплаты: ФормаОтчета 🛛 💶 🗙
Отчет Учет оплаты
Действия - 🥐 💭
Сформировать Закрыты
🗐 Диалог 🗏 Модуль 🚦 Реквизиты
Для получения подс САР NUM ги マ 👘 0, 0 📌 374 x 161

Рис. 1.70. Экранная форма отчета УчетОплаты

Запрос — это объект системы 1С:Предприятие, которым мы воспользовались в этой процедуре. Ему мы здесь отведем небольшой комментарий, но заметим, что в последующих главах запросы будут подробно рассмотрены на примерах различной степени сложности. В рассмотренной процедуре продемонстрировано эффективное использование запроса для извлечения информации из имеющихся данных.

Примечание

Запросы в системе 1С:Предприятие предназначены для выборки информации из базы данных. Можно считать, что запрос является требованием к системе для выбора и обработки данных. Например, с помощью запроса можно легко подсчитать внесенные суммы по конкретным курсам. В системе 1С:Предприятие существует специальный язык запросов, цель которого заключается в формулировании требований к системе — что нужно отобрать из имеющихся данных. А как система справляется с запросом, нас уже не интересует, главное — поставить задачу.

В первой строке листинга 1.7 создается новый объект типа Запрос, а ссылка на него присваивается переменной, которую мы также назвали словом Запрос. Далее формируется текст запроса, в котором говорится о том, что нас интересует документ ВнесениеОплаты, а более конкретно табличная часть этого документа — Оплатакурсов. Для этого используется ключевое слово из, а далее для интересующей нас табличной части документа вводится более короткое название:

```
Документ.ВнесениеОплаты.ОплатаКурсов
```

```
КАК ВнесениеОплатыЗаКурсы
```

которое можно использовать вместо полного имени. В тексте запроса указывается, что из этой табличной части нас интересует поле ВнесениеОплатыЗаКурсы.Курс, которому дается краткое имя Курс (ВнесениеОплатыЗаКурсы.Курс КАК Курс), а также сумма всех значений в поле ВнесеннаяСумма, которой дается краткое имя ОплаченнаяСумма. Конструкция сгруппировать по дает установку системе, что необходимо просуммировать отдельно все оплаты по каждому курсу.

Выполнение рассматриваемого запроса в результате приводит к формированию таблицы из двух колонок и числа строк, равного числу различных курсов, за которые были произведены оплаты. В первой колонке будет отражено название курса, а во второй — внесенная учащимися сумма за этот курс. После выполнения строки

```
Результат = Запрос.Выполнить ().Выбрать ()
```

в переменной Результат формируется ссылка на итоговую таблицу запроса. После этого начинается уже знакомое нам заполнение табличного документа. Теперь для полноты картины осталось перейти в режим работы 1С:Предприятие и посмотреть заполненный отчет (рис. 1.71).

Таким образом, мы рассмотрели пример формирования сводной информации по документам, связанным с оплатой за обучение. Когда количество документов станет значительным, то объем вычислительных операций приведет к заметной задержке при формировании отчета. Для устранения этой задержки будет правильным использовать регистры. Так, эффективный вариант построения информационной системы заключается в фиксировании в регистре накопления оплаченных сумм по мере создания каждого очередного документа. Эти и другие возможности информационной системы 1С:Предприятие 8.1 рассматриваются в последующих главах.

🏪 1С:Предприятие (учебная верс	ия) - Конфигурация 📃 🔲 🗙	
📱 🕂 айл Правка Таблица Операции	<u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
I 🗋 🛋 📓 👗 🗳 🚆	🗄 🎆 📄 🧟 M M+ M- 🎇 🖕	
🛕 Таблица	_ 🗆 ×	
Отчет по оплата	и за курсы	
Курс	Оплаченная сумма	
PowerPoint	700	
Базы данных	500	
Информатика (Excel)	1 200	
Таблица		
Для получения подсказки нажмите F1 САР NUM		

Рис. 1.71. Сформированный отчет УчетОплаты в режиме 1С:Предприятие

Отчет по клиентам

Предыдущий отчет позволял извлечь информацию о внесенных денежных средствах по каждому учебному курсу. Из рис. 1.71 мы видим, сколько внесено по каждому курсу ("Базы данных", "PowerPoint"). Не менее важно получить интегральную информацию о том, кто из наших клиентов уже заплатил и сколько конкретно. В этом разделе рассматривается решение подобной задачи. Сама разработка будет аналогична примеру предыдущего раздела. При этом мы воспользуемся уже созданным отчетом и фактически выполним небольшую его модернизацию. А именно мы внесем изменения в макет и в процедуру формирования отчета.

На рис. 1.72 показан новый макет для данного отчета — МакетОтчетаПоКлиентам. Относительно рис. 1.67 изменения незначительны. Текст процедуры формирования отчета, выполняемой при нажатии кнопки Сформировать, необходимо скорректировать. Новый вариант показан в листинге 1.8.



Рис. 1.72. Макет отчета по оплатам клиентов

Листинг 1.8. Процедура формирования отчета по оплатам клиентов

```
Процедура КнопкаСформироватьНажатие (Элемент)
Запрос = Новый Запрос;
Запрос. Текст = "ВЫБРАТЬ
| ВнесениеОплатыЗаКурсы.Ссылка.Клиент КАК Клиент,
 СУММА (ВнесениеОплатыЗаКурсы.ВнесеннаяСумма)
| КАК ОплаченнаяСумма
I N3
| Документ.ВнесениеОплаты.ОплатаКурсов
| КАК ВнесениеОплатыЗаКурсы
| СГРУППИРОВАТЬ ПО ВнесениеОплатыЗаКурсы.Ссылка.Клиент";
Результат = Запрос.Выполнить ().Выбрать ();
ТабДок = Новый ТабличныйДокумент;
Макет = Отчеты. УчетОплаты. Получить Макет ("МакетОтчетаПоКлиентам");
// Заголовок
Область = Макет.ПолучитьОбласть ("Заголовок");
ТабДок.Вывести (Область);
// Шапка
Область = Макет.ПолучитьОбласть ("Шапка");
ТабДок.Вывести (Область);
```

```
// ОплатаКурсов
Пока Результат.Следующий() Цикл
Область = Макет.ПолучитьОбласть("Строка");
Область.Параметры.Клиент = Результат.Клиент;
Область.Параметры.ОплаченнаяСумма = Результат.ОплаченнаяСумма;
ТабДок.Вывести(Область);
КонецЦикла;
ТабДок.Овывести(Область);
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.ТолькоПросмотр = Ложь;
ТабДок.Показать();
КонецПроцедуры
```

Ключевое изменение заключается в конструкции:

ВнесениеОплатыЗаКурсы.Ссылка.Клиент

которая позволяет обратиться к информации, расположенной в шапке рассматриваемого документа.

Примечание

На уровне объектов базы данных информация о документе хранится в двух таблицах — в основной таблице документа и таблице табличной части документа. В таблице табличной части имеется поле Ссылка, которое представляет ссылку на документ, которому подчинена рассматриваемая запись табличной части.

🏪 1С:Предприятие (учебная версия) - Конфигурация	
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
I 🗋 🔟 📓 🔏 🛍 🥌 🔍 🎽 🎆 🗐 🧝 м м	+ M- 🞇 🖵
🐱 Внесение оплаты: Внесение оплаты 000000005:0	18 _ 🗆 🗙
Действия - 🛃 🔃 🎦 🏄 ?	
Номер: 00000005	
Дата: 31.08.2008 12:59:08	
Клиент: Николаев Сергей Федорович	×
Телефоны: 490-77-55	
엄벌롱철믋수부왕똜	
N Курс Внесенная сумма	
1 Базы данных	1 000
ОК Записать Закры	ять Печать
政 Отчет Уч 🍠 Справочн 🏊 Докумен 🕌 Таблица 🛛 🛽	Внесе:08
Для получения подсказки нажмите F1 САР	P NUM //

Рис. 1.73. Документ о внесении денежных средств учащимся

Ранее, рассматривая примеры, мы в режиме 1С:Предприятие ввели информацию только по одному учащемуся. Теперь для проверки функциональности разработки добавим еще одного клиента, "запишем его на курсы" и с помощью документа внесениеОплаты оформим оплату обучения (один из документов показан на рис. 1.73). После этого у нас в базе данных будет присутствовать информация по двум учащимся, которые частично оплатили обучение нескольким дисциплинам.

Теперь, если сформировать рассматриваемый отчет, получим результат оплат с группировкой по клиентам (рис. 1.74).



Рис. 1.74. Отчет с группировкой по клиентам

Отчет по клиентам с детализацией курсов

Для полноты картины разработаем еще один отчет, в котором будет отражено, кем, сколько и за какие учебные курсы внесено денежных средств. В предыдущем отчете мы получили информацию по итоговым оплатам учащихся. Для начала создадим новый макет — МакетСДетализацией (рис. 1.75). Его содержание не требует комментария, т. к. все технические действия мы уже проделывали в предыдущих разработках. Полученный отчет в режиме 1С:Предприятие показан на рис. 1.76, а текст процедуры формирования отчета приведен в листинге 1.9.

Листинг 1.9. Процедура формирования отчета с детализацией по курсам

```
Процедура КнопкаСформироватьНажатие (Кнопка)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
| ВнесениеОплатыЗаКурсы.Ссылка.Клиент КАК Клиент,
| ВнесениеОплатыЗаКурсы.Курс КАК Курс,
| СУММА (ВнесениеОплатыЗаКурсы.ВнесеннаяСумма)
```

```
| КАК ОплаченнаяСумма
I N3
| Документ.ВнесениеОплаты.ОплатаКурсов
| КАК ВнесениеОплатыЗаКурсы
| СГРУППИРОВАТЬ ПО ВнесениеОплатыЗаКурсы.Ссылка.Клиент,
| ВнесениеОплатыЗаКурсы.Курс";
Результат = Запрос.Выполнить ().Выбрать ();
ТабДок = Новый ТабличныйДокумент;
Макет = Отчеты. УчетОплаты. Получить Макет ("МакетсДетализацией");
// Заголовок
Область = Макет.ПолучитьОбласть ("Заголовок");
ТабДок.Вывести (Область);
// Шапка
Область = Макет.ПолучитьОбласть ("Шапка");
ТабДок.Вывести (Область);
// ОплатаКурсов
Пока Результат.Следующий() Цикл
    Область = Макет.ПолучитьОбласть ("Строка");
    Область.Параметры.Клиент = Результат.Клиент;
    Область.Параметры.Курс = Результат.Курс;
    Область.Параметры.ОплаченнаяСумма = Результат.ОплаченнаяСумма;
    ТабДок.Вывести (Область);
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Ложь;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать ();
КонецПроцедуры
```

	📩 Конфигуратор (учебная версия) - Конфигурация						<u>_ </u>		
	т жаил Правка Тарлица конфигурация Отладка Администрирование Сервис Окна Справка П 🔄 🖬 🖌 🖻 🖻 🔄 🔄 🗘 לא 🔁 👘 א 🔽 🕹 🕲 א 👘 🖉 🔅						🦹		
ſ	\Lambda Отчет У	ІчетО	плати	ы: МакетСДеталі	изацией			_	. 🗆 🗙
			1	2		3	1	4	^
	Заголовок	1	<u> </u>						
		2		Отчет по	оплата	ам клиент	гов		
		3							
	Шапка	4		Клиент		Курс		Оплаченная сумма	
	Строка	5		≺Клиент≻		<Курс>		<ОплаченнаяСумма	3>
		6							
		7	1						
ш									
Ţ	Для получения подсказки нажмите F1 САР NUM ги • //							CAP NUM	ru▼ //

Рис. 1.75. Измененный макет отчета

🟪 1С:Предприятие (учебная версия) - Конфигурация 📃 🔲 🗙					
<u>∲</u> айл <u>П</u> равка Таблица Операц	ии <u>С</u> ервис <u>О</u> кна Сп <u>р</u> а	вка			
D 🛋 🖬 👗 🖪 📾 🖾	🕽 t t 🐥 📰	🔜 🧟 M M+ M-	₩		
🚂 Таблица			∃ × [
Отчет с детализаций					
Клиент	Курс	Оплаченная сумма			
Сидоров Евгений Петрович	PowerPoint	1 200			
Сидоров Евгений Петрович	Базы данных	500			
Сидоров Евгений Петрович	Сидоров Евгений Петрович Информатика (Excel) 1 200				
Николаев Сергей Федорович Базы данных 1 700					
< ×					
💷 Отчет Учет оплаты 💦 Таблица					
Для получения подсказки нажмите	F1	CAP NUM	1 //		

Рис. 1.76. Отчет по оплатам с детализацией учебных курсов

Итоги знакомства с конфигуратором

Итак, данная глава завершена, и стоит оглянуться на выполненные разработки и подчеркнуть основные итоги. Наверное, главный результат заключается в том, что мы узнали, как выглядит технология разработки конфигураций для информационной системы 1С:Предприятие 8.1. Возможно, что до этого вы работали только с готовыми конфигурациями как пользователь. Теперь же можно взглянуть на эту сложную информационную систему и глазами разработчика (программиста) конфигураций.

Удобное комбинирование работы в режиме конфигуратора и режиме 1С:Предприятие позволяет существенно облегчить работу по созданию прикладных решений на платформе 1С:Предприятие 8.1.

На примере использования конструктора печати мы познакомились с возможностями, которые позволяют разработчику существенно ускорить процесс создания программного кода. В этом случае от него требуется только включение небольших собственных фрагментов в автоматически созданные процедуры.

В ходе рассмотрения примеров этой главы мы получили практические навыки работы с наиболее популярными прикладными объектами системы 1С:Предприятие 8.1 — перечислениями, справочниками, документами и табличными документами.

В целом же итог данной главы заключается в том, что рассмотренные примеры являются фундаментом для более сложных разработок, которые встретятся в последующих главах книги.

глава 2



Документы

В этой главе основное внимание будет уделено работе с документами — с ними мы уже немного познакомились в *главе 1*. Объект конфигурации *"документ"* является прикладным и предназначен для фиксирования информации о событиях, происходящих в организации.

В *главе 1* один из примеров был связан с разработкой в конфигураторе документа, фиксирующего денежные поступления от клиентов. Мы познакомились с электронной формой документа, а также с его печатным вариантом (печатной формой). Рассмотренные примеры можно считать базой для данной и последующих глав книги, где работе с документами будет уделено значительное внимание.

С практической точки зрения документ можно считать прикладным объектом, назначением которого является ввод и фиксация информации о событиях, происходящих в организации. Эти события, как правило, влияют на количественные показатели в рассматриваемой предметной области (чаще всего учет связан с движением денежных средств и материальных ценностей). В качестве примеров можно предложить факт выписки денежных сумм сотрудникам, передачу товаров и материалов с одного склада на другой и т. д. Именно для того чтобы отражать подобные факты в системе учета, объект конфигурации "документ" и был разработан.

События, происходящие в организации, отличаются привязкой ко времени, в связи с чем параметр, характеризующий точное время, является обязательным реквизитом любого документа. Мы уже видели, что действия при разработке прикладных документов начинаются в конфигураторе с создания структуры документа — набора полей с указанием их типов данных. Кроме того, с помощью программных процедур реализуются необходимые действия с документом. Результатом работы в конфигураторе является формирование информационной структуры, которая позволяет пользователю в режиме 1С:Предприятие с такими документами работать (создавать, вносить в них информацию, изменять, удалять).

Основной чертой, отличающей документы от других объектов, является их способность *проведения* (возможность формировать проводки, являющиеся результатом информации, присутствующей в документах). Это очень важная особенность, и ее необходимо пояснить. В информационной системе 1С:Предприятие имеются разнообразные структуры информации, которые описывают количественное состояние дел в организации (например, стоимостный и количественный учет разнообразных товаров). Проведение документов означает факт внесения изменений в эти структуры данных (причина изменений связана с информацией, содержащейся в документах). Например, поступление по документу определенного товара изменяет суммарное количество данного товара на складе.

До тех пор, пока документ не проведен, состояние учета неизменно. В этом случае функциональность документа не реализована, но сам он, тем не менее, зарегистрирован в базе данных.

С другой стороны, в информационной системе при необходимости должна существовать и возможность отмены проведения (возвращения информации в первоначальное состояние). Например, при отмене проведения документа "Поступление товара" должна быть скорректирована информация в регистре, который используется для учета количества товаров на складах. Подобная возможность *отмены проведения* документов в системе 1С:Предприятие также реализована.

Важно подчеркнуть, что документы, разработанные для различных стандартных конфигураций системы 1С:Предприятие, являются аналогами реальных документов, которые используются в работе организаций.

Многочисленные документы, используемые в реальном документообороте, различны по структуре. Например, основное содержание одних документов могут составлять таблицы (скажем, списки номенклатуры, услуг или сотрудников), а в других они могут отсутствовать. Существуют также документы с наличием нескольких табличных частей (например, одна табличная часть отводится для списка сотрудников, работающих на постоянной основе, а другая — для списка совместителей). В связи с этим объект конфигурации *"документ"* системы 1С:Предприятие должен обладать хорошими возможностями конструирования. Использование рассматриваемого объекта конфигурации при разработке прикладных решений должно позволять легко создавать различные документы — приходную накладную, счет или документ о перемещении товаров со склада на склад. В системе 1С:Предприятие 8.1 ситуация выглядит именно так — все конструктивные возможности во власти разработчика.

Опишем теперь основные параметры рассматриваемого объекта конфигурации. В любом документе (созданном в системе 1С:Предприятие, а также в реальных документах) присутствует информация, которая характеризует его в целом. Это дата создания и порядковый номер (в последовательности документов одного типа).

Примечание

Поскольку документ вносит изменения в состояние учета, он всегда привязан к кон-кретному моменту времени.

Кроме того, в любом документе присутствует и общая информация другого плана. Например, организация, подразделение или ответственное лицо. Для подобных параметров в объекте конфигурации "документ" отводится область, называемая шапкой. Ее фактически составляют реквизиты, характеризующие документ в целом.

Кроме общей области, документ может содержать список данных, схожих по структуре. Например, список сотрудников, организаций или товаров. Каждый из

этих списков включает в себя еще и уточняющую и дополнительную информацию. Например, вместе с названием товара указываются количество и цена. Для отражения такой информации в разрабатываемый объект конфигурации следует включить один или несколько подчиненных (внутренних) объектов, которые называются *табличными частями*.

Теперь мы рассмотрим практические приемы программирования документов. При этом мы предварительно разработаем новую конфигурацию, включающую необходимые объекты.

Подготовка конфигурации

Основные технические действия для создания новой конфигурации были подробно рассмотрены в *главе 1*. И первым объектом конфигурации, который мы создадим, будет справочник Фирмы. Дополнительные реквизиты в этом справочнике нам не потребуются, поэтому перейдем в режим 1С:Предприятие и внесем в него несколько записей (рис. 2.1).

🏪 1С:Предприятие	📙 1С:Предприятие (учебная версия) - Конфигурация 📃 🗖 🗙			
🛛 <u>Ф</u> айл <u>П</u> равка Оп	ерации <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка			
I 🗋 🛋 🖬 🛛 🕹 🗉	Ъ 🛍 🐥 📰 📰 🧝 и м м+ м- 🞘 🗸			
🗐 Справочник Ф	ирмы _ 🗆 🗙			
Действия 🕶 😫 😫	Действия 🕶 🔄 🛬 ≼ 🍇 🎢 🎦 🗃 - 🐼 😥 🕐			
Код	Наименование 🔹 🔿			
- 000000001	Меда			
- 000000002	Тико			
- 000000003	Юниор			
▼				
🗐 Справочник Фирмы				
Для получения подск	азки нажмите F1 САР NUM			

Рис. 2.1. Заполнение справочника Фирмы

Также создадим справочник номенклатура, где будут перечислены названия наших товаров (рис. 2.2).

Документ Заказ

С технологией создания объекта конфигурации типа "документ" мы уже познакомились в *славе 1*. Поэтому при разработке документа заказ подчеркнем только наиболее важные моменты практических действий. На рис. 2.3 показана закладка Данные окна редактирования объекта конфигурации. Видно, что область шапки документа содержит один дополнительный реквизит — заказчик (тип данных — СправочникСсылка.Фирмы).

🏪 1 С:Предприятие (учебная версия) - Конфигурация 💦 💶 🗵				
🛯 <u>Ф</u> айл <u>П</u> равка Опер	рации <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка			
I 🗋 🛋 🖬 📈 🖻	в 🕞 🥌 🐥 і 🎹 🧮 🧝 і м. м+ м- 🎘 🖕			
🗐 Справочник Но	менклатура _ 🗆 🗙			
Действия • 😫 🛬 素 🏪 🏹 🛅 🗃 • 🐼 🙋 ?				
Код	Наименование 🗸 🔼			
- 000000003	Кухня Уют-3			
- 000000001	Мягкая мебель			
- 00000002	Стенка (мод. 15-7)			
Э Справочник Фирмы Э Справочник Номенклатура				
Для получения подсказки нажмите F1 САР NUM				

Рис. 2.2. Внесение данных в справочник Номенклатура

Для табличной части документа выберем имя перечень и внесем в нее два реквизита:

- Товар (ТИП ДАННЫХ СправочникСсылка. Номенклатура);
- Количество (тип данных число без дробной части) для внесения количества заказанных изделий.

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🗙				
<u>Ф</u> айл <u>П</u> равка Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна				
Справка				
I 🗋 🝙 🕺 🗟 🗳 I 🛞 🎽 📃 📃	•			
🗟 Документ Заказ _ 🗆 🗙				
Основные 😋 💻 🛬 🛊 🔛	I			
▶ Данные	I			
Нумерация	I			
Движения	I			
Последовательности	I			
Журналы 🏤 🛀 式 🎽 🛔 🛓	I			
Формы	I			
Макеты	I			
Подсистемы	I			
Ввод на основании Количество	I			
Действия 🔹 <Назад Далее> Закрыть Справка				
🐱 Документ Заказ				
Для получения подсказки нажмите F1 САР NUM ги •				

Рис. 2.3. Структура документа Заказ

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🔲 🗙				
📱 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кн	а Сп <u>р</u> авка			
🗋 🚄 🔛 X 🖻 🖻 🕌 🔍 💙	🎹 🛅 🧝 M M+ M- 🎇 🖕			
😼 Заказ: Заказ 000000001 от 17.08	3.2008 8:35:38 * 💶 🗙			
Действия - 🛃 🔃 😼 👔 🤶)			
Номер: 000000001				
Дата: 17.08.2008 8:35:38	17.08.2008 8:35:38			
Заказчик: Меда	x			
엄벌롱철물루 1 문란 5				
N Товар	Количество			
1 Мягкая мебель	5			
2 Стенка (мод. 15-7)	7			
3 Кухня Уют-3	11			
	ОК Записать Закрыть			
<	>			
🗐 Справочник Фирмы 🛛 🚢 Документы З	аказ 🛛 🐱 Заказ: Заказ 00:38 *			
Для получения подсказки нажмите F1 САР NUM				

Рис. 2.4. Заполненный документ Заказ

Документ заказ будет фиксировать в базе данных заказы товаров нашими партнерами. Все необходимые действия в конфигураторе выполнены, поэтому перейдем в режим 1С:Предприятие и заполним несколько документов заказ. Один из них показан на рис. 2.4.

Основные действия с документами связаны с их проведением. Об этом мы поговорим в следующей главе (это связано с тем, что кроме документов к проведению имеют отношение и регистры). Здесь же мы рассмотрим ряд вопросов, связанных с программным созданием документов и доступом к реквизитам документов.

Программное создание и запись документов

Для интерфейса разработки мы создадим *обработку*. Данный объект конфигурации достаточно часто используется для реализации различных алгоритмов обработки данных. Поэтому в уже знакомом нам окне конфигурации щелкнем правой кнопкой мыши по пиктограмме **Обработки** и в открывшемся контекстном меню воспользуемся пунктом **Добавить**. Далее в окне редактирования объекта конфигурации дадим обработке имя РаботаСДокументами (рис. 2.5).

В этой обработке нам потребуется подчиненный объект конфигурации — форма, на которой будет размещено несколько элементов управления. Для ее создания перейдем на закладку **Формы**. Здесь с помощью соответствующей пиктограммы (Добавить) начнем разработку формы. В диалоге с конструктором (рис. 2.6) введем имя формы — формадлятеста. В результате уже знакомых действий мы получим экранную форму (рис. 2.7) с двумя стандартными кнопками: Выполнить и Закрыть.

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🔲 🗙				
<u>Ф</u> айл Правка Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна				
Справка				
🗋 📫 📕 🔏	è 🖻 🧉 🖉	5 C 🕅 M		
よ Обработка Ра	ботаСДокумента	ми _ 🗆 Х		
• Основные	14			
Данные	имя.	Работасдокументами		
Формы	Синоним:	Работа с документами		
Макеты	Комментарий:			
Подсистемы	••••••	·		
Права				
Интерфейсы				
Прочее				
Действия 🗸	(Назад	Далее> Закрыть Справка		
<mark>а</mark> ∲ Обработка РаботаСДокуме				
Для получения подск	азки нажмите F1	CAP NUM ru -		

Рис. 2.5. Окно редактирования обработки РаботаСДокументами

Конструктор формы обработки Х		
Выберите тип • Форма обр Произволы • Назначить	Формы: аботки ная форма форму основной	
Имя:	ФормаДляТеста	
Синоним:	Форма для теста	
Комментарий:		
🗌 Командная	панель Формы сверху	
— Командная панель формы снизу		
< H	азад Далее > Готово Отмена Справка	

Рис. 2.6. Окно конструктора формы обработки

Теперь добавим на форму элемент управления — *поле ввода*. Для этого в меню Форма воспользуемся пунктом Вставить элемент управления. Выберем Поле ввода, в качестве имени укажем ФирмаЗаказчик, а в качестве типа данных — СправочникСсылка.Фирмы (рис. 2.8).



Рис. 2.7. Окно формы обработки РаботаСДокументами в режиме конфигуратора

🛃 Конфигуратор (учебная версия) - Конфигурация		
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование <u>С</u> ервис		
Окна Справка		
I 🗋 🛁 🖬 🔏 📾 🥌 🖄 📩 🖒	» •	
🚟 Обработка РаботаСДокументами: ФормаДляТеста 🛛 💶 🗙		
Обработка Работа с документами		
Фирма заказчик:		
Выполнить Закрыть		
🔜 Диалог 🗏 Модуль 📑 Реквизиты		
🛃 Обработка РаботаСДо 🚍 Обработка Ра: Форма 📰 Обр: ФормаДляТеста		
Для получения подсказ САР NUM ги т 0, 0		

Рис. 2.8. Добавление в форму обработки поля ввода

Все основные технические действия по разработке произойдут на закладке **Модуль**, предназначенной для размещения текстов программных процедур на встроенном языке. На этой закладке следует расположить процедуру обработки нажатия на кнопку **Выполнить**. Ее текст приведен в листинге 2.1. Здесь мы воспользовались методом СоздатьДокумент() для программного создания документа, что аналогично интерактивным действиям.

Встроенная функция текущаядата () выдает текущую (системную) дату на компьютере, которую мы заносим в качестве даты создаваемого документа.

После создания нового документа его следует сохранить в базе данных. Сохраняется документ с помощью метода Записать ().

Листинг 2.1. Процедура для создания нового документа

```
Процедура КнопкаВыполнитьНажатие (Кнопка)
НовыйЗаказ = Документы.Заказ.СоздатьДокумент();
НовыйЗаказ.Дата = ТекущаяДата();
НовыйЗаказ.Заказчик = ФирмаЗаказчик;
НовыйЗаказ.Записать();
КонецПроцедуры
```

После размещения процедуры можно перейти в режим 1С:Предприятие и в меню Операции воспользоваться пунктом Обработка. Здесь следует выбрать только что созданную обработку Работа с документами и открыть ее форму. Теперь нужно указать в поле ввода одну из фирм и нажать кнопку Выполнить. В результате в списке документов Заказ появится новый документ с внесенными датой и фирмой. Заметим, что табличная часть заказа пуста.

Поиск документов

Все введенные документы, как мы уже видели, отражаются в форме списка. Однако часто требуется из многочисленного количества документов отобрать только определенную категорию документов по какому-либо принципу. Продемонстрируем, как это сделать с помощью программных ресурсов (система 1С:Предприятие предоставляет широкие возможности по поиску и отбору отдельных документов).

Рассмотрим ситуацию, когда требуется обеспечить отбор документов по дате. Здесь нам потребуется форма с новым набором элементов управления. Для практической работы можно воспользоваться несколькими вариантами:

- создать новую обработку и разработать форму ввода;
- скорректировать имеющуюся форму обработки РаботаСДокументами;
- создать новую форму в обработке РаботаСДокументами.

Воспользуемся первым вариантом и создадим обработку выборкадокументов. На закладке Формы окна редактирования объекта конфигурации сформируем форму, на которой разместим набор элементов (рис. 2.9):

- ♦ поле ввода дата1 для ввода начальной даты (тип данных дата);
- поле ввода дата2 для ввода конечной даты (тип данных дата);
- поле списка ПереченьДокументов.

Теперь на закладке **Модуль** оформим процедуру обработки нажатия на кнопку **Выполнить** (листинг 2.2).



Рис. 2.9. Форма для организации отбора документов

Листинг 2.2. Процедура отбора документов по дате

```
Процедура КнопкаВыполнитьНажатие (Кнопка)
НачальнаяДата = НачалоДня (Дата1);
КонечнаяДата = КонецДня (Дата2);
ВыборкаЗаказ = Документы.Заказ.Выбрать (НачальнаяДата, КонечнаяДата);
ПереченьДокументов.Очистить ();
Пока ВыборкаЗаказ.Следующий () Цикл
Информация = "Номер документа: " + ВыборкаЗаказ.Номер +
" Дата документа: " + ВыборкаЗаказ.Дата;
ПереченьДокументов.Добавить (Информация);
КонецЦикла;
```

КонецПроцедуры

Здесь поля дата1 и дата2 используются для указания, соответственно, начала и окончания интересующего нас интервала по дате. Для того чтобы получить документы, попадающие в интервал, мы воспользовались двумя встроенными функциями:

началодня (), которая возвращает дату и время начала дня для указанной даты;

• конецдня (), которая определяет дату и времени конца дня для указанной даты.

Отбор найденных документов отражается в поле списка. Для этого мы его предварительно очищаем, вызывая метод Очистить (), а затем с помощью метода Выбрать () производим выборку документов в диапазоне указанных дат. Первый параметр метода задает начальную дату поиска, а второй — конечную. После этого организуется цикл обхода отобранных записей с помощью метода Следукщий (). Теперь, если перейти в режим 1С:Предприятие, открыть форму обработки и установить даты, мы получим список отобранных документов (рис. 2.10).

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🗖 🗙
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
I 🗋 🛁 Х ங 🛍 🗉 🔯 🐥 I 🎹 🛅 🧟 м м+ м- 🎘 🖕
🔹 Обработка Выборка документов 🛛 _ 🗆 🗙
Дата1: 29.08.2008 🗰 Дата2: 01.09.2008 🕮
Номер документа: 000000012 Дата документа: 31.08.2008 16:13:25 Номер документа: 000000013 Дата документа: 31.08.2008 16:21:05 Номер документа: 000000014 Дата документа: 31.08.2008 16:24:11 Номер документа: 000000015 Дата документа: 31.08.2008 16:24:14
Выполнить Закрыть
🖋 Обработка. Выборка докум
Для получения подсказки нажмите F1 САР NUM

Рис. 2.10. Результат отбора документов по дате

Если в методе выбрать () не указывать интервал по дате (это необязательные аргументы), то из базы данных будут выбраны все документы данного вида.

Включение в условие отбора параметра

Часто требуется выбирать не все записи из базы данных, а лишь те, которые имеют конкретное значение в определенном поле (например, заказы конкретной фирмы). В этом разделе мы продемонстрируем типичные действия в данной ситуации. На рис. 2.11 приведена форма с элементами, которые нам потребуются.

Технически в обработке выборкадокументов мы создадим новую форму.

Расположим на ней следующие элементы управления:

- поле ввода фирма (тип данных СправочникСсылка. Фирмы);
- поле списка ПереченьДокументов.

Теперь для кнопки **Выполнить** следует создать процедуру обработки в соответствии с листингом 2.3.

Листинг 2.3. Процедура отбора документов по фирме

```
Процедура КнопкаВыполнитьНажатие(Кнопка)
Условие = Новый Структура("Заказчик", Фирма);
ВыборкаДокументов = Документы.Заказ.Выбрать(, , Условие);
ПереченьДокументов.Очистить();
```

```
Пока ВыборкаДокументов.Следующий() Цикл
Информация = "Номер документа: " + ВыборкаДокументов.Номер +
" Дата: " + ВыборкаДокументов.Дата;
ПереченьДокументов.Добавить (Информация);
КонецЦикла;
КонецПроцедуры
```

📩 Конфигуратор (учебная версия) - Конфигурация			
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование <u>С</u> ер			
<u>О</u> кна Сп <u>р</u> авка			
I D 🛋 🖬 X P≥ @ 🛎 @ S & M ×	» ▼		
🚟 Обработка ВыборкаДокументов: Форма2	_ 🗆 ×		
Обработка Выборка документов	I		
Действия - 🥐 💭			
	I		
Фирма: В	I		
Перечень документов:	I		
	I		
	I		
	I		
🔛 Выполнить Закрыты			
📃 Диалог 🗏 Модуль 📑 Реквизиты			
🛐 Служебные сообщения			

Рис. 2.11. Форма для отбора документов по фирме



Рис. 2.12. Пример отбора документов по фирме

В тексте листинга 2.3 для передачи параметра мы использовали объект Структура. В данном случае нас не интересует интервал дат при выборе документов из базы данных. Поэтому первые два аргумента функции оставлены пустыми, а условие реализовано в третьем аргументе:

```
ВыборкаДокументов = Документы.Заказ.Выбрать(, , Условие);
```

На рис. 2.12 продемонстрирован результат выбора документов из базы данных по указанному условию.

Упорядочивание

Несколько изменим предыдущий пример. Будем считать, что нам требуется дополнительно упорядочить результат отбора по дате в порядке возрастания. Для этого в методе выбрать () предназначен четвертый аргумент. В листинге 2.4 представлена процедура, позволяющая отобрать документы, относящиеся к нужной фирме, и упорядочить их по дате.

🏪 1 С:Предприятие (учебная версия) - Конфигурация 📃 🗆 🗙		
<u>Файл Правка Операции Сервис Окна Справка</u>		
I 🗋 🔟 📓 🔏 📾 🥌 🗐 🎾 I 🎆 🛅 🧝 им м+ м- 🎇 🖕		
🔹 Обработка Выборка документов 🛛 💶 🗙		
Действия - 🕐		
Фирма: Тико х		
Перечень документов:		
Номер документа: 000000004. Дата: 19.08.2008 17:54:27		
Номер документа: 000000011 Дата: 26.08.2008 9:41:30		
Номер документа: 000000008 Дата: 26.08.2008 12:00:00		
Номер документа: 000000009 Дата: 26.08.2008 12:00:01		
Номер документа: 000000013 Дата: 31.08.2008 16:21:05		
Номер документа: 000000014 Дата: 31.08.2008 16:24:11		
Выполнить Законть		
<mark>а^ф Обработка. Выборка докум</mark>		
Для получения подсказки нажмите F1 САР NUM		

Рис. 2.13. Форма для отбора документов по фирме с упорядочиванием по дате

Листинг 2.4. Процедура отбора документов с упорядочиванием по дате

```
Процедура КнопкаВыполнитьНажатие(Кнопка)
Условие = Новый Структура("Заказчик", Фирма);
Упорядочивание = "Дата Возр";
ВыборкаДокументов = Документы.Заказ.Выбрать(, , Условие, Упорядочивание);
```

```
ПереченьДокументов.Очистить();
Пока ВыборкаДокументов.Следующий() Цикл
Информация = "Номер документа: " + ВыборкаДокументов.Номер +
" Дата: " + ВыборкаДокументов.Дата;
ПереченьДокументов.Добавить (Информация);
КонецЦикла;
КонецПроцедуры
```

В режиме 1С:Предприятие результат выполнения данной процедуры показан на рис. 2.13.

Программная установка свойств документов

В системе 1С:Предприятие имеется объект документобъект, который позволяет обратиться к значениям реквизитов, задаваемым в конфигураторе. Для получения данного объекта следует воспользоваться методом получитьобъект (). Используя этот объект, мы можем не только считывать, но и устанавливать необходимые реквизиты документа. В листинге 2.5 приведена процедура, которая позволяет сменить даты документов на текущую дату. Ее можно расположить в качестве процедуры обработки нажатия на кнопку, расположенную на форме. Метод Записать () позволяет записать элемент справочника в базу данных; если его не вызвать, изменения будут утеряны. Результат выполнения данной процедуры отражен в форме списка документов (рис. 2.14).

📙 1С:Предприятие (учебная версия) - Конфигурация 📃 🗆 🗙			
<u>Файл Правка Операции Сервис Окна Справка</u>			
I 🗋 🛁 Х 🖻 🛍 🗉 🖸 🖕 🕈 I 🏧 🧮 🦉 м. м+ м- 🎘 🖕			
🚢 Документы Заказ _ 🗆 🗙			
Действия 🗸 🔄 🔄 式 😽 (e) 🕂 🕅 🐚	📲 • 🏹 🖸 🕐	
Дата 🔻	Номер	Заказчик	
19.08.2008 17:54:27	000000001	Меда	
19.08.2008 17:54:27	00000002	Меда	
19.08.2008 17:54:27	00000003	Меда	
19.08.2008 17:54:27	000000004	Тико	
19.08.2008 17:54:27 000000005 Юниор			
🖶 Оорасотка: высорка докум 📇 документы Заказ			
Для получения подсказки нажмите F1 САР NUM			

Рис. 2.14. Результат изменения дат документов

Листинг 2.5. Процедура установки даты документов

Процедура КнопкаВыполнитьНажатие(Кнопка) ВыборкаЗаказ = Документы.Заказ.Выбрать();

```
Пока ВыборкаЗаказ.Следующий() Цикл
Объект = ВыборкаЗаказ.ПолучитьОбъект();
Объект.Дата = ТекущаяДата();
Объект.Записать();
КонецЦикла;
КонецПроцедуры
```

Заполнение документа на основании имеющихся данных

Рассмотрим программное заполнение одного документа на основании данных, содержащихся в другом, уже заполненном документе. Это очень полезное действие — оно исключает повторный ввод информации, что повышает производительность работы и уменьшает вероятность ошибки. Один документ (Заказ) у нас есть, и для начала требуется создать еще один — Заявка (рис. 2.15).

На рис. 2.15 показана закладка Данные окна редактирования объекта конфигурации. Видно, что область шапки документа содержит один дополнительный реквизит — ФирмаЗаказчик (тип данных — СправочникСсылка.Фирмы). Для табличной части документа выберем имя перечень Товаров и внесем в нее два реквизита:

- Номенклатура (ТИП Данных СправочникСсылка. Номенклатура);
- количество (тип данных число без дробной части) для внесения количества заявляемых изделий.



Рис. 2.15. Окно редактирования документа Заявка

После этого потребуется в режиме 1С:Предприятие заполнить несколько документов по созданному шаблону (сформировать несколько заявок). Один из заполненных документов показан на рис. 2.16.

Теперь наша задача заключается в том, чтобы на основании заполненной заявки программно сформировать документ заказ. В нем будет указана фирма, а также заполнена табличная часть с перечнем товаров. Для технической реализации данной задачи создадим форму обработки (рис. 2.17). На форме расположим поле вво-

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🔲 🗙			
<u>∯</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка			
I 🗋 🔟 🐰 🖻 🔒 🥌 🔍 📍 🎆 🧮 🎇 🖉 м м+ м- 🎘 🗸			
🐱 Заявка: Заявка 000000001 от 19.08.2008 21:34:44 🛛 💶 🗙			
Действия 🗸 🛃 💀 🎦 🍒 🕐			
Номер:	000000001		
Дата:	19.08.2008 21:3	34:44	
Фирма заказчик:	Юниор		×
엄벌롟쎀틃↑↓뢠뚋			
N Номенкла	N Номенклатира Количество		
1 Мягкая м	ебель		3
2 Стенка (мод. 15-7) 1			1
3 Кухня Уют-3 5			
📇 Документы Заявка 🛛 🐱 Заявка: Заявка 0000000:44			
Для получения подсказки нажмите F1 САР NUM			

Рис. 2.16. Заполнение заявки в режиме 1С:Предприятие

📩 Конфигуратор (учебная версия) - Конфигурация	<u> </u>	
🛛 🛨 айл Правка Форма Конфигурация Отладка Администрирование .	<u>С</u> ервис	
<u>Окна</u> Сп <u>р</u> авка		
🗋 🛁 X 🖻 🛍 🎽 🕀 🍟	- -	
📓 Обработка РаботаСДокументами: ФормаДляТеста 💶 🗶 📃		
Обработка Работа с документами		
Заявка:	×	
Выполнить Закрыть		
🔜 Диалог 🗏 Модуль 🚦 Реквизиты		
🔐 Обработка РаботаСДокуме 🚟 Обработ: ФормаДляТеста		
Для получения подсказ САР NUM и тит - 0, 0 👫 391 x 133		

Рис. 2.17. Форма обработки для создания заказа на основании заявки

да Заявка (тип данных — документСсылка.Заявка). Теперь для обеспечения функциональности нажатия на кнопку Выполнить следует добавить процедуру, представленную в листинге 2.6.

Листинг 2.6. Процедура создания заказа на основании заявки

```
Процедура КнопкаВыполнитьНажатие (Кнопка)

НовыйЗаказ = Документы.Заказ.СоздатьДокумент();

НовыйЗаказ.Заказчик = Заявка.ФирмаЗаказчик;

НовыйЗаказ.Дата = ТекущаяДата();

Для Каждого СтрокаЗаявка Из Заявка.ПереченьТоваров Цикл

СтрокаЗаказТовар = НовыйЗаказ.Перечень.Добавить();

СтрокаЗаказТовар.Товар = СтрокаЗаявка.Номенклатура;

СтрокаЗаказТовар.Количество = СтрокаЗаявка.Количество;

КонецЦикла;

НовыйЗаказ.Записать();

КонецПроцедуры
```

Здесь мы с помощью уже знакомого нам метода Создать Документ () создаем новый документ Заказ, копируем заказчика и устанавливаем дату создаваемого документа. Далее организуется цикл, проходящий по всем записям в документе Заявка:

Для Каждого СтрокаЗаявка Из Заявка.ПереченьТоваров Цикл

뜸 IC:Предприятие (учебная версия) - Конфигурация 📃 🔲 🗙			
<u>∎ Ф</u> айл <u>П</u> ра	🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка		
I 🗋 🔟 📓 🔏 🛍 🥌 💽 🍟 I 🎹 🛅 🧝 м. м+ м- 🎘 🗸			
屋 Заказ: Заказ 000000007 от 19.08.2008 21:51:11 🛛 💶 🗙			
Действия 🗸 🛃 🔯 🎥 🥐			
Номер:	00000007		
Дата:	19.08.2008 21:51:11		
Заказчик:	Юниор	x	
· 달 달 -	열 별 등 철 다 🖡 🔮 🛱		
N T	овар	Количество	
1	Иягкая мебель	3	
20	2 Стенка (мод. 15-7) 1		
3 Кухня Уют-3 5			
ОК Записать Закрыть			
📇 Документы Заказ 🛛 🐱 Заказ: Заказ 000000007:11			
Для получения подсказки нажмите F1 САР NUM			

Рис. 2.18. Программно созданный документ Заказ

В цикле производятся добавление строки в формируемый документ заказ и заполнение табличной части в этом документе. В заключительной части процедуры созданный документ вносится в базу данных:

НовыйЗаказ.Записать ();

Для проверки работоспособности созданной обработки следует перейти в режим 1С:Предприятие и открыть только что созданную форму. Выбрав в поле списка одну из имеющихся заявок и выполнив процедуру обработки щелчка по кнопке **ОК**, мы получим новый документ Заказ (рис. 2.18).

Обработка событий

Наиболее существенную составляющую прикладных решений на платформе 1С:Предприятие составляет обработка событий. В этом разделе мы рассмотрим обработку типичных событий, которые могут встретиться на практике.

Воспользовавшись событием приЗаписи, можно отменить запись, что приведет к тому, что документ не будет записан в базу данных. Например, мы можем программно запретить запись документов, если указана дата ранее текущей.

Событие приЗаписи определено для модуля объекта. Поэтому в окне редактирования документа заказ нужно перейти на закладку **Прочее** и воспользоваться кнопкой **Модуль объекта**. Здесь и необходимо создать текст процедуры (листинг 2.7). В этом случае мы проверяем дату записываемого документа. Если эта дата меньше текущей даты, то происходит отмена записи информации в базу данных.

```
Листинг 2.7. Процедура обработки события ПриЗаписи
```

```
Процедура ПриЗаписи (Отказ)
Если ЭтотОбъект.Дата < ТекущаяДата() Тогда
Отказ = Истина;
КонецЕсли;
КонецПроцедуры
```

Еще одно событие — передудалением. С помощью его обработки можно оставить необходимые документы (не допустить их удаления из базы данных). Так, с помощью процедуры, приведенной в листинге 2.8, производится запрет удаления тех документов, дата которых ранее 5 августа 2008 года. Рассматриваемую процедуру также необходимо расположить в модуле объекта.

Листинг 2.8. Процедура обработки события ПередУдалением

Процедура ПередУдалением(Отказ) Если ЭтотОбъект.Дата < Дата(2008,08,05) Тогда Отказ = Истина; КонецЕсли; КонецПроцедуры

глава 3



Регистры накопления

Основное назначение *регистров накопления* просматривается уже из названия — они предназначены для реализации механизма количественного учета информации. Когда в конфигураторе мы создаем новый регистр накопления (далее в этой главе это будут конкретные регистры под именами информация ОТоварах и Заявки), то в программной памяти создается структура данных для реализации механизма накопления численных показателей.

Ручное внесение информации в регистры накопления, как правило, исключается. Они не предназначены для интерактивного редактирования.

Примечание

Вообще говоря, редактирование содержимого регистров накопления возможно, однако стандартная работа с регистрами накопления заключается в модификации их содержимого с помощью алгоритмов обработки.

Основной режим работы по добавлению данных (а также модификации и удалению) в регистры накопления связан с действиями, выполняемыми в программных процедурах, относящихся к другим объектам. Часто, даже можно сказать в основном, этими объектами являются документы.

Данные, хранящиеся в регистрах накопления, делятся на *измерения* и *ресурсы*. Накопление количественной информации может производиться в разрезе нескольких измерений, которые описываются при разработке конфигураций. Например, в качестве измерений может выступать склад и номенклатура товаров.

Виды числовой информации, аккумулируемой регистром накопления, называются *ресурсами*. Они являются подчиненными объектами и описываются в конфигураторе. Так, регистр накопления может накапливать информацию о количестве товаров на складах.

Далее в этой главе рассматривается разработка конфигурации, в которой мы познакомимся с программными приемами внесения информации в регистры накопления и извлечения из них необходимых данных при построении отчетов.
Регистр накопления остатков

Для демонстрации функциональности регистров накопления создадим несложную конфигурацию. Она содержит справочники фирмы и номенклатура, которые не имеют дополнительных реквизитов и полностью соответствуют тем, которые были использованы в *главе 2*.

Также создадим документ поступление. В связи с тем, что технология разработки нам уже знакома, подчеркнем только наиболее информативные моменты выполнения практических действий. В области шапки документа разместим один дополнительный реквизит — Заказчик (тип данных — СправочникСсылка.Фирмы). Для табличной части документа выберем имя перечень и внесем в нее два реквизита:

- Товар (ТИП ДАННЫХ СправочникСсылка. Номенклатура);
- количество (тип данных число с точностью до одной десятой) для внесения количества поступивших товаров.

После работы в конфигураторе следует перейти в режим 1С:Предприятие и заполнить несколько документов.

Далее создадим новый объект конфигурации — регистр накопления информацияотоварах (рис. 3.1). Важный момент, на который следует обратить внимание на этом рисунке, — поле со списком **Вид регистра**, в котором следует установить вид регистра накопления — **Остатки**. Далее мы разберем смысл этой установки и ее альтернативы — **Обороты**.

После этого в окне редактирования объекта конфигурации необходимо перейти на закладку Данные и создать для данного регистра:

- измерение Организация (тип данных СправочникСсылка.Фирмы);
- измерение товар (тип данных СправочникСсылка. Номенклатура);
- ресурс количество (в качестве типа данных выберем число).

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🗙						
<u>Ф</u> айл <u>П</u> равка Конфи	гурация Отладка	Администрирование <u>С</u> ервис	<u>О</u> кна			
Справка						
I 🗋 🛋 🖬 🛛 🔏 🖻	🖻 🧉 🔤	C M M	» •			
🕼 Регистр накоплен	ния Информация	ОТоварах	_ 🗆 ×			
• Основные						
Данные	Имя:	Информация U Говарах				
Регистраторы	Синоним:	Информация о товарах				
Формы	Комментарий:					
Макеты						
Подсистемы						
Права	Вид регистра	Остатки	-			
Интерфейсы						
Действия 👻	<Назад Дал	nee> Закрыть Cr	равка			

Рис. 3.1. Окно редактирования регистра Информация ОТоварах

На рис. 3.2 показано заполнение закладки Данные в соответствии с описанной структурой создаваемого регистра накопления. Однако сразу после этого переход в режим 1С:Предприятие для работы с созданным регистром не имеет смысла — необходимые действия в конфигураторе еще не завершены. Дело в том, что создание регистра накопления без документов, которые с ним взаимодействуют (формируют движения по регистру), не принимается информационной системой. При попытке сохранения конфигурации вы увидите на экране соответствующее диагностическое сообщение системы 1С:Предприятие.

Для устранения этой недоработки следует исправить документ поступление, в котором с помощью программной процедуры нужно реализовать выполнение движений по созданному регистру информацияОтоварах. Все записи, фиксирующие заказы, будут выполняться "незаметно" для пользователя в процессе записи документа Поступление.



Рис. 3.2. Структура регистра накопления Информация ОТоварах

Примечание

При записи документа осуществляется и его проведение — движение по регистру. Таким образом, для пользователя каких-либо новых действий в режиме 1С:Предприятие не требуется — все обеспечивает традиционный щелчок по кнопке **ОК** в окне формы документа.

Изменения в документе Поступление

В предыдущей главе мы создали аналогичный документ, и здесь нам потребуется его доработать. В режиме конфигуратора откроем окно редактирования объекта конфигурации и займемся дополнительными изменениями в документе. А именно перейдем на закладку **Движения** (рис. 3.3), где от нас требуется определить, в каких регистрах будут происходить изменения при проведении документа Поступление. В разделе **Регистры накопления** отражаются все имеющиеся регистры накопления, по которым можно организовать движения. В данном случае в нашей конфигурации мы располагаем только одним регистром накопления — информацияОтоварах, и только он присутствует в окне. От нас требуется поставить галочку напротив его имени.

🚵 Конфигуратор (учебная	а версия) - Конфигурация	
<u>Ф</u> айл <u>П</u> равка Конфигурац	ция Отладка Администриро	вание <u>С</u> ервис <u>О</u> кна
Справка		
🗋 📫 🔛 🐰 🖻 🛍	a 🗟 5 C 🌆 M	» •
🐱 Документ Поступлени	ie	_ 🗆 ×
Основные	Проведение:	Разрешить 💌
Данные	Оперативное проведение:	Разрешить 💌
Нумерация	Историтирное проведение.	
• Движения	Эдаление движении.	Эдалять движения (•
Последовательности	🗄 🚰 Регистры накопл	ения
Журналы	термац	иястоварах
Формы		
Макеты		
Подсистемы		
Ввод на основании	🝡 Конструктор движе	ний 🗧
Права	ИнформацияОТоварах	
Интерфейсы		
Действия 🔻 <На	зад Далее> З	акрыты Справка

Рис. 3.3. Закладка Движения окна редактирования документа Поступление

Примечание

Система 1С:Предприятие позволяет организовать движения документа по нескольким регистрам, и в книге [7] рассмотрен ряд практических примеров разработок на эту тему.

Мы подошли к следующему этапу разработки, и далее можно пойти двумя путями: либо самостоятельно написать процедуру, которая будет формировать движения при проведении подобных документов, либо воспользоваться возможностями конструктора движений. Соответствующая кнопка активизируется сразу после указания конкретного регистра (рис. 3.3). Выберем второй путь и, используя конструктор движений, автоматически получим необходимую программную процедуру в результате несложного диалога. Подобным образом в предыдущих главах мы уже несколько раз создавали печатные формы справочников и документов.

В режиме диалога с конструктором движений ключевым (для построения процедуры) является представленное на рис. 3.4 окно, которое стоит прокомментировать. В разделе **Регистры** указаны те регистры, по которым можно формировать движения в документе. В данном случае регистр у нас один — Информация ОТоварах, который и выбран для дальнейших действий конструктора.

Конструктор движения регистров			□ ×
월 불 철 ↑ ↓			
Регистры	Реквизи	гы документа	
🕈 РегистрНакопления.ИнформацияОТоварах	🕳 Дата	i i i i i i i i i i i i i i i i i i i	
	💻 Ном	ер	
	🞽 Зака	зчик	
Тип движения регистра: 💿 Приход 🛛 Расход			
Табличная часть:		< <u>Н</u> азад Дале	ee>
Поле Выражение		<u>З</u> аполнить выражен	ния
🛴 Организация			40
↓ Товар		Очистить выражен	ия
🧯 Количество			
		ОК	
		Отмена	
		Справка	

Рис. 3.4. Окно конструктора движений регистра

В правое верхнее поле автоматически уже внесены реквизиты шапки данного документа. Это связано с тем, что в указанное поле включаются параметры документа, которые могут участвовать в движениях.

Но кроме шапки в документе имеется еще и табличная часть, и для добавления реквизитов табличной части в поле **Реквизиты** документа следует выполнить несложную операцию. В центральной части рис. 3.4 присутствует поле со списком (с соответствующей подписью — **Табличная часть**), где перечислены имеющиеся табличные части в документе (в нашем документе табличная часть одна — перечень). Для перенесения в поле **Реквизиты** документа реквизитов табличной части необходимо выбрать ее имя в списке табличных частей (рис. 3.5).

На этом работа с данным окном еще не завершена, и нам следует перейти к левой нижней части окна конструктора движений. Здесь необходимо указать соответствие полей регистра накопления и реквизитов документа. Технически это можно сделать либо с помощью кнопки Заполнить выражения, либо с помощью двойного щелчка мышью по необходимому реквизиту документа в правом верхнем разделе на рис. 3.5. После этого напротив названий полей регистра появятся соответствующие выражения (рис. 3.6).

Теперь все действия в окне конструктора движений регистров выполнены, и осталось щелкнуть по кнопке **OK**. Это приводит к генерации конструктором программной процедуры (листинг 3.1), которая формирует движения по регистру при проведении документа. В дальнейшем, если потребуется эту процедуру найти (скажем, для коррекции или просто просмотра), в окне редактирования объекта конфигурации следует перейти на закладку **Прочее**. Здесь после нажатия кнопки **Модуль объекта** мы увидим текст, представленный в листинге 3.1.

Конструктор движения регистров			□ ×
😫 🛬 🦄 ↑ ↓ Регистры (← РегистрНакопления Информация0Товарах		ты документа а ер ізчик СтрокаПеречень СтрокаПеречень	ь.Товар ь.Количество
Тип движения регистра: • Приход • Расход			
Табличная часть: Перечень 💌		< <u>Н</u> азад	<u>Д</u> алее>
Поле Выражение		<u>З</u> аполнить	выражения
↓ Организация ↓ Товар И Количество		Оуистить в	зыражения
		0	IK
		Отм	1ена
		Спр	авка

Рис. 3.5. Отображение реквизитов табличной части в окне конструктора

Конструктор движ	ения регистров			□ ×
월호철↑↓				
Регистры		Реквизи	ты документа	
🔄 РегистрНакопле	ния.ИнформацияОТоварах	🗕 Дата	3	
		🗕 Ном	ер	
		🔏 Зака	азчик	
		🚛 Тек(СтрокаПеречені	ь.Товар
		🚛 Текί	СтрокаПеречені	ь.Количество
т				
і ип движения регис	тра: • Приход • Расход			
Табличная часть: [Теречень 💌		< <u>Н</u> азад	<u>Д</u> алее>
Поле	Выражение		<u>З</u> аполнить	выражения
🛴 Организация	Заказчик		0	
🛴 Товар	ТекСтрокаПеречень.Товар		Очистить	выражения
🧯 Количество	ТекСтрокаПеречень.Количество)		
			ж	
			Оты	иена
			Спр	авка

Рис. 3.6. Заполнение выражений в окне конструктора движений регистра

Обратим внимание еще на одну важную деталь в окне конструктора движений регистров на рис. 3.6. В качестве значения параметра **Тип движения регистра** по умолчанию с помощью переключателя установлен вариант **Приход**. Это соответствует добавлению ресурсов регистра при проведении документа. Альтернативный вариант **Расход** приводит к уменьшению ресурсов.

Листинг 3.1. Процедура, выполняемая при проведении документа Поступление Процедура ОбработкаПроведения (Отказ, Режим) //{{ КОНСТРУКТОР ДВИЖЕНИЙ РЕГИСТРОВ // Данный фрагмент построен конструктором. // При повторном использовании конструктора внесенные вручную // изменения будут утеряны!!! Для Каждого ТекСтрокаПеречень Из Перечень Цикл // регистр ИнформацияОТоварах Приход Движение = Движения.ИнформацияОТоварах.Добавить (); Движение.ВидДвижения = ВидДвиженияНакопления.Приход; Движение.Период = Дата; Движение. Организация = Заказчик; Движение.Товар = ТекСтрокаПеречень.Товар; Движение.Количество = ТекСтрокаПеречень.Количество; КонецЦикла; //}} КОНСТРУКТОР ДВИЖЕНИЙ РЕГИСТРОВ КонецПроцедуры

Основное содержание процедуры ОбработкаПроведения () составляет цикл, в котором перебираются строки табличной части Перечень документа ИнформацияОТоварах: Для Каждого ТекСтрокаПеречень Из Перечень Цикл

При каждом проходе цикла с помощью метода добавить () создается очередная запись в регистре. В последующих строках процедуры устанавливаются значения полей регистра в соответствии с данными из очередной строки табличной части документа.

Работа с конструктором движений завершена, и можно подчеркнуть главный результат, заключающийся в том, что вследствие несложных визуальных действий в документе Поступление реализовано формирование движений по регистру накопления (информацияОТоварах). Автоматически созданную программную процедуру ОбработкаПроведения () в дальнейшем можно при необходимости скорректировать либо дополнить.

Сейчас стоит перейти в режим 1С:Предприятие и проверить результат выполненных изменений (протестировать созданный документ). В меню Операции выберем пункт Документы, а затем в выпадающем меню выберем пункт, соответствующий только что скорректированному документу — Поступление. С помощью соответствующей кнопки пиктографического меню можно создать документ и заполнить его информацией (например, как показано на рис. 3.7). Теперь щелчок по кнопке **OK** приводит к записи созданного документа в информационную базу. Кроме того, осуществляется и его проведение (формируются движения по регистру). На рис. 3.8 показана форма списка регистра накопления с внесенными записями, связанными с документом.

Данной набор записей называется *движениями* по регистру. Каждая запись содержит информацию о регистраторе (это документ, который вызвал движение). Также каждая запись содержит значения измерений и значения приращений ресурсов.

🏪 1С:Пред	приятие (учебная версия) -	Конфигур	ация		
<u>∎ Ф</u> айл <u>П</u> ра	авка Операции <u>С</u> ервис <u>О</u> кна	а Справка			
I 🗋 📫 🛯	X 🖻 🛍 🧉 🐥	🎹 🗎 🐔	M M+	м- 🎘 🖕	
🐱 Заказ:	Поступление (Новый) *			_ 🗆 ×	
Действия 🕶	- 🛃 🔯 💽 🎦 🌆 Пер	рейти 🕶 ?)		
Номер:	00000025				
Дата:	01.09.2008 0:00:00				
Заказчик:	Юниор			🗙	
물통	🕺 🔄 🖬 🏌 🖡 🛃 🛃				
N T	овар	Количеств)		
1 N	Іягкая мебель			7	
2 K	(ухня Уют-З			5	
30	Стенка (мод. 15-7)			12	
		OK	Записать	Закрыты	
📇 Документы Поступление 🛛 🐱 Заказ: Поступление (Новы					
Для получен	ния подсказки нажмите F1		CAP	NUM //	

Рис. 3.7. Заполнение формы документа Заказ в режиме 1С:Предприятие

	1С:Предприятие (у	учебная версия	а) - Конфигурация			
≣ .	<u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка					
Ē	🗋 🛋 🖬 🛛 🔏 🖻	66	5 C M		▼ 🙀 🐥 📰	🛅 🌌 М М+ М- 🎇 🖕
	🕆 Список Информация о товарах 🛛 _ 🗆 🗙					
	Действия 🖌 [++) 🎢	🎦 🖷 - 🌾	2 ?			
	Регистратор	Номер Акти	Период 🔹	Организация	Товар	Количество
	+ Поступление 0	1 🗸	01.09.2008 15:44:24	Юниор	Мягкая мебель	7
	+ Поступление 0	2 🗸	01.09.2008 15:44:24	Юниор	Кухня Уют-З	5
	+ Поступление 0	3 🗸	01.09.2008 15:44:24	Юниор	Стенка (мод. 15-7)	12
2	📇 Документы Поступление 🛛 👔 Список Информация о това					
Д	ля получения подсказ	ки нажмите F1				CAP NUM

Рис. 3.8. Форма списка регистра для регистрации поступлений

Расход по регистру накопления

Мы рассмотрели ситуацию, когда документ обеспечивает добавление ресурса регистра накопления. Для полноты картины создадим документ Отгрузка, который будет уменьшать ресурс регистра. Данный документ (рис. 3.9) по своей структуре похож на уже разработанный нами документ Поступление.



Рис. 3.9. Закладка Данные окна редактирования документа Отгрузка

После создания этого документа следует перейти на закладку Движения и здесь с помощью конструктора (рис. 3.10) подготовить создание процедуры обработки (листинг 3.2) проведения документа. Важно, что в данном случае в качестве типа движения регистра указан Расход.

```
Листинг 3.2. Процедура, выполняемая при проведении документа Отгрузка
```

```
//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

На рис. 3.11 показана форма списка регистра накопления с внесенными записями, связанными с документом Отгрузка.

Конструктор движен	ия регистров			□ ×
불불불↑↓				
Регистры		Реквизи	ъ документа	
듣 РегистрНакоплени	я.ИнформацияОТоварах	🗕 Дата	-	
			эр	
		🔏 Фир	ма	
		🚛 ТекС	трокаПеречены	Товар
		🚛 ТекС	трокаПеречены	ь.Количество
Тип движения регистр	а: 🔿 Приход 💿 Расход			
Табличная часть: Пе	речень		< <u>Н</u> азад	<u>Д</u> алее>
Поле	Выражение		Заполнить	выражения
∔ Организация	Фирма		-	
ե Товар	ТекСтрокаПеречень.Товар		Очистить і	выражения
🧃 Количество	ТекСтрокаПеречень.Количество			
			0	IK
			Отм	иена
			Спр	авка

Рис. 3.10. Окно конструктора движений документа Отгрузка

🎨 1С:Предприятие (учебная версия) - Конфигурация								
🗄 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка								
 I 🗋 🖬 🕹 📾 📾 🖾 🗢 😋 🛤 🔜 🐨 🖓 💭 🏧 🎬 🖉 🖉 M M+ M- 😕 🧅								
🗜 Список Информация о товарах 💶 🗆 🗙								
Дей	ствия 🕶 🔂 🕅	1 10 1	F• 74	2 ?				
	Регистратор	Номер	Активно	Период 🔹	Организация	Товар	Количество	^
+	Поступлени	1	~	01.09.2008 15:44:24	Юниор	Мягкая мебель		7
+	Поступлени	2	×	01.09.2008 15:44:24	Юниор	Кухня Уют-З		5
+	Поступлени	3	~	01.09.2008 15:44:24	Юниор	Стенка (мод. 15-7)		12
-	Отгрузка 00	1	~	01.09.2008 16:32:37	Тико	Мягкая мебель		1
-	Отгрузка 00	2	~	01.09.2008 16:32:37	Тико	Стенка (мод. 15-7)		1
-	Отгрузка 00	3	~	01.09.2008 16:32:37	Тико	Кухня Уют-З		3
Документы Отгрузка Ге Список Информация о това Для получения подсказки нажмите F1 САР NUM								

Рис. 3.11. Форма списка регистра Информация ОТоварах

Оборотный регистр накопления

В ряде ситуаций нет необходимости учитывать остатки. Например, в случае подачи заявок на продукцию представляет интерес подсчет заявок за определенный интервал. Или, скажем, наша организация предоставляет образовательные услуги. В этом случае важны услуги, оказанные за определенный период. Для подобных ситуаций предназначены оборотные регистры накопления.

В общем, оборотные регистры накопления и регистры остатков отличаются не очень сильно. Оборотные регистры накапливают обороты, а остатки в них не учитываются. В этом случае система создает только одну виртуальную таблицу — таблицу оборотов. Фактически это единственное отличие оборотного регистра.

В предыдущей главе мы создали документ Заявка. И, используя данный документ, продемонстрируем работу с регистрами оборотов. Итак, в конфигураторе приступим к разработке нового регистра. На рис. 3.12 отражен первый шаг.

Важно на закладке **Основные** сделать установку в поле **Вид регистра** — **Обороты**. После этого в окне редактирования объекта конфигурации необходимо перейти на закладку **Данные** и создать для данного регистра:

- измерение Фирма (тип данных СправочникСсылка. Фирмы);
- измерение Товар (тип данных СправочникСсылка. Номенклатура);
- ресурс количество (в качестве типа данных выберем число).

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🗙						
<u>Ф</u> айл <u>П</u> равка Конфи	<u>Ф</u> айл Правка Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна					
🛛 Справка						
🗋 🚄 🔛 🔏 🖻	🖻 🧟 ち	c 🗿 🙀	» •			
👍 Регистр накопле	ния Информация	ОЗаявках _	. 🗆 🗙			
• Основные	Muo:	Mutan				
Данные	KIMH.	информация Сраявках				
Регистраторы	Синоним:	Информация о заявках				
Формы	Комментарий:					
Макеты						
Подсистемы						
Права	Вид регистра	Обороты	-			
Действия 🔻 <Н	азад Далее	» Закрыты Спр	авка			

Рис. 3.12. Закладка Основные окна редактирования регистра накопления

На рис. 3.13 показано заполнение закладки Данные в соответствии с указанной структурой создаваемого регистра накопления.

Теперь перейдем к следующему этапу, который связан с формированием движений по регистру. Как и ранее, для этого воспользуемся конструктором. Поэтому в окне редактирования документа Заявка на закладке Движения (рис. 3.14) следует указать (с помощью флажка) только что созданный регистр. Далее, используя конструктор движений, надо проделать уже знакомые шаги (рис. 3.15), что приведет к автоматическому созданию процедуры (листинг 3.3).



Рис. 3.13. Структура регистра Информация ОЗаявках

🛃 Конфигуратор (учебная	я версия) - Конфигурация				
<u>Ф</u> айл <u>П</u> равка Конфигура.	ция Отладка Администриров	зание <u>С</u> ервис <u>О</u> кна			
Сп <u>р</u> авка					
D 🛋 🖬 X 🖻 🛍	a 🗟 5 C 🖪 M	» •			
🗟 Документ Заявка 📃 🗆 🗙					
Основные	Проведение:	Разрешить 👻			
Данные					
Нумерация	оперативное проведение.				
• Движения	Удаление движений: 	Удалять движения 🔻			
Последовательности	🗆 🚰 Регистры накопле	ения			
Журналы	— Ца Информаци	ияОТоварах			
Формы	⊻ 👣 Информаци	яр заявках			
Макеты					
Подсистемы	🐂 Конструктор движен	ний 🗧			
Ввод на основании	ИнформациоВ Задеках				
	ANY OPPOSITION OUNDER				
Действия - (Наз	ад Далее> Зан	(рыть Справка			

Рис. 3.14. Закладка Движения окна редактирования документа Заявка

Листинг 3.3. Процедура, выполняемая при проведении документа Заявка

```
// регистр ИнформацияОЗаявках
Движение = Движения.ИнформацияОЗаявках.Добавить();
Движение.Период = Дата;
Движение.Фирма = ФирмаЗаказчик;
Движение.Товар = ТекСтрокаПереченьТоваров.Номенклатура;
Движение.Количество = ТекСтрокаПереченьТоваров.Количество;
КонецЦикла;
```

КонецПроцедуры

Конструктор движе	ния регистров			□ ×
열 날 철 ↑ ↓ Регистры ● РегистрНакоплен	ия Информация0Заявках	Реквизит Дата Номе Фирт ТекС ТекС	ы документа р иаЗаказчик трокаПеречени трокаПеречени	ъТоваров.Но ъТоваров.Кол
Табличная часть: 🔲	ереченьТоваров 💌		< <u>Н</u> азад	Далее>
Поле	Выражение		Заполнить	выражения
🛴 Фирма	ФирмаЗаказчик			
🛴 Товар	ТекСтрокаПереченьТоваров.Ног	менклат	О <u>ч</u> истить	выражения
🧯 Количество	ТекСтрокаПереченьТоваров.Кол	ичество		
				К
			Отр	иена
			Спр	авка

Рис. 3.15. Окно конструктора движений регистра

🏪 1С:Предприятие (учебная версия) - Конфи	урация				
🛛 🕂 Файл Правка Операции Сервис Он	на Сп <u>р</u> ав	ка				
I 🗋 🔟 🐰 🖕 👘 🌆 🚳 🔯 🗠	bdla	4	▼ # *	🕯 🕒 🗳 🖩 🛅	👷 🛛 м м+	м- 🎘 🖕
👫 Список Информация о заявках						_ 🗆 🗙
Действия 🕶 [++) 🎢 🖆 🌃 - 🌾 🛛) ?					
Periverperen Heven	Avenueu		(thursus	Toppo	Kagunagraa	
 Заявка 00000002 от 01.0. 	. Активн 1 🐱	Период 01.09.2008.16:40:	тико	товар Магкаа мебель	КОЛИЧЕСТВО	5
 Заявка 00000002 от 01.0 	2 🗸	01.09.2008 16:40:	Тико	Стенка (мод. 15-7)		3
 Заявка 00000003 от 03.0 	1 🗸	03.09.2008 11:13:	Меда	Стенка (мод. 15-7)		7
		-	-	· · · · ·	1	
Список. Информация о зая Для получения подсказки нажмите F1	ументы За	явка			CAP	NUM

Рис. 3.16. Список зарегистрированных заявок

Теперь можно в режиме 1С:Предприятие внести одну или несколько заявок и посмотреть на содержание регистра (рис. 3.16).

Получение выборки информации из регистра

Рассмотрим программный доступ к информации, содержащейся в регистрах накопления. Выборка из регистра производится с помощью метода Выбрать (). Данный метод имеет четыре параметра. С их помощью можно установить период выборки, условие отбора и способ сортировки записей.

Аналогично примерам предыдущей главы для демонстрации работы мы создали обработку и на форме обработки (рис. 3.17) расположили поле списка, которое назвали переченьдокументов. В листинге 3.4 приведен пример процедуры для извлечения информации из рассматриваемого в этой главе регистра Информация0Заявках.

🝌 Конфигуратор (учебная версия) - Конфигурация	- 1	
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка		
I 🗋 🖬 😹 📾 📾 🔯 🖆 📾 🛤 📃 🐨 🖓 🖄 🎘 ?	in []) *
🚟 Обработка ПолучениеВыборки: Форма	_ C	×
Обработка Получение выборки		- 1
Действия - ?		- 1
Перечень документов:		
		- 1
💭 Выполнить Закрыт	гь	
р Для получения подсказки нажмите F1 САР NUM ги → +±0, 0 ↓±568 x 2	220	

Рис. 3.17. Форма обработки для формирования перечня документов

На рис. 3.18 показан результат выполнения процедуры извлечения информации из регистра накопления.

Листинг 3.4. Процедура извлечения информации из регистра накопления

```
Процедура КнопкаВыполнитьНажатие(Кнопка)
РегистрЗаказ = РегистрыНакопления.ИнформацияОЗаявках;
Данные= РегистрЗаказ.Выбрать();
ПереченьДокументов.Очистить();
```

```
Пока Данные.Следующий() Цикл
```

```
Информация ="Регистратор: " + Данные.Регистратор +
```

- " Номенклатура: " + Данные.Товар +
 - " Количество: " + Данные.Количество;

ПереченьДокументов.Добавить (Информация);

КонецЦикла;

КонецПроцедуры

🐫 1С:Предприятие (учебная версия) - Конфигурация 📃	
🗄 🙅 айл Правка Операции Сервис Окна Справка	
I 🗋 🔟 📓 🔏 📾 🥌 🗢 🕻 א 👋 📲 📰 🖉 м м+ м-	℀ 🗸
🗳 Обработка Получение выборки 💶 [⊐ × □
Действия - ?	
Перечень документов: Регистратор: Заявка 000000002 от 01.09.2008 16:40:44 Номенклатура: Мягкая мебель Количество: 5 Регистратор: Заявка 000000002 от 01.09.2008 16:40:44 Номенклатура: Стенка (мод. 15-7) Количество: 3 Регистратор: Заявка 000000003 от 03.09.2008 11:13:49 Номенклатура: Стенка (мод. 15-7) Количество: 7	
Выполнить Зак	рыть
3 ^о Обработка Получение выб	
Для получения подсказки нажмите F1 САР NUM	1 //

Рис. 3.18. Результат выборки информации из регистра накопления

Получение остатков

Регистр накопления информацияОтоварах, который мы создали, является регистром остатков. Это говорит о том, что вместе с таблицей движений формируется еще и таблица остатков. Программно можно легко извлечь данную информацию (посмотреть, сколько каких товаров осталось).

Для начала создадим обработку с формой (рис. 3.19), на которой разместим поле ввода товар (тип данных — СправочникСсылка.Номенклатура). Пользователь должен указать в этом поле название товара, по которому его интересует остаток. По щелчку на кнопке **Выполнить** должна выполняться процедура, приведенная в листинге 3.5.

Листинг 3.5. Получение остатков товаров по названию

Процедура КнопкаВыполнитьНажатие(Кнопка) РегистрЗаказОстатки = РегистрыНакопления.ИнформацияОТоварах; ОтборПоТовару = Новый Структура();

```
ОтборПоТовару.Вставить ("Товар", Товар);
Данные = РегистрЗаказОстатки.Остатки (,ОтборПоТовару);
Если Данные.Количество()>0 Тогда
Товары = Данные.Итог("Количество");
Сообщить ("Количество данных товаров " + Товары);
КонецЕсли;
КонецПроцедуры
```

🙏 Конфигиратор (ичебная версия) - Конфигирания
<u>Файл</u> Правка Форма Конфигурация Отладка Администрирование Сервис
Окна Справка
I 🗋 💶 X 🖻 🛍 📾 🗹 🕨 🖝 🛤
🚟 Обработка ПолучениеОстатов: Форма 💶 🗖 >
Обработка Получение остатов
Действия - ?
Товар: х
💭 Выполнить Закрыты
🔜 Диалог 🗏 Модуль 📑 Реквизиты
Для получения подсказ САР NUM ru + 10, 0 + 400 x 224

Рис. 3.19. Форма обработки для получения остатков

Для получения остатков регистра накопления предназначен метод Остатки(). Он имеет четыре необязательных параметра: момент времени, критерий отбора, измерения и ресурсы. В данном случае мы не указали момент времени, поэтому получили остатки по указанному товару на текущую дату. Метод Итог() позволяет извлечь итог по остаткам.

Примечание

Для вывода информации в рассматриваемом примере мы воспользовались функцией Сообщить (), которая выводит текст сообщения в окно сообщений. Если в момент вызова окно сообщений отсутствует, то будет открыто новое окно сообщений.

Как итог, на рис. 3.20 приведен результат применения данной процедуры к имеющимся данным в информационной базе. Таким образом, мы легко можем получать информацию по остаткам конкретного товара.

陽 1С:Предприятие (учебная версия) - Конфигурация 📃 🗆 🗙
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
III 🗋 📓 🔏 🖷 🦉 🔭 🖬 🖬 🖉 м м+ м- 🞇 🖕
💣 Обработка Получение остатов 🛛 💶 🗙
Действия 🕶 🕐
Товар: Стенка (мод. 15-7) 🗙
Выполнить Закрыть
Служебные сообщения 🛛 🕂 🗙
 Количество данных товаров 11
×>
🕹 Обработка Получение оста
Для получения подсказки нажмите F1 САР NUM

Рис. 3.20. Информация по остатку товара

глава 4



Регистры сведений

Регистры сведений — это новые объекты конфигурации, которые появились в версии программного продукта 1С:Предприятие 8 (в широко распространенной в нашей стране системе 1С:Предприятие 7.7 их не было). Примеры данной главы преследуют цель познакомить читателей с практическими приемами работы с данными объектами. В главе 5 будет рассмотрена достаточно сложная конфигурация с их использованием. И они в ней будут играть ключевую роль.

Как и ранее, в процессе выполнения практических действий по тексту главы будет приводиться необходимая справочная информация, дающая представление об используемых программных конструкциях на встроенном языке 1С.

Организация непериодических регистров сведений

Наша задача заключается в том, чтобы создать конфигурацию, которая позволит продемонстрировать визуальные и программные действия с регистрами сведений. Рассмотрим ситуацию, когда в организации имеется несколько филиалов, каждый из которых включает несколько подразделений. И в каждом конкретном подразделении филиала определено ответственное лицо. Таблица 4.1 отражает данную ситуацию.

Филиал	Подразделение	Ответственный за поставки
Окский	1	Петров Л. В.
Окский	2	Иванов О. В.
Волжский	1	Сидорова О. М.
Волжский	3	Николаев Л. Т.

Таблица 4.1. Информация об ответственных лицах

Комбинация названия филиала и номера подразделения является уникальной, и она представляет ключ при записи либо извлечении информации об ответственном

за поставки. Совокупности названия филиала и номера подразделения соответствует не более чем один ответственный за поставки (это функциональная зависимость, если проводить аналогию с математикой).

Для хранения информации о филиалах создадим новое перечисление с соответствующим названием — Филиалы. В режиме конфигуратора на знакомой закладке Данные заполним созданное перечисление данными (рис. 4.1). Также добавим справочник сотрудники без дополнительных реквизитов и в режиме 1С:Предприятие внесем в него несколько фамилий.

🛃 Конфигуратор (з	учебная версия) - Конфигурация
<u>Ф</u> айл <u>П</u> равка Кон	юригурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна
Справка	
🗋 🛋 🖬 🐰 🛛	🚡 📾 🥌 🖸 🖕 🖬 м 🛛 🐥
🔛 Перечисление	Филиалы _ 🗆 🗙
Основные	Значения перечисления:
▶ Данные	월 국 철 ★ ↓ 문
Формы	
Макеты	⊟™ Значения
Подсистемы	Воджский
	— Центральный
Деиствия •	назад далее> Закрыть Справка
 [] Перечисление Фи	лиалы
Для получения подск	азки нажмите F1 САР NUM ги -

Рис. 4.1. Окно редактирования перечисления Филиалы

Теперь перейдем непосредственно к разработке регистра сведений. Начальные действия по созданию объекта конфигурации нам уже знакомы — щелчок правой кнопкой мыши на названии **Регистры сведений** в окне **Конфигурация** и выбор пункта **Добавить** в контекстном меню. На закладке **Основные** (рис. 4.2) кроме имени (ответственныелица) регистра следует указать, что он является *непериодическим*. Это означает, что *время* в качестве дополнительного аргумента в создаваемом регистре использоваться не будет. В дальнейшем мы рассмотрим и пример создания периодического регистра сведений, однако сейчас нам это не требуется.

На той же закладке (см. рис. 4.2) в процессе создания регистра укажем **Незави**симый режим записи — в этом случае мы можем напрямую производить запись информации в регистр сведений. В противоположность этому при указании режима **Подчинение регистратору** появляется возможность формировать записи в регистре автоматически в результате обработки информации из документов.

Теперь перейдем на закладку Данные окна редактирования объекта конфигурации, где укажем измерения и ресурсы регистра в соответствии с табл. 4.1. Первое измерение — Филиал, для которого определим тип — Перечисление Ссылка. Филиалы. Для второго измерения (Подразделение) выберем числовой тип данных. Ресурсом регистра назначим — Ответственный (тип — СправочникСсылка. Сотрудники). После этого перейдем в режим 1С:Предприятие и внесем несколько записей (рис. 4.3).

🛃 Конфигуратор (уче	бная версия) - Ко	нфигурация
<u>Ф</u> айл <u>П</u> равка Конфи	гурация Отладка А	Администрирование <u>С</u> ервис <u>О</u> кна
Справка		
I 🗋 🚄 🖬 🐰 🖻	🔒 🧉 🗟 🖕	c 🗿 🕺 🚽 🚽
👖 Регистр сведений	а Ответственные)	Лица _ 🗆 🗙
Основные		
Данные	Имя:	ОтветственныеЛица
Регистраторы	Синоним:	Ответственные лица
Формы	Комментарий:	
Макеты	Konnorn aprin.	
Подсистемы		
Права	Периодичность:	Непериодический
Интерфейсы	Режим записи:	Независимый
Обмен данными	0	
Прочее	Основной отоор	по периоду
Действия 👻	<Назад	Далее> Закрыть Справка
Для получения подсказк	и нажмите F1	CAP NUM ru -

Рис. 4.2. Окно редактирования регистра сведений ОтветственныеЛица

🏪 1C	🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🔲 🗙			×		
∏ <u>Ф</u> ай	іл <u>П</u> равка Опе	рации <u>С</u> ервис <u>О</u> кі	на Справка			
	🛋 🖬 🛛 🕹 🖻	• 🖻 🧉 🖕	2 📰 🛅 🧟 🛛 1	M M+	м- 🎘	•
í (Список Ответс	твенные лица			_ 🗆	×
Дей	ствия 🖌 🔤 불	e 🛃 🎽 🖦 🕅	``ēn ¶• 际 [2 ?		
	Филиал	Подразделение	Ответственный			~
	Окский	1	Петров Л.В.			
	Окский	2	Сидорова О.П.			
	Волжский	1	Орлова К.Д.		[
					E	~
🌈 Cni	исок Ответствен	ные лица				
Для п	олучения подска	ізки нажмите F1		CAP	NUM	///

Рис. 4.3. Внесение информации в регистр сведений ОтветственныеЛица

Создание справочника Номенклатура

Для дальнейшей работы нам потребуется вспомогательный справочник, который будет содержать названия товаров.

Его мы назовем номенклатура. Дополнительных реквизитов вводить не будем, а что касается напрашивающегося реквизита цена, то для этого мы организуем периодический регистр сведений цены. Это связано с тем, что необходимо хранить информацию обо всех изменениях цен на товары (на практике это важно, т. к. часто приходится выполнять действия "задним числом"). Таким образом, в отличие от предыдущих примеров, фактический параметр товара в справочнике будет храниться в другой информационной структуре (в регистре сведений).

Теперь, после создания справочника в конфигураторе, следует перейти в режим 1С:Предприятие и внести несколько товаров (рис. 4.4).

🏪 1С:Предприятие	(учебная версия) - Конфигурация 📃 🔲 🗙
🛛 <u>Ф</u> айл <u>П</u> равка Опе	рации <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
I 🗋 🛋 🖬 🛛 🕹 🖻	a 🛍 🥌 🐥 🖩 🎹 🛄 🧟 🛛 M M+ M- 🎘 🗸
🗐 Справочник Но	менклатура 🗕 🗆 🗙
Действия 🕶 🚖 🛬	🛋 😼 🕅 To 🗃 - 🌾 🛛 😧 🕐
Код	Наименование 🔹 🛆
- 000000001	Двигатель 234-7
- 00000002	Комплект 55а
- 000000003	КПП 76
П Справочник Номен	клатура
Для получения подска	зки нажмите F1 САР NUM

Рис. 4.4. Внесение информации в справочник Номенклатура

Документ Поступление

Для иллюстрации программной работы с регистром сведений создадим документ поступление. На рис. 4.5 показана закладка Данные окна редактирования объекта конфигурации.

В качестве типов данных реквизитов установим:

- 🔶 Филиал ПеречислениеСсылка.Филиалы;
- Подразделение Число;
- Ответственный СправочникСсылка.Сотрудники.

В табличной части документа выполним следующие установки:

- Название СправочникСсылка.Номенклатура;
- ♦ Цена Число;

📩 Конфигуратор (учебная в	ерсия) - Конфигурация 📃 🗆 🗙
🛛 <u>Ф</u> айл <u>П</u> равка Конфигурация	Отладка Администрирование <u>С</u> ервис <u>О</u> кна
Справка	
I 🗋 🛋 🖬 🐰 🖻 🛍 🧉	i 🖸 🕁 🖬 🛤 🧳 🌷
🐱 Документ Поступление	_ 🗆 ×
Основные	
▶ Данные	
Нумерация	В. В. Визиты
Движения	Попразделение
Последовательности	Ответственный
Журналы	
Формы	
Макеты	(월 일 등 월 수 대 팀 🔡 🚺
Подсистемы	
Ввод на основании	⊡
Права	———— Название
Интерфейсы	— Цена
Обмен данными	— Количество
Прочее	:= Uумма
Действия 👻 <Назад	Далее» Закрыть Справка
<u></u>	

Рис. 4.5. Окно редактирования документа Поступление

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🛛 🗙
Файл Правка Форма Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна Справка
□ 🔟 🖬 🔏 📾 📾 🗢 🗢 🗭 א 📃 👻 🍹
🚟 Документ Поступление: ФормаДокумента 💶 🗶
Поступление
Действия - 🛃 📧 💺 👔 🔀 ?
Номер:
Дата:
Филиал: х
Подразделение: 0
Ответственный: х
N Название Цена Количество Сумма
labi (llone beoga) labi (llone beoga) labi (llone beoga) labi (llone beoga)
p
💭 ОК Записать Закрыть
🗐 Диалог 🗏 Модуль 📑 Реквизиты

Рис. 4.6. Форма документа Поступление

- Количество Число;
- ♦ Сумма Число.

Для размещения программных процедур нам потребуется создать форму документа (рис. 4.6).

Процедура установки ответственного в поле документа

Учитывая имеющуюся информацию, в регистре сведений необходимо обеспечить заполнение поля ответственный после выбора филиала и подразделения в документе поступление. Для определенности произведем это действие при выборе в поле подразделение номера подразделения. Для этого в окне свойств поля подразделение воспользуемся событием приИзменении (рис. 4.7).

В соответствующей процедуре (листинг 4.1) производится запрос к регистру сведений Ответственныелица. Параметрами запроса являются филиал и Подразделение, которые в тексте запроса выделяются префиксом &. Для установки значений параметров используется метод установитьПараметр(). В результате выполнения запроса возможны две ситуации:

- имеется единственная запись об интересующем нас человеке;
- записи с указанными измерениями нет.

В первом случае мы записываем информацию в поле Ответственный.

Å	🛃 Конфигуратор (учебная версия) - Конфигурация								_		×
1	<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование <u>С</u> ерв				ис <u>О</u> кна Сп <u>р</u> ав	ка					
Ī	1) 🖬 🔏 🛛	b 6	🍯 🖻 🖕 C	PA PA		▼ Ĥ' Ĥ	🖻 📜 🚧	, 🖪 🖻		
	📟 Документ Поступление: ФормаДокумента				Свойства: По				×		
	Пост	упление					21 🛯 🎽 🗉	x v 👘			
	Дейст	гвия 👻 👍	0	š 🐴 🛃 🥐 🕻	0		ПорядокОбход	1a	9		^
							Активизирова	тьПоУмолчан	ию		
	Номе	sp:					▼ События:				
	Дата	c –		::		=	ПриИзменени	Подразделен	иеПриИзн 🔻	đ	
	Фил	иал:				x	НачалоВыбор			· Q	
							НачалоВыбор			đ	
	подр	азделение:			. .		Очистка			· Q	
	Отве	тственный:				×	Регулировани			· Q	
	2	눌 🔍 🎽	🖬 1)		Открытие		•	· Q	
	N	Название		Цена	Количество	Сумма	ОбработкаВы			· Q	
		аы «Поле і	ввода>	аы <Поле ввода>	аы <Поле ввода:	» 🔒 «Поле в	АвтоПодборТ			ď	
							ОкончаниеВво			· Q	
	7						Имя, по которо	ми билет выпо	пняться		Ň
							обращение к эл	ементу управ.	ления в		
l					💭 ок 💠	Записать Зан	алгоритме моду	иля			~

Рис. 4.7. Форма документа Поступление в режиме конфигуратора

Листинг 4.1. Процедура, выполняемая при выборе номера подразделения

Теперь перейдем в режим 1С:Предприятие и убедимся в работоспособности разработанной процедуры (рис. 4.8).

🏪 1С:Предприяти	е (учебная версия) - Конфигурация 📃 🗖 🗙						
<u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка							
I 🗋 📫 🖬 🕹	🖺 🛍 🥌 🙋 🤚 🔭 📲 🛅 🧟 м. м+ м- 🎇 🧅						
🐱 Поступление	:: Поступление (Новый) * 🛛 💶 🗙						
Действия 🕶 🛃	🔯 🍇 🔁 🛃 🥐						
Номер:	00000002						
Дата:	04.09.2008 0:00:00						
Филиал:	Окский х						
Подразделение:	1						
Ответственный:	Петров Л.В 🗙						
📇 Документы Пост	📇 Документы Поступление 🛛 🐼 По: Поступление (Новый) *						
Для получения подо	Для получения подсказки нажмите F1 САР NUM						

Рис. 4.8. Заполнение поля документа информацией из регистра сведений

Для придания разработке завершенности аналогичную процедуру необходимо выполнять и при выборе филиала (листинг 4.2), поскольку мы не можем гарантировать порядок заполнения полей пользователем.

Листинг 4.2. Процедура, выполняемая при выборе филиала

```
Процедура ФилиалПриИзменении(Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ *
| ИЗ РегистрСведений.ОтветственныеЛица
```

```
| ГДЕ Филиал=&Филиал И
| Подразделение =&Подразделение";
Запрос.УстановитьПараметр ("Филиал", ЭлементыФормы.Филиал.Значение);
Запрос.УстановитьПараметр ("Подразделение",
ЭлементыФормы.Подразделение.Значение);
Результат = Запрос.Выполнить ().Выбрать ();
Если Результат.Следующий () > 0 Тогда
ЭлементыФормы.Ответственный.Значение = Результат.Ответственный;
КонецЕсли;
КонецПроцедуры
```

Создание периодического регистра сведений Цены

Периодический регистр сведений обладает способностью хранить информацию в привязке ко времени. В этом случае в регистре сведений находятся и актуальные (последние) значения, и все предшествующие. Периодичность регистра сведений можно определить одним из следующих значений:

- в пределах секунды;
- в пределах дня;
- в пределах месяца;
- в пределах квартала;
- в пределах года;
- ◆ в пределах регистратора (если установлен режим записи Подчинение регистратору).

Периодический регистр сведений содержит поле период, добавляемое системой 1С:Предприятие автоматически. Тип данных поля период — дата, а назначение данного поля связано с соотнесением конкретной записи с определенным периодом. Важно отметить, что когда производится запись в регистр, автоматически осуществляется ее привязка к началу периода, в который она попадает.

Примечание

Свойство Период позволяет указать, с какой периодичностью регистр сведений должен сохранять значения ресурсов. Для непериодического регистра можно получить только последнее введенное значение (предыдущие значения не сохраняются). Для периодического регистра существует единственное значение ресурса внутри периода. Если указан период — *месяц*, то в регистре фиксируется одно значение для каждого месяца. Вариант периодичности *секунда* реально не накладывает ограничений при вводе записей в любой периодический регистр сведений.

Система контролирует уникальность записи в периодический регистр сведений. При этом ключом записи является совокупность значений измерений регистра и периода.

Продемонстрируем работу с регистром сведений на конкретном примере. Наша задача заключается в том, чтобы создать периодический регистр сведений цены (рис. 4.9). Периодичность укажем — В пределах секунды. После этого на закладке Данные создадим измерение Номенклатура (тип данных — СправочникСсылка. Номенклатура). Разумно указать, что данное поле является ведущим (рис. 4.10).

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🔲 🗙							
Файл Правка Конфигурация Отладка Администрирование Сервис							
<u>О</u> кна Сп <u>р</u> авка							
I 🗋 🔟 🖌 🖻 💼 🥌 🖄 🛨 🖉 🛤 👋 🐥 🎽							
🥤 Регистр сведений	й Цены	-	. 🗆 🗙				
• Основные	. Има:	Haut					
Данные	100.A.	цены					
Регистраторы	Синоним:	Цены					
Формы	Комментарий:						
Макеты							
Подсистемы							
Права	Периодичность:	В пределах секунды	▼				
Интерфейсы	Режим записи:	Независимый	.				
▲ ▼							
Действия • <Наз	зад Далее>	Закрыть Сп	равка				

Рис. 4.9. Закладка Основные регистра Цены

Гегистр сведении	Цены	Свойства: Измерение 🕴
∪сновные		2+ <u>с</u> + <u>с</u>
Регистраторы Формы Макеты Подсистемы Права Интерфейсы Обмен данными	В-, ↓ Измерения └ ↓ Номенклатура В ↓ Номенклатура Ресурсы ↓ ↓ Цена ↓ ↓ Цена Реквизиты	Имя Номенклатура Синоним Номенклатура Комментарий Ведущее ✓ Основной отбс✓ Запрет незапс
Прочее		Имя объекта метаданных

Рис. 4.10. Закладка Данные регистра Цены

В этом случае при удалении из справочника определенного элемента автоматически будут удалены все записи из регистра, в которых указан данный элемент

справочника. Также создадим в регистре ресурс цена (тип данных — число). После этого перейдем в режим 1С:Предприятие и внесем несколько записей о ценах (рис. 4.11).

🏪 1С:Предприятие (учебная версия) - Конфигурация 🛛 💶 💌							
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка							
🛭 🛋 🝙 🐰 🖻 🛍 🙄 📲 🛅 🧟 м м+ м- 🎘 🖕							
🥤 Список Цены 💶 🗆 🗙							
Действия - 😫 🛬 🔩 🍇 🔛 🙌 🎘 🐚 🖷 -	· 🕅 🛛 🖉 🕐						
Период 🔹 Номенлатура	Цена 🛆						
🚾 12.08.2008 11:00:20 Комплект 55а	2 500						
🔽 03.09.2008 11:10:35 Комплект 55а	2 670						
🔽 04.09.2008 15:21:21 Комплект 55а	2 600						
	×						
г 👔 Список Цены							
Для получения подсказки нажмите F1	CAP NUM						

Рис. 4.11. Внесение информации о ценах

Использование регистра Цены в документе Поступление

В документе поступление необходимо после выбора товара обеспечить автоматическую подстановку цены на указанную дату. Для этого в окне редактирования формы документа поступление несколько расширим ее функциональность. Наша задача заключается в том, чтобы написать процедуру, выполняемую при вводе товара в табличной части документа. Для этого в окне свойств следует перейти к обработке события приизменении (рис. 4.12).

Осталось написать процедуру (листинг 4.3), в которой используется метод получитьПоследнее (), требующий некоторого комментария. Вообще, для получения информации о значении ресурса регистра сведений можно использовать метод получить (). В этом случае мы можем выбрать конкретное значение ресурса, соответствующее набору измерений (в данной ситуации у нас измерение одно) и значению периода. Например, это можно выполнить с помощью следующего программного кода:

ЗначениеРесурсов=РегистрыСведений.Цены.Получить (Дата, ОтборЦены);

Использование подобной конструкции вполне допустимо, но при условии, что для каждой даты в регистр сведений занесено значение тарифа. Разумеется, в реальной ситуации это не так. Поэтому если на указанную дату информации в регистре сведений нет, то метод получить () вернет пустое значение.

🛃 Конфигуратор (учебная версия) - Конфи	гурация					
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна						
Справка						
🗋 🔟 📓 🔏 🚳 🥌 🖕 с 🦗 м 📃 👻 🖗 👻						
🗮 Документ Поступление: ФормаДокум	іента _ 🗆 🗙					
Поступление						
Действия - 🛃 🔯 😼 👔 ? 🛄	\$‡ [6‡] 1€ × ∨					
Hermon	▼События:					
номер:	ПриИзменені 🚽 🔍					
Дата:	НачалоВыбор 🗸 🗸					
Филиал:	НачалоВыбор 🗸 🗸					
Подразделение	Очистка 🗸 🔍					
	Регулировані 🗸 🔍					
Ответственный:	Открытие 🗸 🗸					
	ОбработкаВь 🗸 🗸					
N Название Цена К	АвтоПодборТ 🚽 🔍 🚽					
аы «Поле ввода» аы «Поле ввода» [ОкончаниеВв 🛛 🗸 🔍					
	Будет доступен данный элемент управления					
	или нет Доступность, Enabled					
📃 🔤 Диалог 📃 Модуль 📑 Реквизиты						

Рис. 4.12. Окно формы документа Поступление

Метод получить Последнее () позволяет извлечь из регистра последнюю по времени цену (не позднее указанной даты) для выбранного товара. Как раз это нам и требуется для рассматриваемой задачи.

Листинг 4.3. Процедура заполнения цены

```
Процедура ПереченьНазваниеПриИзменении(Элемент)

Товар = ЭлементыФормы.Перечень.ТекущиеДанные;

ОтборЦены = Новый Структура("Номенклатура", Товар.Название);

ЗначениеРесурсов = РегистрыСведений.Цены.ПолучитьПоследнее(Дата, ОтборЦены);

Товар.Цена = ЗначениеРесурсов.Цена;

КонецПроцедуры
```

Таким образом, процедура, представленная в листинге 4.3, вносит правильное значение цены на дату документа. После этого в форме документа необходимо ввести количество, что приведет к автоматическому расчету суммы. Для этого требуется соответствующим образом оформить процедуру ПриИзменении для поля количество (листинг 4.4).

Листинг 4.4. Процедура заполнения поля Сумма

```
Процедура ПереченьКоличествоПриИзменении(Элемент)
Товар = ЭлементыФормы.Перечень.ТекущиеДанные;
Товар.Сумма = Товар.Цена * Товар.Количество;
КонецПроцедуры
```

глава 5



Программирование форм

Данная глава преследует цель познакомить читателей с практическими приемами программирования на примере создания конфигурации для учета денежных поступлений. В ней будут использоваться уже встречавшиеся нам объекты. Кроме программирования форм ввода информации (что вынесено в название главы), большое внимание будет уделено и регистрам сведений.

Как и ранее, в процессе выполнения практических действий по тексту главы будет приводиться необходимая справочная информация, дающая представление об используемых программных конструкциях на встроенном языке 1С. Все рассматриваемые в главе процедуры являются частью единой конфигурации, которая заметно сложнее, чем примеры, представленные в предыдущих главах книги.

В заключительном разделе главы приведен пример организации обмена данными между системой 1С:Предприятие 8.1 и приложением Microsoft Word. В частности, рассматривается технология создания документа Microsoft Word из программной процедуры на внутреннем языке 1С. В конце книги данная тема получит развитие, а именно будут рассмотрены примеры обмена данными между информационной базой на платформе 1С:Предприятие 8.1 и приложением Microsoft Excel.

Постановка задачи

Рассматриваемая здесь прикладная задача отражает ситуацию, которая часто встречается при предоставлении различными организациями платных образовательных услуг. В практической работе автоматизация учета всегда используется, и для этого применяются разнообразные программные продукты — Microsoft Excel, Microsoft Access и др. Однако часто после автоматизации большинства участков работы все равно остаются еще и ручные действия — например, заполнение квитанций для оплаты в отделении банка.

В данном случае нам требуется спроектировать конфигурацию, позволяющую эффективно контролировать внесение оплаты за обучение. Созданная разработка на платформе 1С:Предприятие 8.1 будет обеспечивать выписку квитанций на оплату и вести учет сделанных учащимися платежей. На рассматриваемом примере мы по-

пробуем показать, как полностью разгрузить офисного работника от вычислений и ручного заполнения бланков квитанций.

В качестве схемы оплаты взят вариант, используемый в одном из высших учебных заведений. Для определенности будем считать, что наша организация устанавливает помесячную индексацию оплаты за обучение. Поясним, как это реально выглядит на примере. Допустим, что для зачисленного (поступившего) учащегося срок обучения по выбранной им специальности составляет 36 месяцев, но месячная стоимость обучения при этом возрастает с течением времени (точнее, от месяца к месяцу). И если в сентябре 2008 года стоимость за месяц составляла 2000 рублей, то, заплатив именно в сентябре 2008 года по этому тарифу всю сумму (72 000 рублей), учащийся больше ничего учебному заведению не должен. Если же он по своему желанию прекратит обучение, то учебное заведение возвратит ему сумму за не полностью предоставленную услугу (за оставшиеся месяцы после прекращения учебы).

В реальной ситуации учащиеся платят не за весь период, а только за несколько месяцев вперед, и технически это выглядит следующим образом. Учащемуся предварительно выписывается квитанция, которая фиксируется в неком реестре (в информационной базе). Оплатив данную квитанцию в отделении банка и отметив это в соответствующем отделе учебного заведения, он получает следующую квитанцию (на один или несколько следующих месяцев), оплачивает ее и т. д.

Наша задача заключается в создании удобной технологии для ведения описанного учета сотрудниками учебного заведения. Автоматизация должна привести к сокращению времени выполнения технических операций и минимизации количества ошибок в работе. Для более точного восприятия ситуации в табл. 5.1 представлен фрагмент информации о тарифах месячных оплат.

В представленной таблице содержание колонки "Вид тарифа" определяет различные формы обучения — полная учебная программа, ускоренная, сокращенная, индивидуальная и т. д. Каждому учащемуся в зависимости от выбранной им формы обучения устанавливается определенный вид тарифа. Конкретная уникальная комбинация номера месяца, года и вида тарифа по табл. 5.1 позволяет узнать стоимость обучения в определенном месяце в указанном учебном году по выбранной учащимся форме обучения. Эта комбинация и является ключом при записи либо извлечении информации о месячной стоимости обучения.

Номер месяца	Год	Вид тарифа	Стоимость
9	2008	Основной	2000
10	2008	Основной	2020
9	2008	Сокращенная форма	2100
10	2008	Сокращенная форма	2150

Таблица 5.1. Информация о тарифах месячных оплат

Другие важные данные для создания системы учета оплаты представлены в табл. 5.2. Это необходимые сведения об учащихся для ведения учета оплаты. В приведенной таблице каждая строка представляет собой запись о конкретном человеке. При зачислении учащегося с ним заключается договор, информация о котором и отражается в первой колонке таблицы. В последующие поля заносятся данные об учащемся и плательщике (часто оплату вносит один из родственников), а в поле "Число оплачиваемых месяцев" вносится количество месяцев обучения по выбранной форме. Таким образом, это и есть то количество месяцев по договору, которые человек должен оплатить. Оплатить он может сразу при поступлении — тогда индексации не будет, — а при постепенном внесении платежей должно учитываться изменение месячной стоимости с течением времени.

Номер договора	ФИО студента	ФИО плательщика	Число оплачиваемых месяцев	Вид тарифа
100/234	Петров Петр Петрович	Петров Иван Иванович	36	Основной
101/234	Иванов Иван Иванович	Иванов Петр Иванович	36	Основной
102/234	Сидорова Ольга Николаевна	Сидорова Ольга Николаевна	32	Сокращенная форма
103/234	Сидоров Олег Петрович	Сидоров Олег Петрович	32	Сокращенная форма

Таблица 5.2. Информация о договорах

Цель автоматизации описана, и теперь можно перейти к разработке конфигурации, которая позволит организовать на платформе 1С:Предприятие 8.1 удобную систему для ведения учета оплаты. Информация, приведенная в табл. 5.1 и 5.2, будет рассматриваться в качестве исходных данных для создаваемой системы учета.

Для отражения информации каждой из упоминаемых таблиц будет создан соответствующий регистр сведений. После этого, добавив необходимые компоненты интерфейса и запрограммировав алгоритмы обработки, мы получим удобную информационную систему для организации учета оплаты за обучение.

Представление информации о тарифах

Вернемся к табл. 5.1. Приведенные в ней данные с точки зрения математического представления можно трактовать как функцию от трех переменных, в качестве которых выступают год, месяц и вид тарифа. Для каждого уникального набора этих данных в таблице имеется не более одного значения функции — стоимости месяца обучения.

Первой практической задачей в режиме конфигуратора системы 1С:Предприятие, которую мы решим, будет создание регистра сведений под названием тарифы, предназначенного для хранения информации об имеющихся тарифах. Разумеется, эту информацию можно будет вносить, а также извлекать. Если взглянуть на табл. 5.1, то видно, что содержимое колонки "Вид тарифа" представляет собой значения из достаточно ограниченного набора — форм обучения реально немного (как правило, не более 5—6). В этом случае для заполнения колонки "Вид тарифа" удобнее создать вспомогательный справочник с аналогичным названием. Тогда, заполнив его один раз названиями тарифов, мы далее при внесении информации в колонку "Вид тарифа" будем просто выбирать значения из данного справочника.

Итак, теперь можно считать вступительную часть главы завершенной, и со следующего раздела мы перейдем к практическим действиям по созданию новой конфигурации.

Вспомогательный справочник ВидТарифа

Как и в предыдущих главах, начнем работу в режиме конфигуратора с пустой информационной базы. Первым объектом новой конфигурации будет справочник ВидТарифа (рис. 5.1).

🚵 Конфигуратор (учеб	ная версия) - Кон	фигурация						
🛛 🛨 айл Правка Конфигурация Отладка Администрирование Сервис Окна								
Справка								
I 🗋 🖬 X 🖻 🖷 🛎 🖬 🖌 🚆 🎇 🕷 🍣								
菌 Справочник ВидТар	菌 Справочник ВидТарифа 🛛 🗕 🗗 🗙							
• Основные	14	Durtanuta						
Иерархия	имя.	видтарифа						
Владельцы	Синоним:	Вид тарифа						
Данные	Комментарий:							
Нумерация								
Формы								
Действия 🔹 <Назад Далее> Закрыть Справка								
🧮 Справочник ВидТарифа								
Для получения подсказки	нажмите F1	CAP NUM	ru ▼ //					

Рис. 5.1. Окно редактирования справочника ВидТарифа

📙 1С:Предприятие (учебная версия) - Конфиг 💶 🗖 🗙	
📱 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
🗐 Справочник Вид тарифа 🛛 💶 🗙	
Действия - 🔤 🔄 🔩 🍇 🖦 🕅 🎦 🔞 - 🌾 🔶 💙	
Код Наименование 🔹 🛆	Ри
= 1 Основной	Вил
Сокращенная форма	
З ЭСКОРЕННАЯ ФОРМа	
🔄 Справочник Вид тарифа	
Для получения подсказки нажмите F1 САР NUM	



Данный справочник достаточно прост, и нам не потребуется создавать экранные формы для его заполнения — вполне достаточно воспользоваться формой списка, генерируемой системой по умолчанию. На рис. 5.2 показан фрагмент заполнения этого справочника возможными видами тарифов в режиме 1С:Предприятие.

После этого, имея список названий видов тарифов, можно перейти в следующем разделе к созданию регистра сведений тарифы.

Регистр сведений Тарифы

Начальные действия по созданию такого объекта конфигурации нам уже знакомы — щелчок правой кнопкой мыши на названии **Регистры сведений** в окне **Конфигурация** и выбор пункта **Добавить** в контекстном меню. На закладке **Основные** кроме имени регистра необходимо указать, что он является *непериодическим*. Это означает, что *время* в качестве дополнительного аргумента в создаваемом регистре использоваться не будет.

На той же закладке (рис. 5.3) в процессе создания регистра укажем **Независимый** режим записи — в этом случае мы можем напрямую производить запись информации в регистр сведений. В противоположность этому при указании режима Подчинение регистратору появляется возможность формировать записи в регистре автоматически в результате обработки информации из документов. В следующих главах мы рассмотрим и эту возможность, однако при разработке данной конфигурации для всех создаваемых регистров сведений будем использовать независимый режим записи.

🛃 Конфигуратор (уче	📩 Конфигуратор (учебная версия) - Конфигурация 📃 🔲 🗙							
<u>Ф</u> айл <u>П</u> равка Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна								
Справка								
I 🗋 🖬 🔏 👘 🦓 🖓 🖓 📕 🔍 🗸 🗸								
🥤 Регистр сведений Тарифы 🛛 💶 🗙								
• Основные			^					
Данные	Имя:	Тарифы						
Регистраторы	Синоним:	Тарифы						
Формы	Комментарий:							
Макеты								
Подсистемы								
Права	Периодичность:	Непериодический	-					
Интерфейсы	Режим записи:	Независимый	-					
	Основной отбор	по периоду	~					
Действия 👻	<Назад Да	алее> Закрыть Справ	ка					
🌈 Регистр сведений Тај	рифы							
Для получения подсказк	ки нажмите F1	CAP NUM ru	• //					

Рис. 5.3. Окно редактирования регистра сведений Тарифы

Теперь перейдем на закладку Данные, где укажем измерения и ресурсы регистра в соответствии с табл. 5.1. Первое измерение — номермесяца, для которого выберем числовой тип данных. В окне свойств данного реквизита в пункте Индексировать сделаем установку Индексировать. Аналогичные установки выполним и для следующего измерения — год. В соответствии с табл. 5.1 третьим измерением является ВидТарифа, в качестве типа значения которого следует установить СправочникСсылка.ВидТарифа. Кроме того, отметим, что данное измерение является ведущим.

Ресурсом регистра сведений тарифы назначим Стоимость (числовой тип данных). После этого можно сказать, что структура регистра создана. Далее следует перейти в режим 1С:Предприятие и заполнить регистр значениями нескольких тарифов. Вполне достаточно для этого воспользоваться формой ввода по умолчанию. На рис. 5.4 приведен фрагмент заполнения регистра сведений данными о тарифах.

📙 1С:Предприятие (учебная версия) - Конфигурация								
<u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка								
🗋 🛋 🖬 X 🖻 🛍	a 🗟 5 c M	» ▼	🔜 🧟 м м+ м- 🦻	- %				
🥤 Список Тарифы	🥤 Список Тарифы							
Действия 🕶 😫 去 🤅	5 🖬 🕅 Te 🖷 -	Tx 🖸 ?						
Номер месяца	Год	Вид тарифа	Стоимость	^				
8	2 008	Основной	2 00	0				
8	2 008	Сокращенная форма	2 15	0				
8	2 008	Ускоренная форма	2 10	0				
9	2 008	Основной	2 03	0				
9	2 008	Сокращенная форма	2 20	0				
10	2 008	Сокращенная форма	2 22	0				
		-	·	\sim				
🗐 Справочник Вид тарифа 🦷 Список Тарифы								
Для получения подсказки на	жмите F1		CAP NUM	///				

Рис. 5.4. Информация о тарифах за обучение

Контроль ошибок в форме регистра Тарифы

Сделаем еще одну доработку, связанную с автоматической проверкой возможных ошибок при вводе информации. Так, форма, автоматически создаваемая системой, не обеспечивает контроля вводимых данных. А наша цель — установить интервал целых чисел от 1 до 12 при вводе номера месяца, а также задать разумные интервалы при указании года (скажем, от 2008 до 2020). Все это уменьшит технические неточности при работе пользователя. Но для этого вместо формы, генерируемой системой по умолчанию, нам потребуется создать свою форму для ввода информации.

Первый, уже знакомый нам шаг заключается в том, что в окне редактирования объекта конфигурации (рис. 5.5) на закладке **Формы** необходимо щелкнуть мышью по пиктограмме с изображением лупы напротив поля Списка в группе Основные формы.

法 Конфигуратор (учебная версия) - Конфигурация				
<u>Фаил Правка Конфигурация Отладка Администрирование Сервис Окна</u>				
📕 Регистр сведений Гарифы 📃 🗌 🗙				
Основные	— Редактиров	ание	^	
Данные	• В списке			
Регистраторы	О В диалоге			
🕨 Формы	О Обоими способами			
Макеты				
Подсистемы	Ссновные ф	рормы		
Права	Списка	Формасписка	×ч	
Интерфейсы	Записи		x Q	
Обмен данными				
Прочее				
Н-Э Формы ФормаСписка				
Действия - <Назад Далее> Закрыть Справка				
Для получения подсказки нажмите F1 САР NUM ги -				

Рис. 5.5. Закладка Формы окна редактирования регистра сведений

🛃 Конфигуратор (учебная версия) - Конфигурация				
<u>Ф</u> айл Правка Форма Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна				
Справка				
I 🗋 🖬 🔏 🚳 🚳 🖉 🖈 🔺 🔻 🔻 💙				
🚟 Регистр сведений Тарифы: ФормаСписка 🗕 🗌 🗙				
Список Тарифы				
Действия - 🔄 🔄 🔩 🎽 🖬 🎢 🎦 🔞 - 🌾 🔯 🕐 💭				
Номер месяца Год Вид тарифа Стоимость				
аы <Поле ввода>				
🔜 Диалог 🗏 Модуль 🚦 Реквизиты				
🌈 Регистр сведений Тарифы 🛛 🚾 Регистр св: ФормаСписка				
Для получения подсказки нажми САР NUM гит +0,0 ↓ 568 x 262				

Рис. 5.6. Форма списка регистра Тарифы в окне редактирования объекта конфигурации Далее, в следующем открывшемся окне достаточно просто указать, что нас интересует форма списка, после чего щелкнуть по кнопке **Готово**. В открывшемся после этого окне на закладке **Диалог** (рис. 5.6) мы увидим структуру формы с расположенными на ней элементами управления. Теперь можно с помощью мыши при необходимости изменить ширину колонок таблицы, а также размеры табличной части и самой формы.

После двойного щелчка по полю, предназначенному для ввода номера месяца, открывается окно его свойств (рис. 5.7), в котором следует установить упомянутые выше интервалы для вводимой информации. Аналогичные установки нужно выполнить и для поля, предназначенного для указания года.

Этап работы по разработке объекта конфигурации завершен, и теперь следует перейти в режим 1С:Предприятие и посмотреть, как сказываются выполненные изменения. Так, если при заполнении регистра данными о тарифах мы неправильно введем месяц либо год, то система сообщит нам об этом.

Таким образом, в результате проделанной работы мы получили электронный аналог табл. 5.1, но при этом добавили в нее автоматический контроль ошибок при вводе.



Рис. 5.7. Окно редактирования свойств элемента управления

Справочник Договоры

Следующая задача заключается в создании электронного варианта табл. 5.2. Содержащаяся в ней информация достаточно статична, и поэтому для ее хранения лучше всего подходит справочник.

Так что на очереди у нас создание справочника договоры, который будет соответствовать структуре табл. 5.2. При его создании, кроме обязательных полей кода и наименования, добавим еще следующие реквизиты:

- ФИОучащегося (тип данных строка не более 100 символов);
- ♦ ФИОплательщика (тип данных строка не более 100 символов);

- ♦ ДатаДоговора (ТИП Данных Дата);
- ВидТарифа (ТИП Данных СправочникСсылка.ВидТарифа);
- КоличествоОплачиваемыхМесяцев (ТИП данных Число).

Значение параметра количествоОплачиваемыхМесяцев определяет, сколько месяцев обучения студент должен оплатить по заключаемому договору. После создания указанных реквизитов каких-либо еще действий для разработки справочника нам не требуется — при работе в режиме 1С:Предприятие мы воспользуемся формами ввода, генерируемыми системой по умолчанию. Пример заполнения формы элемента справочника договоры приведен на рис. 5.8.

Таким образом, теперь у нас есть объекты конфигурации для хранения информации о стоимости тарифов за обучение и для фиксации сведений о договорах. Следующая задача заключается в создании механизма выписки и регистрации квитанций. Это как раз и составляет основную содержательную часть разработки.

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🔍				
<u>∰ Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка				
I 🗋 🛁 🐰 🖻 🛍 🥌 🙄 I 🎹 🛅 🧝 м. м+ м- 🎾				
🖹 Договоры: 123/34 📃 🗆 🗙				
Действия 🗸 🛃 🐼 Перейти 🗸 ?				
Код:				
Наименование:	123/34			
ФИОучащегося:	Петров Петр Петрович			
ФИОплательщика:	Петрова Ольга Николаевна			
Дата договора:	01.12.2006			
Вид тарифа:	Основной 🗙			
Количество оплачиваемых месяцев:	36			
	ОК Записать Закрыть			
🚞 Договоры: 123/34				
Для получения подсказки нажмите F1 CAP NUM				

Рис. 5.8. Форма элемента справочника Договоры в режиме 1С:Предприятие

Регистр сведений Квитанции

Перейдем к разработке регистра сведений квитанции, в котором будет храниться информация о выписанных квитанциях, а также отмечаться факт их оплаты через отделение банка. Данный регистр мы сделаем периодическим, что и отметим на закладке Основные в поле Периодичность.

Для решаемой нами задачи разумно указать периодичность в течение секунды, что позволит хранить практически любое количество квитанций по конкретному договору. Также на этой же закладке установим **Независимый** режим записи в рассматриваемый регистр.
Теперь перейдем на закладку Данные, где от нас требуется задать структуру полей регистра (рис. 5.9). Измерений в регистре квитанции будет два — договор и период (как уже говорилось, второе измерение автоматически добавляется для всех периодических регистров).

Для поля договор в качестве типа данных укажем СправочникСсылка. Договоры, а также отметим, что данное поле является ведущим. В этом случае при удалении объекта информационной системы автоматически производится удаление записей регистра сведений, в которых имеется ссылка на удаляемый объект. Например, записи квитанций по конкретному договору будут автоматически удалены системой при удалении договора. Для ресурса СуммаПоКвитанции следует указать числовой тип данных. Каждая запись в регистре сведений информирует о конкретной квитанции, и для отражения денежной суммы выписанной квитанции отводится данный ресурс.



Рис. 5.9. Структура регистра сведений Квитанции

Реквизиты регистра несут уточняющую информацию по каждой записи в нем. Так, для поля оплачено следует установить булев тип данных, после чего в указанном поле можно отмечать, произведена ли оплата по данной квитанции. Для других реквизитов (МесяцНачала, ГодНачала, МесяцОкончания, ГодОкончания) установим числовой целочисленный тип данных.

Теперь можно подвести небольшой промежуточный итог. Мы создали необходимые регистры (тарифы и квитанции), а также справочники — видтарифа и договоры. Теперь для работы с квитанциями следует разработать специальную электронную форму, которая за счет программных процедур обеспечит сервисные возможности пользователю.

Примечание

Свойства измерений, ресурсов и реквизитов редактируются при помощи уже знакомого нам окна свойств. В целом свойства регистров сведений аналогичны свойствам других объектов конфигурации. Однако есть и ряд свойств, характерных именно для регистров. Так, установка свойства **Ведущее** используется, если у рассматриваемого измерения тип данных — ссылка на объект конфигурации. В этом случае запись регистра сведений имеет смысл, только пока существует сам объект конфигурации. При удалении объекта записи по нему удаляются системой автоматически. Среди свойств измерений присутствует флажок **Запрет незаполненных значений**, который запрещает запись регистра с пустым значением измерения. Установка свойства **Индексировать** приводит к созданию отдельного индекса, что в ряде случаев увеличивает производительность работы с регистром.

Форма обработки для учета квитанций по оплате

В этом разделе мы самостоятельно разработаем электронную форму для выписки новых квитанций на оплату. При этом первым шагом на этом пути будет создание новой обработки (рис. 5.10). Фактически в создаваемой обработке нам потребуется только объект "форма", который является подчиненным обработке.

🛃 Конфигуратор (учебная версия)	- Конфигурация	
<u>Ф</u> айл <u>П</u> равка Ко	нфигурация Отлад	ка Администрирование <u>С</u>	ервис
<u>О</u> кна Сп <u>р</u> авка			
I 🗋 🚢 🖬 🕺	r r 6	5 C 🕅 M	» •
💣 Обработка Уч	етКвитанций		<u>– 🗆 ×</u>
• Основные	14		
Данные	имя:	9четквитанций	
Формы	Синоним:	Учет квитанций	
Макеты	Комментарий:		
Подсистемы			
Права			
Интерфейсы			
Прочее			
Действия 🗸 <	Назад Дале	е» Закрыть (Справка
å [●] Обработка УчетКвитанций			
Для получения подск	казки нажмите F1	CAP NUM	ru▼ //

Рис. 5.10. Окно редактирования обработки УчетКвитанций

Примечание

Для выполнения различных действий с данными в системе 1С:Предприятие используются объекты конфигурации под названием "обработки". Обработка может содержать одну или несколько форм, с помощью которых организуется ввод необходимых параметров, а также вывод информации в желаемом виде. С точки зрения функционального назначения работы объекты "отчет" и "обработка" довольно похожи.

Итак, в окне редактирования объекта конфигурации (см. рис. 5.10) необходимо перейти на закладку **Формы** и щелкнуть мышью по изображению лупы в разделе **Основная форма обработки**. После этого в окне конструктора формы обработки (рис. 5.11) необходимо сделать уже знакомые установки (в частности, для типа формы выберем вариант **Произвольная форма**).

Конструктор фо	онструктор формы обработки 🛛 🗙 🗙		
Выберите тип формы: Форма обработки Произвольная форма Назначить форму основной			
Имя:	ФормаУчетаКвитанций		
Синоним:	Форма учета квитанций		
Комментарий:			
🗌 Командная	панель Формы сверху		
Командная панель формы снизу			
Казад Далее > Готово Отмена Справка			

Рис. 5.11. Окно конструктора формы обработки

В результате описанных действий с конструктором мы получим пустую экранную форму, на которой далее разместим необходимые элементы управления. Для этого следует воспользоваться пунктом Вставить элемент управления меню Форма.

Первый фрагмент необходимых элементов представлен на рис. 5.12, который далее прокомментируем. Для достижения такого результата прежде всего необходимо на форме расположить элемент **Рамка группы**. В качестве подписи группы укажем **Информация о договоре**. Этот элемент носит чисто оформительский характер и каких-то функциональных действий не выполняет.

С помощью элемента **Рамка группы** производится визуальное объединение нескольких элементов управления, относящихся к определенной тематической категории. Внешне **Рамка группы** на форме выглядит как контейнер, окружающий другие (более функциональные) элементы управления. Теперь в созданной рамке расположим поле ввода и укажем тип данных СправочникСсылка.Договоры, что позволит при работе в режиме 1С:Предприятие выбирать необходимый договор из предварительно сформированного списка договоров (в справочнике договоры). Дадим этому элементу имя договор.

Примечание

Элемент управления *поле ввода* предназначен для ввода и редактирования информации. Поле ввода используется как для ввода значений непосредственно с клавиатуры, так и для выбора значений из списков.



Ниже списка договоров размещаются семь элементов типа *надпись*, при этом четыре надписи являются просто поясняющими и далее в программных процедурах не используются, а три (на рис. 5.12 они имеют более светлый оттенок) при работе с формой будут автоматически (в результате выполнения программы) заполняться при выборе пользователем конкретного договора. Имена этих трех надписей выделены угловыми скобками, и их мы будем использовать в программных процедурах. Заметим, что для того чтобы эти три надписи выглядели в соответствии с рис. 5.12, в окне свойств этих объектов не следует заполнять свойство Заголовок.

Примечание

Надпись предназначена для размещения на форме подписей к другим элементам управления. Однако надпись можно использовать и для отображения текстовой информации, которая не предназначена для коррекции и изменения (этим элемент управления надпись отличается от элемента поле ввода).

Следующая порция элементов управления показана на рис. 5.13. Их основное назначение связано с извлечением информации о величине тарифа из регистра све-

дений тарифы. Кроме надписей и полей ввода нам понадобится кнопка, для имени которой выберем сочетание Расчеттарифа (надпись на кнопке автоматически установится аналогичной, но только с разделяющим пробелом между словами).

🛃 Конфигуратор (учебная версия) - Конфигурация	_ _ ×
🛛 🕂 дайл Правка Форма Конфигурация Отладка Администрир	рвание <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
I 🗋 🔟 🖌 🗞 🛍 🥌 🔍 🖕 🖒 🕐 🕅 🕷 👘	👻 🐼 🖓 🛅 🧏 🐅 🗳
🚟 Обработка УчетКвитанций: ФормаУчетаКвитанций	×
Форма	<u> </u>
Информация о договоре Договор: × ФИО <<Учащийся>> Дата заключения договора <<<на:	амация о тарифе произведения оплаты: 0 ризведения оплаты: 0 звание Тарифа>>
Месяц <<МесяцДоговора>> Год <<ГодДоговора>> Расч	тарифа <<ТарифЗаМесяц <<ТарифЗаМесяц>>
🗶 📰 Диалог 🗏 Модуль 📑 Реквизиты	
😰 Обработка УчетКвитанций 🗮 Об: ФормаУчетаКвитанций Для получения подсказки нажмите F1 САР	NUM ru → +=0, 0 +==597 x 376

Рис. 5.13. Элементы управления для отображения информации о договоре и тарифе

Примечание

Кнопка — это элемент управления, который предназначен для выполнения различных действий. Нажатие кнопки связано с выполнением определенной процедуры.

В рамке **Информация о тарифе** располагаются два поля ввода — для указания месяца и года, когда учащийся собирается произвести оплату квитанции. Имена этих элементов выберем соответственно — месяцПроизведенияОплаты и ГодПроизведенияОплаты. Мы использовали здесь поля ввода, а не надписи, поскольку по смыслу задачи пользователю необходимо предоставить возможность ручной установки даты оплаты (с точностью до месяца), если учащийся собирается оплатить квитанцию в последующие месяцы.

Надпись название Тарифа автоматически заполняется при выборе договора в рамке **Информация о договоре**, а надпись ТарифЗаМесяц также автоматически заполняется величиной тарифа для указанного месяца (конечно, если соответствующая запись имеется в регистре Тарифы). Эти действия будут выполняться в программных процедурах, которые приводятся далее.

Теперь нам осталось разместить элементы в рамках **Информация о последней** оплате и Расчет следующей оплаты (рис. 5.14), причем элементы управления в левой рамке должны автоматически заполняться данными из регистра квитанции при выборе договора (выборе договора с учащимся в поле ввода договор). Это также будет реализовано с помощью программной процедуры. В рамке Расчет следующей оплаты располагаются элементы, которые позволяют пользователю рассчитать сумму для следующей квитанции и занести необходимую информацию в регистр квитанции. Выбор начала оплаты производится с помощью двух полей ввода — месяцначалаОплаты и годначалаОплаты. В эти поля после нахождения последней оплаченной квитанции (что происходит программно при выборе пользователем договора учащегося) будет заноситься следующий месяц (номер месяца и год), начиная с которого клиент должен продолжить оплату. Учитывая, что данные поля являются полями ввода, пользователь в режиме 1С:Предприятие может при необходимости эту автоматически сформированную информацию изменить.

В поле ввода количествооплачиваемых Месяцев пользователь при работе в режиме 1С:Предприятие должен ввести число месяцев, за которые учащийся собирается заплатить, а кнопка **Расчет суммы** позволяет рассчитать сумму оплаты и вывести ее в соседнем с ней справа поле (с именем суммаоплаты). Кнопка **Записать квитан**цию реализует запись информации о квитанции в предназначенный для этого регистр сведений квитанции. О назначении еще одной кнопки (Передача в Word) мы поговорим в конце данной главы.

📩 Конфигуратор (учебная версия) - Конфигураци	яX
🛛 🕂 айл Правка Форма Конфигурация Отладка Адм	иинистрирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
I 🗋 🖬 🐰 🏷 🛍 📾 📾 🐿 🖕 🖬 🕷	🔹 🐨 🕅 👘 🎘 🐅 🖉 唑 🖕
🧮 Обработка УчетКвитанций: ФормаУчетаКви	ганций _ 🗆 🗙
Форма	
Информация о договоре	-Информация о тарифе
Договор: х	Месяц произведения оплаты: 0
ФИО <<Учащийся>>	Год произведения оплаты: 0
Дата заключения договора	<<НазваниеТарифа>>
Месяц <<МесяцДоговора>>	Тариф за указанный месяц
Год <<ГодДоговора>>	Расчет тарифа <<ТарифЗаМесяц>>
Информация о последней оплате	-Расчет следующей оплаты
Начало предыдущего интервала	Месяц начала оплаты: 0
<<<МесяцНачалаИнтервала>>	Год начала оплаты:
< <<i одначалайнтервала="">> Окончание предыдушего интервала</i>	Количество оплачиваемых месяцев: 0
<<МесяцОкончанияИнтервал	Расчет суммы 0
<<ГодОкончанияИнтервала>>	Записать квитанцию Передача в Word
Диалог Модуль 📑 Реквизиты	
🛃 Обработка УчетКвитанций 🛛 🚟 Об: ФормаУчетаК	витанций
Для получения подсказки нажмите F1	CAP NUM ru▼ +0,0 ∓2597 x 376

Рис. 5.14. Форма обработки УчетКвитанций в режиме конфигуратора

Таким образом, мы выполнили предварительную конструкторскую разработку формы для внесения информации о квитанциях, а также рассмотрели основные функциональные действия, связанные с элементами формы. В следующих разделах главы будут рассмотрены программные процедуры работы с самой формой и с элементами управления, расположенными на ней.

Процедуры для работы с формой

В этом разделе мы воспользуемся программным доступом к информации, содержащейся в регистрах сведений, а именно будем в программе осуществлять запись в эти регистры, а также извлекать из них информацию.

К этому моменту мы создали форму с необходимыми элементами управления, и теперь можно приступить к разработке программных процедур. Прежде всего, необходимо сделать так, чтобы при открытии формы автоматически заполнялись поля МесяцПроизведенияОплаты и ГодПроизведенияОплаты, исходя из информации о сегодняшней дате. Это параметры даты оплаты по умолчанию, которые пользователь может при необходимости изменить.

Первые технические действия для перехода к программированию начинаются с двойного щелчка мышью в области формы, что позволяет открыть окно ее свойств (рис. 5.15). Далее следует в разделе События напротив пункта ПриОт-крытии щелкнуть по пиктограмме с изображением лупы. В результате перед нами откроется знакомая закладка Модуль с уже созданной заготовкой для процедуры ПриОткрытии (). От нас здесь потребуется ввести небольшой фрагмент программного кода, показанный в листинге 5.1, который мы прокомментируем далее.



Рис. 5.15. Окно свойств формы обработки УчетКвитанций

Листинг 5.1. Процедура, выполняемая при открытии формы

```
Процедура ПриОткрытии()
```

```
ЭлементыФормы.МесяцПроизведенияОплаты.Значение = Месяц(ТекущаяДата());
ЭлементыФормы.ГодПроизведенияОплаты.Значение = Год(ТекущаяДата());
КонецПроцедуры
```

Здесь используется функция месяц (<дата>), в качестве обязательного входного параметра которой должно присутствовать значение типа дата. Значение, возвращаемое этой функцией, представляет собой числовое значение месяца в указанной дате. Похожая по назначению функция Год (<дата>) возвращает числовое значение года в исходной дате, а для получения непосредственно самой даты мы воспользовались еще одной функцией системы 1С:Предприятие — текущаядата(). Если теперь перейти в режим 1С:Предприятие и открыть форму обработки учетквитанций, то рассматриваемые два поля на форме будут автоматически заполнены, исходя из сегодняшней даты.

Примечание

Предопределенная процедура ПриОткрытии () выполняется при открытии формы, до показа ее пользователю.

Процедура, выполняемая при выборе договора

Перейдем теперь к разработке следующей процедуры, выполняемой, когда пользователь при работе в режиме 1С:Предприятие укажет в поле ввода Договор номер интересующего его договора с клиентом. Такой выбор автоматически должен привести к заполнению нескольких полей на форме, и далее описаны необходимые программные действия для реализации подобного эффекта.

Итак, в режиме конфигуратора откроем окно свойств элемента договор и перейдем к разделу События, где от нас требуется щелкнуть по изображению лупы напротив строки ПриИзменении. В результате перед нами откроется окно редактора программного кода, а курсор ввода будет установлен на заготовке для соответствующей программной процедуры. Мы будем разрабатывать эту процедуру в несколько этапов, последовательно добавляя в нее различные усовершенствования.

Первоначальный вариант текста процедуры, обеспечивающий минимальные действия по заполнению полей формы, приведен в листинге 5.2.

Листинг 5.2. Процедура, выполняемая при выборе договора (вариант 1)

Процедура ДоговорПриИзменении (Элемент)

```
Договор = ЭлементыФормы.Договор.Значение;
```

```
ЭлементыФормы.Учащийся.Значение = Договор.ФИОучащегося;
```

```
ЭлементыФормы.МесяцДоговора.Значение = Месяц (Договор.ДатаДоговора);
```

```
ЭлементыФормы.ГодДоговора.Значение = Год (Договор.ДатаДоговора);
```

```
ЭлементыФормы.НазваниеТарифа.Значение = Договор.ВидТарифа;
```

```
КонецПроцедуры
```

Прокомментируем текст листинга 5.2. В начале процедуры мы вводим переменную договор для того, чтобы в дальнейшем использовать ее в программных конструкциях вместо длинного сочетания ЭлементыФормы. Договор. Значение. Далее в созданные ранее надписи на форме программно подставляются значения из справочника договоры. Таким образом, заносится информация о месяце и годе заключения договора, а также о названии тарифа. Кроме того, с помощью данной процедуры фамилия учащегося заносится в надпись учащийся. Для проверки можно перейти в режим 1С:Предприятие и убедиться в работоспособности программных конструкций.

Теперь выполним небольшое усовершенствование. Ранее на форме мы расположили поле тарифЗаМесяц, которое должно быть заполнено, исходя из даты произведения оплаты и названия тарифа, по которому оплачивает обучение конкретный учащийся. Данная информация уже присутствует на форме — внесены значения в поля МесяцПроизведенияОплаты и ГодПроизведенияОплаты, а также заполнена надпись название Тарифа. Это было выполнено с помощью процедур, представленных в листингах 5.1 и 5.2. Модифицированная процедура договорПриИзменении (), обеспечивающая заполнение поля тарифЗаМесяц, приведена в листинге 5.3.

Листинг 5.3. Процедура, выполняемая при выборе договора (вариант 2)

```
Процедура ДоговорПриИзменении (Элемент)
Договор = ЭлементыФормы.Договор.Значение;
ЭлементыФормы.Учащийся.Значение = Договор.ФИОучащегося;
ЭлементыФормы.МесяцДоговора.Значение = Месяц (Договор.ДатаДоговора);
ЭлементыФормы.ГодДоговора.Значение = Год (Договор.ДатаДоговора);
ЭлементыФормы. Название Тарифа. Значение = Договор. Вид Тарифа;
Запрос = Новый Запрос;
Запрос. Текст = "ВЫБРАТЬ *
      РегистрСведений. Тарифы
I N3
| ГДЕ ВидТарифа.Наименование=&НазваниеТарифа И
| НомерМесяца = «Месяц И
| Год = &Год";
Запрос.УстановитьПараметр ("НазваниеТарифа",
        ЭлементыФормы. НазваниеТарифа. Значение);
Запрос.УстановитьПараметр ("Месяц",
        ЭлементыФормы. МесяцПроизведенияОплаты. Значение);
Запрос.УстановитьПараметр ("Год",
        ЭлементыФормы.ГодПроизведенияОплаты.Значение);
Результат = Запрос.Выполнить ().Выбрать ();
Если Результат.Следующий () > 0 Тогда
    ЭлементыФормы. ТарифЗаМесяц. Значение = Результат. Стоимость;
Иначе
    ЭлементыФормы. ТарифЗаМесяц. Значение = 0;
    Сообщить ("По указанным данным информации о тарифе нет");
КонецЕсли;
КонецПроцедуры
```

Примечание

Здесь и в ряде последующих процедур используются запросы. Детально они будут рассмотрены в следующей главе. Здесь используются несложные конструкции запросов, которые поясняются по тексту.

В рассматриваемой процедуре производится запрос к регистру сведений тарифы. Параметрами запроса являются вид тарифа и текущая дата (месяц и год). Для установки значений параметров используется уже знакомый нам метод Установить Параметр (). В результате выполнения запроса возможны две ситуации:

• запрос возвращает единственную запись об интересующем нас тарифе;

• выполнение запроса показало, что записи о тарифе на указанную дату нет.

В первом случае мы выводим информацию о тарифе в соответствующую надпись на форме, а во втором — в окно сообщений выводится соответствующее сообщение.

Теперь еще немного усложним процедуру. Набор надписей, расположенных внутри рамки **Информация о последней оплаченной квитанции**, необходимо автоматически заполнить информацией о самой свежей квитанции, по которой учащийся внес оплату (листинг 5.4).

Листинг 5.4. Процедура, выполняемая при выборе договора (вариант 3)

```
Процедура ДоговорПриИзменении (Элемент)
Договор = ЭлементыФормы.Договор.Значение;
ЭлементыФормы.Учащийся.Значение = Договор.ФИОучащегося;
ЭлементыФормы.МесяцДоговора.Значение = Месяц (Договор.ДатаДоговора);
ЭлементыФормы.ГодДоговора.Значение = Год (Договор.ДатаДоговора);
ЭлементыФормы.НазваниеТарифа.Значение = Договор.ВидТарифа;
Запрос = Новый Запрос;
Запрос. Текст = "ВЫБРАТЬ *
     РегистрСведений.Тарифы
EN |
| ГДЕ ВидТарифа.Наименование=&НазваниеТарифа И
| НомерМесяца =&Месяц И
| Год = &Год";
Запрос.УстановитьПараметр ("НазваниеТарифа",
        ЭлементыФормы. НазваниеТарифа. Значение);
Запрос.УстановитьПараметр ("Месяц",
        ЭлементыФормы. МесяцПроизведенияОплаты. Значение);
Запрос.УстановитьПараметр ("Год",
        ЭлементыФормы.ГодПроизведенияОплаты.Значение);
Результат = Запрос.Выполнить ().Выбрать ();
Если Результат.Следующий() > 0 Тогда
    ЭлементыФормы. ТарифЗаМесяц. Значение = Результат. Стоимость;
Иначе
    ЭлементыФормы. ТарифЗаМесяц. Значение = 0;
    Сообщить ("По указанным данным информации о тарифе нет");
КонецЕсли;
```

```
ЭлементыФормы. МесяцНачалаИнтервала. Значение = "";
ЭлементыФормы. ГодНачалаИнтервала. Значение = "";
ЭлементыФормы.МесяцОкончанияИнтервала.Значение = "";
ЭлементыФормы.ГодОкончанияИнтервала.Значение = "";
Запрос2 = Новый Запрос;
Запрос2.Текст = "ВЫБРАТЬ *
I N3
     РегистрСведений.Квитанции
| ГДЕ Договор=&Договор И
| Оплачено = «Индикатор";
Запрос2.УстановитьПараметр ("Договор", ЭлементыФормы.Договор.Значение);
Запрос2.УстановитьПараметр ("Индикатор", Ложь);
Результат = Запрос2.Выполнить ().Выбрать ();
Если Результат.Следующий() > 0 Тогда
    Сообщить ("По данному договору есть неоплаченная квитанция");
    Возврат;
КонецЕсли;
СтруктураОтбора = Новый Структура;
СтруктураОтбора.Вставить ("Договор", Договор);
ВыборкаЗаписей =
    РегистрыСведений.Квитанции.Выбрать (,, СтруктураОтбора);
Пока ВыборкаЗаписей.Следующий() Цикл
    Если ЭлементыФормы. Месяц Начала Интервала. Значение = "" Тогда
        ЭлементыФормы. МесяцНачалаИнтервала. Значение =
            ВыборкаЗаписей.МесяцНачала;
        ЭлементыФормы. ГодНачалаИнтервала. Значение =
            ВыборкаЗаписей.ГодНачала;
        ЭлементыФормы. МесяцОкончанияИнтервала. Значение =
            ВыборкаЗаписей.МесяцОкончания;
        ЭлементыФормы. ГодОкончанияИнтервала. Значение =
            ВыборкаЗаписей.ГодОкончания;
    Иначе
        Дата1 = Число (ЭлементыФормы.МесяцОкончанияИнтервала.Значение) +
             12*Число (ЭлементыФормы. ГодОкончанияИнтервала. Значение);
        Дата2 = ВыборкаЗаписей.МесяцОкончания+
             12*ВыборкаЗаписей.ГодОкончания;
        Если Дата2 > Дата1 Тогда
            ЭлементыФормы. МесяцНачалаИнтервала. Значение =
                      ВыборкаЗаписей.МесяцНачала;
            ЭлементыФормы.ГодНачалаИнтервала.Значение =
                      ВыборкаЗаписей.ГодНачала;
            ЭлементыФормы. МесяцОкончания Интервала. Значение =
                      ВыборкаЗаписей.МесяцОкончания;
            ЭлементыФормы.ГодОкончанияИнтервала.Значение =
                      ВыборкаЗаписей.ГодОкончания;
        КонецЕсли;
    КонецЕсли;
КонецЦикла;
КонецПроцедуры
```

В новом добавленном фрагменте процедуры извлекается информация из периодического регистра сведений квитанции. Принципиальным отличием такого регистра от непериодического является присутствие поля период. Для работы с регистром в процедуре мы создаем структуру отбора, которую заполняем информацией о договоре:

СтруктураОтбора.Вставить ("Договор", Договор);

После этого можно использовать метод Выбрать () для периодического регистра сведений в следующей редакции:

РегистрыСведений.Квитанции.Выбрать (,, СтруктураОтбора);

Вызов метода выбрать () требует в общем случае указания нескольких параметров. Третий параметр представляет ссылку на структуру отбора, а два предыдущих определяют интервал по дате — начало интересующего нас интервала и его окончание. Если первые два параметра оставлены пустыми (в нашем вызове метода именно такая ситуация), то в результате мы получим выборку всех квитанций по конкретному договору.

После перебора всех отобранных записей заполняются надписи — МесяцНачала-Интервала, ГодНачалаИнтервала, МесяцОкончанияИнтервала, ГодОкончанияИнтервала. В них фиксируется информация о последней оплаченной квитанции — за какой интервал месяцев она выписывалась. Таким образом, листинг 5.4 представляет собой почти окончательный вариант процедуры договорПриИзменении, за исключением только одного действия — расчета месяца и года, с которого учащийся должен начать следующую оплату. Эта завершающая часть, но без уже рассмотренных выше строк, приведена в листинге 5.5.

Листинг 5.5. Дополнение к процедуре, приведенной в листинге 5.4

```
Процедура ДоговорПриИзменении (Элемент)
// Здесь располагается текст из листинга 5.4
Если ЭлементыФормы.МесяцОкончанияИнтервала.Значение <> "" Тогда
     ЭлементыФормы.МесяцНачалаОплаты.Значение =
         Число (ЭлементыФормы. МесяцОкончанияИнтервала. Значение) + 1;
     Если Число (ЭлементыФормы.МесяцНачалаОплаты.Значение) > 12 Тогда
         ЭлементыФормы.МесяцНачалаОплаты.Значение =
             Число (ЭлементыФормы. МесяцНачалаОплаты. Значение) -12;
         ЭлементыФормы.ГодНачалаОплаты.Значение =
             Число (ЭлементыФормы. ГодОкончанияИнтервала. Значение) +1;
     Иначе
         ЭлементыФормы.ГодНачалаОплаты.Значение =
             ЭлементыФормы.ГодОкончанияИнтервала.Значение;
    КонецЕсли;
КонецЕсли;
КонецПроцедуры
```

Таким образом, мы разработали окончательный вариант процедуры, выполняемой при заполнении поля договор из справочника договоры. Подведем итог — перечислим, какие функциональные действия она обеспечивает:

- вывод информации о фамилии, имени и отчестве учащегося;
- отображение месяца и года заключения договора;
- представление данных о последней оплате;
- установка на форме названия вида тарифа, по которому учащийся оплачивает обучение;
- заполнение полей для указания месяца и года следующей оплаты.

На рис. 5.16 показана разработанная форма, открытая в режиме 1С:Предприятие. Информация в элементах формы заполнена автоматически после выбора названия договора. Разработанные нами процедуры пока являются только подготовительными. Они реализуют извлечение информации из регистра Квитанции. В следующих разделах описываются активные действия со стороны пользователя.

🏪 1С:Предприя	тие (учебная версия) - Конфиг	урация	
<u>∎ Ф</u> айл <u>П</u> равка	Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> ави	a	
I 🗋 📫 🖬 🕽	(🖻 🛍 🥌 🖻 5	🐥 📱 🎆 🧮 🧝 М М+ М	r- 🎘 🖵
🖹 Форма		-	. 🗆 🗙
- Информа	щия о договоре	Информация о тарифе	
Договор:	123/34 x	Месяц произведения оплаты: 9	
ФИО	Петров Петр Петрович	Год произведения оплаты: 2 008	
Дата зак	лючения договора	Основной	
Месяц	12	Тариф за указанный месяц	
Год	2 006	Расчет тарифа 2 030	,
-Информа	ация о последней оплате	Расчет следующей оплаты	- I
Начало пре	едыдущего интервала	Месяц начала оплаты: 11	
	8	Год начала оплаты: 2 007	
Оконцания		Количество оплачиваемых месяцев: 0	
Окончание	10	Расчет суммы 0	
	2 007	2amucatte Keistraatiuwo	
		Записать коитанцию	
рорма 📄			
Для получения по	дсказки нажмите F1	CAP 1	NUM //

Рис. 5.16. Форма для работы с квитанциями в режиме 1С:Предприятие

Обработка нажатия кнопки Расчет тарифа

Предыдущими действиями мы обеспечили автоматическое заполнение поля тарифзамесяц (на рис. 5.16 оно располагается ниже надписи Тариф за указанный месяц) после указания названия договора в поле договор. Для этого поля МесяцПроизведенияОплаты и ГодПроизведенияОплаты должны быть предварительно заполнены, и их заполнение производится при открытии формы (процедура ПриОткрытии ()).

Таким образом, при открытии формы автоматически заполняются поля о дате предполагаемой оплаты, а после выбора договора автоматически вносится сумма тарифа на эту дату. Однако часто учащийся при выписке квитанции уже заранее собирается оплачивать ее не в текущем месяце, а позже. В этом случае и тариф за месяц будет другой — более высокий.

В связи с этим необходимо предусмотреть возможность внесения изменений в поля МесяцПроизведенияОплаты и ГодПроизведенияОплаты, а затем и повторного заполнения надписи тарифЗаМесяц уже правильной информацией.

Описанные действия выполняются при нажатии кнопки **Расчет тарифа**. Перейдем к процедуре (листинг 5.6), с помощью которой все это и реализуется. Фактически ее содержание нам уже знакомо по фрагменту листинга 5.3, где приведена процедура, выполняемая при выборе договора.

Листинг 5.6. Процедура, выполняемая при нажатии кнопки Расчет тарифа Процедура РасчетТарифаНажатие (Элемент) Запрос = Новый Запрос; Запрос. Текст = "ВЫБРАТЬ * | ИЗ РегистрСведений. Тарифы | ГДЕ ВидТарифа.Наименование=&НазваниеТарифа И | НомерМесяца = «Месяц И | Год = &Год"; Запрос.УстановитьПараметр ("НазваниеТарифа", ЭлементыФормы. НазваниеТарифа. Значение); Запрос.УстановитьПараметр ("Месяц", ЭлементыФормы. МесяцПроизведенияОплаты. Значение); Запрос.УстановитьПараметр ("Год", ЭлементыФормы.ГодПроизведенияОплаты.Значение); Результат = Запрос.Выполнить().Выбрать(); Если Результат.Следующий() > 0 Тогда ЭлементыФормы.ТарифЗаМесяц.Значение = Результат.Стоимость; Иначе ЭлементыФормы. ТарифЗаМесяц. Значение = 0; Сообщить ("По указанным данным информации о тарифе нет"); КонецЕсли; КонецПроцедуры

Теперь пользователь, изменив месяц (а при необходимости и год), когда клиент собирается произвести оплату, может всегда вывести на форму величину необходимого тарифа из регистра квитанции.

Процедура расчета суммы квитанции

Следующая процедура (выполняемая при нажатии кнопки **Расчет суммы**) производит простое перемножение чисел, расположенных в элементах тарифЗаМесяц и количествоОплачиваемыхМесяцев. Результат записывается в поле суммаОплаты, при этом в случае необходимости пользователь может его скорректировать (просто внести другую сумму, если это нужно).

```
Листинг 5.7. Процедура, выполняемая при нажатии кнопки Расчет суммы
```

Процедура РасчетСуммыНажатие (Элемент) Количество = Число (ЭлементыФормы.КоличествоОплачиваемыхМесяцев.Значение); ЦенаМесяц = Число (ЭлементыФормы.ТарифЗаМесяц.Значение); Сумма = Количество*ЦенаМесяц; ЭлементыФормы.СуммаОплаты.Значение = Сумма; КонецПроцедуры

Запись информации в регистр Квитанции

Последняя процедура в плане обеспечения функциональности работы с формой связана с записью параметров новой введенной квитанции на оплату в регистр квитанции. Сам текст процедуры приведен в листинге 5.8. В нем программным путем создается новая запись в регистре сведений, а далее в эту запись вносятся значения четырех параметров — договор, Период, МесяцНачала, ГодНачала. Значения трех параметров просто переносятся с соответствующих элементов формы, а для заполнения значения измерения период мы воспользовались уже известной нам функцией текущаяДата ().

Что касается значений месяцОкончания и годОкончания, то содержание этих полей рассчитывается из информации о месяце начала оплаты и количестве оплачиваемых месяцев.

```
Листинг 5.8. Процедура, выполняемая при нажатии кнопки Записать квитанцию
```

```
Процедура ЗаписьКвитанцииНажатие (Элемент)
КоличествоМесяцев =
Число (ЭлементыФормы.КоличествоОплачиваемыхМесяцев.Значение);
ВариантМесяца = КоличествоМесяцев +
Число (ЭлементыФормы.МесяцНачалаОплаты.Значение) — 1;
ВариантГода = Число (ЭлементыФормы.ГодНачалаОплаты.Значение);
Если ВариантМесяца > 12 Тогда
ВариантМесяца = ВариантМесяца — 12;
ВариантГода = Число (ЭлементыФормы.ГодНачалаОплаты.Значение) + 1;
КонецЕсли;
НаборЗаписей = РегистрыСведений.Квитанции.СоздатьНаборЗаписей();
НаборЗаписей.Отбор.Период.Установить (ТекущаяДата());
НоваяЗапись = НаборЗаписей.Добавить();
НоваяЗапись.Договор = Договор;
```

```
НоваяЗапись.Период = ТекущаяДата();
НоваяЗапись.МесяцНачала = ЭлементыФормы.МесяцНачалаОплаты.Значение;
НоваяЗапись.ГодНачала = ЭлементыФормы.ГодНачалаОплаты.Значение;
НоваяЗапись.МесяцОкончания = ВариантМесяца;
НоваяЗапись.ГодОкончания = ВариантГода;
НоваяЗапись.СуммаПоКвитанции = ЭлементыФормы.СуммаОплаты.Значение;
НоваяЗапись.Оплачено = Ложь;
НаборЗаписей.Записать (Истина);
КонецПроцедуры
```

В процедуре, приведенной в листинге 5.8, применяются новые программные конструкции. Так, для работы с регистром сведений используется ряд объектов. Мы воспользовались объектом РегистрСведенийНаборЗаписей.Квитанции, который позволяет выполнять операции над несколькими записями, обеспечивая их считывание и запись в базу данных.

Примечание

РегистрСведенийНаборЗаписей. Квитанции представляет собой коллекцию записей регистра сведений Квитанции. Этот объект позволяет считывать записи по определенному условию из базы данных. Также с его помощью можно добавлять, удалять и модифицировать записи в наборе. Возможно и замещение всех имеющихся в базе данных записей по указанному условию на записи, содержащиеся в наборе.

В процедуре, представленной в листинге 5.8, для создания этого объекта используется метод СоздатьНаборЗаписей (). После выполнения строки

НаборЗаписей = РегистрыСведений.Квитанции.СоздатьНаборЗаписей()

мы получаем возможность в дальнейшем с помощью переменной НаборЗаписей программно обращаться к интересующему нас объекту РегистрСведенийНаборЗаписей.Квитанции. Далее в строке процедуры

НоваяЗапись = НаборЗаписей.Добавить ()

добавляется новая запись в набор записей регистра сведений. Тип возвращаемого методом добавить () значения — РегистрСведенийЗапись.Квитанции. После выполнения этого кода переменная новаяЗапись дает доступ к только что созданной записи. Далее с помощью программных конструкций, подобных

НоваяЗапись.Договор = Договор

производится установка значений полей новой записи информацией с формы.

В последней строке процедуры

```
НаборЗаписей. Записать (Истина)
```

используется метод Записать () для записи новой информации в базу данных. В зависимости от значения входного параметра этот метод ведет себя по-разному. Если параметр имеет значение истина, то перед внесением информации в базу данных существующие записи (удовлетворяющие условиям отбора) будут удалены. Если же в качестве параметра указано ложь, то записи будут дописаны к уже существующим в информационной базе. Заметим, что если входной параметр не указывать, то по умолчанию считается, что он имеет значение истина. Таким образом, первый этап действий по разработке формы учетквитанций завершен. Основной итог заключается в том, что мы получили форму, с помощью которой можно внести в информационную базу сведения о выписанной квитанции. При этом пользователь получает информацию о последней оплаченной квитанции клиента, а также о величине тарифа на дату, когда планируется оплата квитанции. Однако разработка еще не завершена, т. к. не хватает нескольких деталей.

Во-первых, наша система не создает печатный бланк квитанции и, естественно, не производит автоматическое заполнение полей этого документа. А, во-вторых, мы пока не предусмотрели механизма регистрации оплаты (необходимо каким-то образом отмечать, оплачена ли квитанция). В следующем разделе рассматривается создание новой формы с необходимыми для фиксации оплаты элементами управления. А в конце главы приводится разработка автоматизации создания бланка квитанции в приложении Microsoft Word.

Форма регистрации квитанций

Наша задача в этом разделе заключается в разработке удобного механизма фиксации оплаты по квитанциям. Здесь можно пойти двумя путями. Первый путь можно создать новый документ для регистрации квитанций. В этом случае вопросы о том, когда было отмечено, что определенная квитанция оплачена, решаются путем поиска необходимого документа. Здесь же мы для регистрации оплаты воспользуемся уже знакомым объектом конфигурации — обработкой.

Итак, начнем создание новой обработки (это второй путь), которую назовем РегистрацияОплаты. Эти технические действия мы уже выполняли, поэтому пояснения на этот раз будут краткими. Для данной обработки на закладке **Формы** необходимо создать новую форму произвольного вида (рис. 5.17).

верхней части расположим поле В ввода Договор (тип данных – СправочникСсылка. Договоры). Слово договор выбрано и в качестве заголовка, и в качестве имени этого элемента. Ниже размещаются восемь элементов типа надпись, при этом левые надписи являются просто поясняющими и далее в программных процедурах не используются, а правые надписи, выделенные более светлым оттенком, будут автоматически (с помощью программной процедуры) заполняться при выборе договора. Также при выборе договора поле клиент автоматически заполняется фамилией (вместе с именем и отчеством) учащегося.

В программной процедуре (листинг 5.9), выполняемой при выборе договора, производится поиск неоплаченной квитанции, и если она находится, то ее параметры отражаются в элементах формы.

Листинг 5.9. Процедура, выполняемая при выборе договора

```
Процедура ДоговорПриИзменении (Элемент)
ЭлементыФормы.Клиент.Значение = Договор.ФИОучащегося;
ЭлементыФормы.МесяцНачалаОплаты.Значение = "";
ЭлементыФормы.ГодНачалаОплаты.Значение = "";
ЭлементыФормы.МесяцОкончанияОплаты.Значение = "";
```

```
ЭлементыФормы.ГодОкончанияОплаты.Значение = "";
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ *
| ИЗ РегистрСведений.Квитанции
| ГДЕ Договор=&Договор И
| Оплачено = &Индикатор";
Запрос.УстановитьПараметр("Договор", ЭлементыФормы.Договор.Значение);
Запрос.УстановитьПараметр("Индикатор", Ложь);
Результат = Запрос.Выполнить().Выбрать();
Если Результат.Следующий() > 0 Тогда
ЭлементыФормы.МесяцНачалаОплаты.Значение = Результат.МесяцНачала;
ЭлементыФормы.ГодНачалаОплаты.Значение = Результат.МесяцНачала;
ЭлементыФормы.МесяцОкончанияОплаты.Значение = Результат.МесяцОкончания;
ЭлементыФормы.ГодОкончанияОплаты.Значение = Результат.ПодОкончания;
КонецЕсли;
```

КонецПроцедуры

🛃 Конфигуратор (учебная вер	сия) - Конфигурация - [Обр 💶 🗙
🔄 <u>Ф</u> айл <u>П</u> равка Форма Конф	ригурация Отладка Администрирование
<u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	_ & ×
D 🗳 🖬 X 🖻 🚳 🍯	a 🗅 🖒 🦗 🛤 🗳
Форма	
Договор:	X
Учащийся <<Клиент>>	
Начало интервала оплаты	<<МесяцНачалаОплаты>>
	<<ГодНачалаОплаты>>
Окончание интервала оплаты	<<месяцUкончанияUплаты>>
	<<ГодОкончанияОплаты>>
Установить оплату	
📃 Диалог 📃 Модуль 📑 Ре	квизиты
Для получения п САР NUM г	ru ▼ 110,0 ∓11383 x 289

Рис. 5.17. Форма обработки РегистрацияКвитанций

Перейдем теперь к процедуре (листинг 5.10), выполняемой при нажатии кнопки Установить оплату. Ее назначение — для выбранной записи в регистре квитанции в поле оплачено установить значение истина. Для этого мы воспользовались объектом РегистрСведенийМенеджерЗаписи. Квитанции. Следует заметить, что его свойства и методы позволяют обеспечить как программную работу с регистром сведений, так и интерактивную. Обязательным условием применения подобного объекта является независимый режим записи в регистр.

В процедуре, приведенной в листинге 5.10, для создания объекта РегистрСведенийМенеджерЗаписи. Квитанции использован метод СоздатьМенеджерЗаписи(). В результате после выполнения строки

```
Запись = РегистрыСведений.Квитанции.СоздатьМенеджерЗаписи();
```

мы получаем возможность с помощью переменной Запись программно обращаться к объекту РегистрСведенийМенеджерЗаписи. Квитанции.

После этого в процедуре организуется выборка записей из регистра квитанции, которые еще не оплачены. И когда находится неоплаченная квитанция по указанному договору, то устанавливаются значения полей менеджера записи. После этого запись извлекается из базы данных с помощью метода Прочитать (). У извлеченной записи меняется значение реквизита Оплачено, и она с помощью метода Записать () менеджера записи записывается в таблицу базы данных с замещением предыдущей.

Листинг 5.10. Процедура установки оплаты учащегося

```
Процедура УстановитьОплатуНажатие (Элемент)
Запись = РегистрыСведений.Квитанции.СоздатьМенеджерЗаписи();
Договор = ЭлементыФормы.Договор.Значение;
СтруктураОтбора = Новый Структура;
СтруктураОтбора.Вставить ("Оплачено", Ложь);
ВыборкаЗаписей-РегистрыСведений.Квитанции.Выбрать (,, СтруктураОтбора);
Пока ВыборкаЗаписей.Следующий() > 0 Цикл
    Если ВыборкаЗаписей.Договор=Договор Тогда
        Запись.Период = ВыборкаЗаписей.Период;
        Запись.Договор = ВыборкаЗаписей.Договор;
        Запись.Прочитать();
        Если Запись.Выбран() Тогда
             Запись.Оплачено = Истина;
             Запись.Записать ();
             Закрыть ();
        КонецЕсли;
        Прервать;
    КонецЕсли
КонецЦикла
КонецПроцедуры
```

Формирование документа Microsoft Word

В настоящее время интеграция между различными приложениями в практической работе используется достаточно часто. Кроме информационной системы 1С:Предприятие 8.1 в офисной работе очень популярны программы Microsoft Word и Microsoft Excel. Создание большинства офисных документов выполняется именно в этих приложениях. Здесь мы рассмотрим практическую задачу, связанную с автоматическим созданием документа Microsoft Word непосредственно из процедуры, выполняемой в среде 1С:Предприятие 8.1. Реализация этой возможности является завершающим звеном разработки конфигурации данной главы.

Будем считать, что мы располагаем файлом, созданным в приложении Microsoft Word, который содержит бланк квитанции об оплате (см. рис. 5.18). По традиционной технологии этот бланк распечатывается на принтере, а затем заполняется вручную сотрудниками офиса. Наша задача заключается в разработке технологии автоматизированного заполнения полей бланка из процедуры, выполняемой в системе 1С:Предприятие. Мы уже разработали систему регистрации квитанций об оплате, и теперь необходимо обеспечить их распечатку в определенном формате.

На протяжении рассмотрения большинства примеров книги программирование ведется только на встроенном языке системы 1С:Предприятие. Исключением является данный раздел и *глава* 8, где также рассматривается программирование приложений Microsoft Office.

Внесение дополнений в бланк квитанции

Итак, для данной разработки потребуется файл, содержащий квитанцию (рис. 5.18) — документ приложения Microsoft Word 2007. Это обыкновенный документ с необходимым форматированием.

C .) 🖬 🤊 -	ت ت	blank1.docx - Microsoft Word Работа с таблицами	_ = X
	Главная	Вставка	Разметка страницы Ссылки Рассылки Рецензирование Вид Разработчик Конструктор Макет	0
Visu Bas	аl Макросы ic	Запись макр II Ф Пауза Авезопасност Код	роса Аа Аа Э Д Секна Секна сонструктора Секна Сериктора Секна Сериктора Секна Сериктора Секна Сериктора Секна Сериктора Серик	, a
L	$(\gamma,1) \leq 1$	1 2 1 1	· · · · · · · · · · · · · · · · · · ·	16 · · · · 17 · · · 18 🛆 🏛 - 🛐
				â
				Форма № ПД-4
-	U3BF	пение	Наименование получателя платежа: Отделение по Нижегородскому району г. Наименование получателя платежа: Отделение по Нижегородскому району г.	Нижнего
-	HODE	ценне	Новгорода УФК по Нижегородской области ИНН 5260041175 КПП 526001234 (У	лицнглу –
-			л/с 06073772111).	
2			Наименование банка получателя: р/с 40503810800001000099 в ГРКЦ ГУ Банка	а России по
-			Нижегородской оол. г. Нижний Новгород БИК 042201234.	THE HOTER PARTY OF
m			платной образовательной леятельности, лог. № 123/34	ды, получаемые от
-			за обучение в 2007 - 2008 уч. г. С ноябр. 2 007 - Змес	
4			Петров Петров Петрович	
1				
			За	
			Сумма платежа 6 ⁰⁵⁶ руб. 00 коп.	
			Шесть тысяч пятьдесят шесть рублей ОО копсек	
-	Кассиј	р	(Сумма прописью)	-
			Подпись плательщика Дата «»	<u>2 007</u> <u>Γ</u> .
				¥
Стра	ница: 1 из 1	Число слов: 17	77 🕉 русский 🔚	■ 150%

Рис. 5.18. Файл Microsoft Word, содержащий бланк квитанции

В приложении Microsoft Word 2007 нам потребуется работа с элементами управления, поэтому необходимо на ленте открыть вкладку **Разработчик**. С помощью кнопки **Режим конструктора** можно перейти в конструктор, что позволяет размещать в документе необходимые интерфейсные элементы (аналогичные тем, с которыми мы уже сталкивались при работе по конструированию форм в режиме конфигуратора).

В документе, содержащем бланк квитанции, нам потребуется создать несколько элементов управления типа *поле* (более правильное название — *текстовое поле*). Для этого воспользуемся в группе Элементы управления кнопкой Инструменты из предыдущих версий. И здесь в разделе Элементы ActiveX щелкнем по пиктограмме Поле. Функциональное назначение элементов *поле* аналогично ранее встречавшимся в системе 1С:Предприятие элементам *поле ввода*.

После выбора элемента *поле* с помощью мыши следует поместить его в нашем документе (рис. 5.19).



Рис. 5.19. Свойства элемента управления поле

Для того чтобы внешний вид исходного бланка не нарушался, необходимо обеспечить его размещение перед текстом. Для этого следует щелкнуть правой кнопкой мыши по созданному в документе элементу управления и в открывшемся контекстном меню выбрать пункт **Формат объекта**. В появившемся диалоговом окне на вкладке **Положение** следует установить формат **Перед текстом** (рис. 5.20).

В результате проделанных действий внешний вид бланка не изменится, но в нем появятся дополнительные поля для внесения информации. С помощью маркеров, окружающих элемент управления (см. рис. 5.19), мы можем выбрать его размер и месторасположение в документе. Работа с элементами управления в приложении Microsoft Word принципиально не отличается от аналогичной работы при разработ-ке формы в режиме конфигуратора.

Формат элемент	а управл	ения			? ×
Цвета и линии	Размер	Положение	сунок Надпись Зам	иещающий текст 🗎	
Обтекание —					
×		×	×	Ħ	×
в <u>т</u> ексте		вокру <u>г</u> рамки	по к <u>о</u> нтуру	за те <u>к</u> стом	перед текстом
Горизонтальное	выравнив	ание			
					Дополнительно
					ОК Отмена

Рис. 5.20. Окно для установки формата объекта поле

На ленте кнопка Свойства (см. рис. 5.18) позволяет открыть окно свойств активного (выделенного с помощью мыши) элемента управления, которое отражается в левой части рис. 5.19. Установим для свойства Name этого элемента значение Pole_1. Кроме того, с помощью свойства Font следует подобрать желаемый шрифт для отображения в документе номера договора. На рис. 5.19 текстовое поле выделено серым оттенком, что обеспечивается установкой цвета в свойстве BackColor. Однако реально этого делать не требуется — в данном случае цель заключалась просто в придании большей наглядности описываемым действиям.

Аналогичным образом создадим еще семь элементов (рис. 5.21). Для правильной работы дальнейшей программной процедуры необходимо установить у них значения свойства Name, соответственно (сверху вниз) равными Pole_2, Pole_3, Pole_4, Pole_5, Pole_6, Pole_7, Pole_8.

После этого следует выйти из режима конструктора ("отжать" соответствующую кнопку на ленте) и сохранить документ на диске, полное имя для которого следует выбрать C:\blank1.docx. Дело в том, что далее в программной процедуре потребуется указание полного пути месторасположения файла.

Примечание

Можно изменить имя и месторасположение файла, но тогда нужно и скорректировать соответствующую строку в программной процедуре.

Итак, наша конструкторская работа в приложении Microsoft Word завершена — документ для выписки квитанции готов. Рассматриваемая в следующем разделе программная процедура на встроенном языке 1С будет вносить в квитанцию необходимую информацию.

	blank1.docx - Microsoft W	ord	Работа с таблицами	_ = X
Главная Вставка	Разметка страницы Ссылки Рассылки Реце	нзирование Вид Разрабо	тчик Конструктор Макет	0
Visual Макросы Basic Код	оса Аа Аа — Д Конструктора макросов Э д Э Эсенкия Конструктора Д Группировать - Элементи управления	Структура 🎇 Пакеты расшире ХМL	ния Защитить документ • Защитить	о Бласть умента
L · · · Z · · · 1 · · · 2		* 8 * 1 * 9 * 1 * 10 * 1 *	11 - 1 - 12 - 1 - 13 - 1 - 14 -	15 · · · 16 · △ · 17 · · · 6
ИЗВЕЩЕНИЕ	Нанменование получателя плате Новгорода УФК по Нижегородско "/с 06073772111). Наименование банка получателя Нижегородской обл. г. Нижний Н. Наименование платежа: 0733020 платной образовательной деяте за обучение в 2007 – 2008 уч. г Плательщик Петров Петр Петрович	жа: Отделение по Ни й области ИНН 52600 а: р/с 40503810800001 овгород БИК 0422012 01010010000127 Разре ельности. дог. № С ноябр. 2007 - эмес	жегородскому району 41175 КПП 526001234 000099 в ГРКЦ ГУ Ба 34. 2010 - 2010 - 2010 2010 - 2010 - 2010 - 2010 2010 - 2010 - 2010 - 2010 2010 - 2010 - 2010 - 2010 - 2010 2010 - 2010 - 2010 - 2010 - 2010 2010 - 2010 - 2010 - 2010 2010 - 2010	Форма № ПД-4 г. <u>Н</u> ижнего (УИЦ НГЛУ нка России по ходы, получаемые от
-	За Оплата за обучение			
и - -	Сумма платежа <u>6 056</u> руб. (Шесть тыся	00 коп. ач пятьдесят шесть рублей і	00 копеек	
Е Кассир	(Сумма прописью)		205	2 007
·	Подпись плательщика		Дата «»авг	<u> </u>
Страница: 1 из 1 Число слов: 177	🕉 русский 🔚	111		150%

Рис. 5.21. Элементы управления, расположенные в документе приложения Microsoft Word

Разработка процедуры заполнения бланка

Теперь вернемся из приложения Microsoft Word в режим конфигуратора системы 1С:Предприятие. Откроем снова конфигурацию, созданную в данной главе. Далее на форме обработки учетквитанций создадим еще одну кнопку **Передача в Word** (на рис. 5.16 она уже показана), при нажатии которой будет выполняться процедура, приведенная в листинге 5.11.

Листинг 5.11. Процедура для заполнения квитанций в Microsoft Word

```
Процедура ПередачаВWordHажатие (Элемент)

МассивМесяцев = Новый Массив();

МассивМесяцев.Вставить (1, "янв.");

МассивМесяцев.Вставить (2, "февр.");

МассивМесяцев.Вставить (3, "март");

МассивМесяцев.Вставить (4, "апр.");

МассивМесяцев.Вставить (5, "май");

МассивМесяцев.Вставить (6, "июнь");

МассивМесяцев.Вставить (6, "июнь");

МассивМесяцев.Вставить (8, "авг.");

МассивМесяцев.Вставить (9, "сент.");

МассивМесяцев.Вставить (10, "окт.");

МассивМесяцев.Вставить (11, "ноябр.");

МассивМесяцев.Вставить (12, "дек.");

МесяцНачала = ЭлементыФормы.МесяцНачалаОплаты.Значение;
```

```
ГодНачала = ЭлементыФормы. ГодНачалаОплаты. Значение;
Колич = ЭлементыФормы.КоличествоОплачиваемыхМесяцев.Значение;
appWD = Новый СОМОбъект ("Word.Application");
appWD.Visible = Истина;
appWD.Documents.Open("C:\blank1.docx");
appWD.ActiveDocument.Pole 1.Text = Строка (ЭлементыФормы.Договор.Значение);
appWD.ActiveDocument.Pole 2.Text = "C " +
        МассивМесяцев [МесяцНачала] + " " +
        Строка (ГодНачала) + " - " + Строка (Колич) + "мес.";
appWD.ActiveDocument.Pole 3.Text = ЭлементыФормы.Учащийся.Значение;
appWD.ActiveDocument.Pole 4 = "Оплата за обучение";
appWD.ActiveDocument.Pole 5 = Строка (ЭлементыФормы.СуммаОплаты.Значение);
ПараметрыПредметаИсчисления = "рубль, рубля, рублей";
ФорматнаяСтрока = "Л=ru RU";
ВыводимоеЧисло = ЭлементыФормы.СуммаОплаты.Значение;
appWD.ActiveDocument.Pole 6 = ЧислоПрописью (ВыводимоеЧисло,
        ФорматнаяСтрока, ПараметрыПредметаИсчисления);
appWD.ActiveDocument.Pole 7 =
        МассивМесяцев [ЭлементыФормы.МесяцПроизведенияОплаты.Значение];
appWD.ActiveDocument.Pole 8 =
        Строка (ЭлементыФормы. ГодПроизведенияОплаты. Значение);
```

КонецПроцедуры

Процедура отличается от ранее рассмотренных, поэтому мы познакомимся с ее программными конструкциями подробнее. В начале процедуры создается массив элементов и переменной массивмесяцев присваивается ссылка на него. Вообще, *массив* является объектом, предназначенным для работы с временными наборами данных в течение текущего сеанса работы пользователя. Чаще всего использование массивов требуется для некоторых промежуточных действий. В данном случае мы создаем набор соответствий номеров месяцев и их сокращенных названий. Для этого мы могли поступить по-другому — создать стационарный объект, сохраняемый между различными сеансами на энергонезависимом носителе (диске). Однако выбранный путь связан с целью знакомства с массивами на конкретном примере.

В общем, массив можно считать набором значений произвольного типа. При этом каждый элемент массива имеет свой индекс — по значению этого индекса к элементу массива можно обратиться — записать или извлечь информацию. Например, Maccub1[5] — значение пятого элемента массива с именем Maccub1.

Для создания массива в процедуре используется оператор новый в следующей редакции:

```
МассивМесяцев = Новый Массив();
```

Здесь массивмесяцев — имя переменной, которая будет ассоциироваться с создаваемым объектом типа массив. В тексте процедуры после создания массива мы добавляем в него значения строкового типа с помощью метода Вставить (), у которого два параметра. Первый параметр определяет индекс элемента массива, а второй значение элемента массива. Таким образом, в нашей процедуре массив заполняется сокращенными названиями месяцев. В дальнейшем это потребуется при выводе дат в поля квитанции. А именно, в элемент управления Pole_2 заносится название месяца начала оплаты и количество оплачиваемых месяцев. Кроме того, элемент Pole_7 также заполняется строковым названием месяца, в течение которого учащийся собирается произвести оплату.

В процедуре, представленной в листинге 5.11, использована технология OLE (связывание и внедрение объектов). Она предоставляет возможность в одних приложениях работать с объектами, созданными в других приложениях.

В этой технологии существует несколько ключевых понятий. Так, OLEобъектом называется объект, созданный в приложении, отличном от текущего, у которого сохранена связь со своим "родным" приложением. В офисной работе очень популярны приложения Microsoft Word и Microsoft Excel, и, соответственно, используемые документы Microsoft Word и рабочие книги Microsoft Excel могут быть OLE-объектами. Сервером приложения OLE считается приложение, создающее объекты, которые можно поместить в документ-контейнер, создаваемый в другом приложении.

В приведенной в листинге 5.11 процедуре приложение 1С:Предприятие 8 используется как контейнер, а приложение Microsoft Word — как сервер. При таком взаимодействии Microsoft Word предоставляет свои объекты другим приложениям, чем мы и воспользуемся.

В строке

appWD = Новый СОМОбъект ("Word.Application");

мы формируем в переменной appwD ссылку на приложение Microsoft Word. При этом приложение загружается в оперативную память компьютера, но на экране не отображается. Для его показа применяется следующая строка:

appWD.Visible = Истина;

Для дальнейших действий требуется знание месторасположения в компьютере документа Microsoft Word с квитанцией. Как уже говорилось, созданный бланк квитанции мы расположили на диске С: в файле под именем blank1.docx. Далее мы пользуемся методом Open для открытия необходимого документа:

```
appWD.Documents.Open ("C:\blank1.docx")
```

После этого можно начать работу с открытым документом, используя его внутренние объекты, методы и свойства. В последующем программном коде процедуры в объекты документа Microsoft Word (в элементы управления, которые мы сами разместили) переносится информация с формы обработки.

После действий в конфигураторе системы 1С:Предприятие (которые свелись к размещению кнопки на форме и созданию программной процедуры, обслуживающей нажатие кнопки) можно перейти в режим 1С:Предприятие, открыть форму обработки учетКвитанций, внести информацию по создаваемой квитанции и щелкнуть по кнопке передачи информации в Microsoft Word. В результате мы получим заполненную квитанцию.

Заполнение бланка с двумя квитанциями

Мы рассмотрели заполнение бланка, содержащего одну квитанцию. Реально требуется заполнение двух копий квитанций. Для этого в уже знакомой форме создадим соответствующую кнопку Печать двух квитанций (рис. 5.22).

📩 Конфигуратор (учебная версия) - Конфигурация		
🗄 🙅 айл Правка Форма Конфигурация Отладка Администрирование Сервис Окна Справка		
I 🗋 🖬 😹 📾 📾 🔯 🖕 🖕 🕅 🛤 🛛 🎽 🖓 🐥 I		
🚟 Обработка УчетКвитанций: ФормаУчетаКвитанций 🗕	□ ×	
Информация о последней оплате — Расчет следующей оплаты Начало предыдущего интервала Месяц начала оплаты: 0		
<<МесяцНачалаИнтервала>> Год начала оплаты: 0		
<<ГодНачалаИнтервала>>		
Окончание предыдущего интервала Количество оплачиваемых месяцев: U		
С ССМесяцОкончанияИнтервал Расчет суммы 0		
<<ГодОкончанияИнтервала>> Записать квитанцию Передача в Word		
арианалариянанананананананананананананананананана		
Для получения подсказки нажмите F1 САР NUM ru + +440, 338 +132 x 30		

Рис. 5.22. Добавление кнопки для двух квитанций

Для обработки нажатия на новую кнопку разработана соответствующая процедура (листинг 5.12). Она предполагает некоторую модификацию созданного ранее бланка, а именно дублирование текста и создание наряду с восемью текстовыми полями для первой квитанции еще восьми аналогичных элементов управления для второго бланка. Для имен этих новых элементов в процедуре предполагаются следующие имена (порядок элементов сверху вниз, как и в первой квитанции): Pole 1 2, Pole 2 2, Pole 3 2, Pole 4 2, Pole 5 2, Pole 6 2, Pole 7 2, Pole 8 2. Co3данный бланк следует сохранить под именем C:\blank2.docx. Заполненный программной процедурой документ Word показан на рис. 5.23.

Листинг 5.12. Процедура для заполнения бланка с двумя квитанциями

```
Процедура ПередачаBWordHaжатие (Элемент)
МассивМесяцев = Новый Массив();
МассивМесяцев.Вставить (1, "янв.");
МассивМесяцев.Вставить (2, "февр.");
МассивМесяцев.Вставить (3, "март");
МассивМесяцев.Вставить (4, "апр.");
МассивМесяцев.Вставить (5, "май");
```

МассивМесяцев.Вставить (6, "июнь");

```
МассивМесяцев.Вставить (7, "июль");
МассивМесяцев.Вставить (8, "авг.");
МассивМесяцев.Вставить (9, "сент.");
МассивМесяцев.Вставить (10, "окт.");
МассивМесяцев.Вставить (11, "ноябр.");
МассивМесяцев.Вставить (12, "дек.");
МесяцНачала = ЭлементыФормы.МесяцНачалаОплаты.Значение;
ГодНачала = ЭлементыФормы.ГодНачалаОплаты.Значение;
Колич = ЭлементыФормы.КоличествоОплачиваемыхМесяцев.Значение;
appWD = Новый СОМОбъект ("Word.Application");
аррWD.Visible = Истина;
appWD.Documents.Open("C:\blank2.docx");
appWD.ActiveDocument.Pole 1.Text = Строка (ЭлементыФормы.Договор.Значение);
appWD.ActiveDocument.Pole 1 2.Text = Строка (ЭлементыФормы.Договор.Значение);
appWD.ActiveDocument.Pole 2.Text = "C " +
        МассивМесяцев [МесяцНачала] + " " +
        Строка (ГодНачала) + " - " + Строка (Колич) + "мес.";
appWD.ActiveDocument.Pole_2_2.Text = "C " +
        МассивМесяцев [МесяцНачала] + " " +
        Строка (ГодНачала) + " - " + Строка (Колич) + "мес.";
appWD.ActiveDocument.Pole 3.Text = ЭлементыФормы.Учащийся.Значение;
appWD.ActiveDocument.Pole 3 2.Text = ЭлементыФормы.Учащийся.Значение;
appWD.ActiveDocument.Pole 4 = "Оплата за обучение";
appWD.ActiveDocument.Pole 4 2 = "Оплата за обучение";
appWD.ActiveDocument.Pole 5 = Строка (ЭлементыФормы.СуммаОплаты.Значение);
appWD.ActiveDocument.Pole 5 2 = Строка (ЭлементыФормы.СуммаОплаты.Значение);
ПараметрыПредметаИсчисления = "рубль, рубля, рублей";
ФорматнаяСтрока = "Л=ru RU";
ВыводимоеЧисло= ЭлементыФормы.СуммаОплаты.Значение;
appWD.ActiveDocument.Pole 6 = ЧислоПрописью (ВыводимоеЧисло,
        ФорматнаяСтрока, ПараметрыПредметаИсчисления);
appWD.ActiveDocument.Pole 6 2 = ЧислоПрописью (ВыводимоеЧисло,
        ФорматнаяСтрока, ПараметрыПредметаИсчисления);
appWD.ActiveDocument.Pole 7 =
        МассивМесяцев [ЭлементыФормы.МесяцПроизведенияОплаты.Значение];
appWD.ActiveDocument.Pole 7 2 =
        МассивМесяцев [ЭлементыФормы.МесяцПроизведенияОплаты.Значение];
appWD.ActiveDocument.Pole 8 =
        Строка (ЭлементыФормы. ГодПроизведенияОплаты. Значение);
appWD.ActiveDocument.Pole 8 2 =
        Строка (ЭлементыФормы. ГодПроизведенияОплаты. Значение);
КонецПроцедуры
```

G	A Sector Padora c таблицами – C Padora c таблицами – C	= X
	Главная Вставка Разметка страницы Ссылки Рассылки Рецензирование Вид Разработчик Конструктор Макет	
Visual M Basic	Запись макроса но Пауза Макросы Код Макросы Код Макросы Код Макросы Код Макросы Код Макросы Код Макросы Код Кор Кор Кор Кор Кор Кор Кор Кор	
	· 2 · 1 · 3 · 1 · 4 · 1 · 5 · 1 · 6 · 1 · 7 · 1 · 8 · 1 · 9 · 1 · 10 · 1 · 11 · 1 · 12 · 1 · 13 · 1 · 14 · 1 · 15 · 1 · 16 · 🔬 · 17 · 1 ·	ā
	innate domain the intervent of source of the performent of the intervent of the of	
-	платной образовательной деятельности. дог. № 123///	
-	за обучение в 2007 – 2008 уч. г с инын 2 обоч инсе	
4	Плательшик Иванов Иван Петрович	
-	3 оплата за обучение	
-	Сумма платежа 15 050 руб. 00 коп.	
1	Пятнадцать тысяч пятьдесят рублей ОО копсек	
	(Сумма прописью)	
-	Подпись плательщика Дата «»авг2008 г.	
·	Форма № ПД-4	
	Наименование получателя платежа: Отделение по Нижегородскому району г. Нижнего	
-	Новгорода УФК по Нижегородской области ИНН 5260041175 КПП 526001234 (УИЦ НГЛУ	
-	л/с 06073772111).	
σ	Наименование банка получателя: р/с 40503810800001000099 в ГРКЦ ГУ Банка России по	
-	Нижегородской обл. г. Нижний Новгород БНК 042201234.	
	Наименование платежа: 0733020101001000127 Разрешение 1 (пункт 1). Доходы, получаемые от	
1	платной образовательной деятельности дог. № 125///	
	за обучение в 2007 – 2008 уч. г С июль 2 008 - 7мес.	1
11	Иванов Иван Петрович	٥
4		¥
Страница	а: 1 из 1 Число слов: 177 🕉 русский 🔛 🛛 📑 🗐 🖓 🔅 💬 🔍	

Рис. 5.23. Заполненный бланк с двумя квитанциями

глава 6



Извлечение информации с помощью запросов

Если в целом оценить примеры, приведенные в предыдущих главах, то можно сказать, что мы работали в основном с первичной информацией, сосредоточенной в разнообразных объектах — справочниках, регистрах и документах. И если бы нас интересовали данные только в такой исходной форме, то больше в принципе ничего и не нужно. Так, всегда можно открыть необходимый справочник, просмотреть содержимое регистра или документ из списка документов.

Однако для управляющего звена организации представляет интерес информация в существенно обработанном виде (итоговые и сводные данные). Например, в *главе 1* мы создали справочник Специалисты, в котором присутствует табличная часть, отражающая различные места работы (рассматривается ситуация, что у определенного специалиста может быть не одно место работы). И, скажем, может возникнуть необходимость получения отчета обо всех сторонних организациях, где наши сотрудники дополнительно работают.

Конечно, можно было бы при разработке данного справочника пойти по пути предварительного создания вспомогательного (подстановочного) справочника со списком организаций. Да, действительно, именно так и стоило бы поступить, если бы на этапе разработки планировалось, что подобный справочник нам будет в дальнейшем необходим. В данной же ситуации (когда конфигурация разработана и информационная база заполнена данными) для получения отчета требуется проанализировать различные организации исходя из информации, располагающейся в табличных частях элементов справочника Специалисты. Один из вариантов интересующего нас отчета приведен в табл. 6.1.

Название организации
АО "Связьинвест"
НМСИ
АО "Меда"
МТКИ

Таблица 6.1. Список организаций

Технически для составления подобного отчета мы должны перебрать все элементы справочника Специалисты и изучить их табличные части. Понятно, что без автоматизации данную работу выполнять не очень интересно.

Другая аналогичная задача может возникнуть при работе со справочником курсы, также созданным в *главе 1*. В нем представлена информация о дате начала занятий, их продолжительности в неделях, стоимости обучения и преподавателе. При работе с подобным справочником может возникнуть, например, следующая задача — составить список курсов продолжительностью не менее пяти недель, стоимость обучения на которых не превышает 1500 рублей.

Для решения подобных задач в системе 1С:Предприятие используется специальный объект, который называется *запросом*. По существу любой запрос представляет собой требование к системе извлечь выборку с необходимой информацией из базы данных. Но кроме обыкновенного отбора запросы предоставляют еще много полезных возможностей — группировка отобранных данных, вычисление итогов и др. Запросы удобно использовать, когда требуется сформировать сложную выборку данных, сгруппированную и отсортированную необходимым образом.

В конфигурации, разработанной в *главе 5*, мы создали регистр сведений квитанции. Используя запрос с группировкой по учащимся, можно легко подсчитать, кто какую сумму уже оплатил за обучение. Также с помощью запроса можно выбрать тех учащихся, кто имеет неоплаченные квитанции.

С запросами мы уже встречались в книге. В разработанной конфигурации из *главы 1* рассматривалось, каким образом с помощью запроса можно подсчитать суммарную оплату учащимися различных курсов. Также мы выяснили, как получить ответ на вопрос, кто и сколько уже оплатил за конкретный курс, например за курс "Базы данных". Фактически подобные примеры, рассмотренные ранее, являются неким анонсом данной главы. Здесь технология составления запросов будет рассмотрена, начиная с основных положений. На разнообразных задачах вы получите практические навыки по формированию учетной и сводной информации.

Язык запросов. Основные конструкции

Если взглянуть на первые главы в целом, то можно сказать, что мы на примерах познакомились с различными конструкциями встроенного языка программирования системы 1С:Предприятие 8.1. Как и в большинстве подобных языков, действия разработчика сводятся к определению точной последовательности шагов, направленных на достижение конкретной цели, и записи этих шагов в понятном компьютеру виде. Все начинается с формулировки задачи, затем составляется алгоритм действий, а далее подбираются соответствующие программные конструкции языка (этап кодирования).

В концепцию запроса заложена другая идея — мы лишь формулируем задачу для системы на понятном ей языке (языке запросов). При построении запроса необходимо максимально точно описать, какой результат нам нужно получить. Сами же действия, требуемые для извлечения нужной информации, определяются и осуществляются компьютером. В этом разделе мы познакомимся с основными конструкциями языка запросов на различных примерах. Однако при этом мы не ограничимся только справочной информацией, а проиллюстрируем почти все приведенные запросы практическими примерами.

В качестве первого задания рассмотрим запрос для извлечения информации, расположенной в полях договор и СуммаПоКвитанции регистра сведений Квитанции (речь пойдет о конфигурации, разработанной в *главе 5*). Нам необходимо получить список квитанций с указанием номера договора и выписанной суммы. Такой несложный запрос выглядит следующим образом:

```
ВЫБРАТЬ Договор, СуммаПоКвитанции
ИЗ РегистрСведений.Квитанции
```

В результате его выполнения мы получим таблицу из набора строк, в каждой из которых две колонки. Число строк в таблице соответствует числу имеющихся записей в регистре квитанции.

В приведенном запросе использовано ключевое слово выбрать, позволяющее указать список полей для данного запроса. Другое ключевое слово из позволяет определить объекты конфигурации, участвующие в рассматриваемом запросе. Так, в данной ситуации используется регистр сведений квитанции, из которого извлекается информация о двух его полях (измерении договор и ресурсе СуммаПоКвитанции).

В принципе язык запросов позволяет обходиться и без указания ключевого слова из, но в этом случае в конструкции требуется указать полные имена полей, участвующих в запросе. Рассмотренный выше текст можно было оформить и подругому:

```
ВЫБРАТЬ РегистрСведений.Квитанции.Договор,
РегистрСведений.Квитанции.СуммаПоКвитанции
```

Вообще существует много разнообразных синтаксических конструкций запросов. И далее мы увидим, каким образом можно выполнять подобные запросы в системе 1С:Предприятие 8.1. При этом, кроме конструкций языка запросов, безусловно, потребуется и уже в некоторой степени знакомый нам встроенный язык программирования системы 1С:Предприятие 8.1.

Вводная часть завершена, и настала очередь перейти к практике. Начнем с того, что в конфигураторе откроем конфигурацию, разработанную нами в *главе 5*. Результаты запросов потребуется где-то отражать, и для этого можно использовать элементы управления, располагаемые на форме обработки. Для отображения табличных данных хорошо подходит табличный документ. В примерах данной главы этими ресурсами мы и воспользуемся для отображения результатов.

Обработка ИзвлечениеИнформации

Для начала создадим новую обработку. Непосредственные технические действия для этого нам уже знакомы. Ключевой параметр любого объекта конфигурации — его имя, и для него мы укажем значение извлечениеИнформации (рис. 6.1).

В этой обработке нам потребуется подчиненный объект конфигурации — *фор*ма, на которой будет размещено несколько элементов управления. Все основные технические действия по разработке произойдут на закладке Модуль, предназначенной для размещения текстов программных процедур на встроенном языке.

🛃 Конфигуратор (учебная версия)	- Конфигурация		
<u>Ф</u> айл <u>П</u> равка Ко	нфигурация Отлад	ка Администрирование <u>С</u> е	рвис	
<u>О</u> кна Сп <u>р</u> авка				
I 🗋 🛋 🖬 🛛 🗶 🖓	à 6 5 Q	5 C 🗿 M	» •	
🔅 Обработка ИзвлечениеИнформации 💶 🗆 🗙				
• Основные	Ma and	14		
Данные	имя.	извлечениеинформации		
Формы	Синоним:	Извлечение информации		
Макеты	Комментарий:			
Подсистемы				
Права				
Интерфейсы				
Прочее				
Действия 🔹 <	Назад Дали	ее» Закрыть С	правка	
Для получения подск	азки нажмите F1	CAP NUM	ru• //	

Рис. 6.1. Окно редактирования обработки ИзвлечениеИнформации

С закладки **Основные** перейдем на закладку **Формы**, где создадим форму (рис. 6.2) без дополнительных панелей инструментов. Как видно, на форме расположены уже знакомые элементы управления (кнопка и поле списка). Технически, как мы уже знаем, элементы размещаются на форме с использованием пункта Вставить элемент управления (меню **Форма**).

В системе 1С:Предприятие 8.1 существует объект *"список значений"*. Это несохраняемый в базе данных объект, который предназначен для формирования динамических наборов данных. Список значений может быть заполнен значениями любого типа.

Примечание

Объект "список значений" является коллекцией значений, составляющих список.

Для визуального представления объекта "список значений" в системе 1С:Предприятие можно использовать элемент управления *поле списка*, который предназначен для выбора одного или нескольких значений из списка.

Определимся с именами объектов, что необходимо для исключения противоречий с приведенной далее процедурой. Для кнопки выберем имя ОтобратьИнформацию, а для элемента управления *поле списка* — СписокОтобранныхДанных. Для того чтобы в режиме работы 1С:Предприятие появился раздел для открытия созданной формы обработки, необходимо установить эту форму в качестве основной (рис. 6.3). На этом визуальное проектирование завершено, и можно перейти к написанию программного кода. Разработанная форма будет базовой для ряда последующих процедур, выполняемых при нажатии кнопки ОтобратьИнформацию. В последующих примерах потребуется лишь изменение программного кода.

🛃 Конфигуратор (учебная версия) - Конфигурация	
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администриров	ание
<u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
🗋 🛁 🔏 🖻 📾 🕌 🖄 🖕 🛤	» •
🚟 Обработка ИзвлечениеИнформации: Форма 🚬	□ ×
Форма	^
Отобрать информацию	
Список отобранных данных.	
	~
📓 Диалог 🔳 Модуль 📑 Реквизиты	>
Для получения п САР NUM ги - 27, 73 📫 300 х 94	

Рис. 6.2. Форма обработки ИзвлечениеИнформации

🚵 Конфигуратор (учебная версия) - Конфигурация	
Файл <u>П</u> равка Ко Справка	нфигурация Отладка Администрирование	<u>С</u> ервис <u>О</u> кна
🗋 🛋 🖬 🐰 🛛	a 📾 🥌 🖻 to et 🕅 M	» •
💣 Обработка Из	влечениеИнформации	_ 🗆 ×
Основные Данные • Формы	Основная форма обработки: Форма	
Макеты		
Подсистемы Права	⊡-∰ Формы ————————————————————————————————————	
Интерфейсы Прочее		
Действия 👻 🦂	(Назад Далее) Закрыть	Справка
Для получения подск	азки нажмите F1	NUM ru 🕶 🎢

Рис. 6.3. Закладка Формы окна редактирования объекта конфигурации

Конструкции ВЫБРАТЬ и ИЗ

Функционирование формы заключается в том, что при нажатии кнопки Отобрать информацию список значений СписокОтобранных Данных должен заполняться информацией о квитанциях (договор и сумма по квитанции). Процедура, которую следует для этого разработать, приведена в листинге 6.1.

```
Листинг 6.1. Процедура, выполняемая по нажатию кнопки Отобрать информацию
```

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Договор, СуммаПоКвитанции
| ИЗ РегистрСведений.Квитанции";
Результат = Запрос.Выполнить().Выбрать();
СписокОтобранныхДанных.Очистить();
Пока Результат.Следующий() > 0 Цикл
СписокОтобранныхДанных.Добавить (Строка (Результат.Договор) +
" " + Строка (Результат.СуммаПоКвитанции) + " руб.");
КонецЦикла;
КонецПроцедуры
```

В первой строке создается новый объект типа Запрос, а далее в его свойстве текст непосредственно формируется текст запроса. Это тот текст, который мы уже разобрали в начале главы. Как часто получается для большинства запросов, его конструкция занимает более одной строки. В этом случае для продолжения строки используется символ вертикальной черты (1).

В следующей после текста запроса программной конструкции

Результат = Запрос.Выполнить().Выбрать()

сначала используется метод Выполнить (), что приводит к выполнению запроса, и к результату выполнения этого метода применяется другой метод — Выбрать (), который позволяет произвести выборку информации из выполненного запроса. Фактически метод Выбрать () подготавливает работу еще одного метода — Следующий (), который осуществляет выборку очередной строки из результата выполнения запроса. Это позволяет последовательно перебрать строки в полученной выборке.

Далее в процедуре (извлеченной с помощью запроса информацией) заполняется элемент управления СписокОтобранныхДанных. Для этого он предварительно очищается с помощью метода Очистить (), и затем организуется цикл перебора строк, отобранных в результате запроса. В цикле метод добавить () позволяет пополнить список на форме информацией об очередной квитанции (о номере договора и сумме).

Примечание

Строка () — функция преобразования значений, которая преобразует входной параметр в значение типа Строка. Входной параметр может иметь произвольный тип данных.

На рис. 6.4 показан результат выполнения процедуры, приведенной в листинге 6.1, в режиме 1С:Предприятие. В итоге информация из регистра сведений квитанции нашла отражение в списке.

🏪 1С:Предприятие (учебная версия) - Конфигура	ция		
📱 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка			
I 🗋 🔟 🖬 🕺 🛍 🥌 🖻 🖕 🖬 🗌	▼ n n 1	唑 🗸 🛙 🎹 🧾 🧟 🛛 M 🛛	M+ M- 🎇 🛨
🗋 Форма 📃 🗆 🗙	ք Список Квитанции		_ 🗆 ×
	Действия 🗸 🤤 🥞 🛃 🔤 🎼	ə) 🏦 🔎 🔟 - 🌾 🛛 🚺 🕻	?
Отоорать информацию	Период 🝷 Договор	Сумма по квитанции Оплаче	но Месяц наГ 🛆
Список отобранных данных:	10.08.2007 15:21:15 123/34	1 930 🗸	6
123/34_1 930 py6.	10.08.2007 15:22:12 123/34	1 970 🗸	7
123/34 1 970 риб.	10.08.2007 17:46:07 123/34	6 000 🗸	8
123/34 6 000 руб.	■ 17.08.2007 15:00:18 123/35	10 000 🗸	8
			~
< >	<		Σ
Форма 🌈 Список Квитанции			
Для получения подсказки нажмите F1			CAP NUM

Рис. 6.4. Результат выполнения запроса по выписанным квитанциям

Упорядочивание в запросе

Если поля или названия объектов достаточно длинны, то для них удобно назначить псевдоним с помощью ключевого слова как, что позволяет в дальнейшем обеспечить более компактную запись программных конструкций.

Изменим процедуру (см. листинг 6.1) заполнения списка результатами запроса. Теперь она должна выглядеть в соответствии с листингом 6.2, где для поля Суммапоквитанции введен псевдоним — Сумма. Этот псевдоним в тексте запроса мы используем в новой конструкции упорядочить по, что обеспечивает сортировку полученной информации. Ключевое слово убыв говорит о том, что информация должна быть отсортирована по убыванию сумм в квитанциях.

Листинг 6.2. Процедура выполнения запроса с упорядочиванием

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ Договор,
| СуммаПоКвитанции
| КАК Сумма
| ИЗ РегистрСведений.Квитанции
| УПОРЯДОЧИТЬ ПО Сумма УБЫВ";
Результат = Запрос.Выполнить ().Выбрать ();
СписокОтобранныхДанных.Очистить ();
Пока Результат.Следующий () > 0 Цикл
СписокОтобранныхДанных.Добавить (Строка (Результат.Договор) +
" " + Строка (Результат.Сумма) + " руб.");
КонецЦикла;
КонецПроцедуры
```

На рис. 6.5 показан результат выполнения такого запроса с упорядочиванием по сумме (наибольшие значения оказались вверху списка).

В случае необходимости упорядочивания по возрастанию следует применить конструкцию с использованием ключевого слова возр. Например, для рассмотренной ситуации это может выглядеть следующим образом:

УПОРЯДОЧИТЬ ПО Сумма ВОЗР

Примечание

В предложении упорядочить по через запятую перечисляются условия, в соответствии с которыми необходимо упорядочить результат запроса. Выбранные данные упорядочиваются сначала по первому условию, затем по второму и т. д. В результате извлеченная запросом информация становится структурированной и удобной для восприятия пользователями.

陽 1С:Предприятие (учебная версия) - Конфигурация 💶 🗙
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
🛮 🗋 🔲 🐰 🖻 🛍 🐥 🔚 🛅 🌠 м. м+ м- 🎘 🖕
🖹 Форма 📃 🗆 🗙
Отобрать информацию
Список отобранных данных:
123/35 10 000 руб.
123/34 6 000 py6.
123/34 1 9/0 py6.
123/34 1 330 pyo.
🔓 Форма
Для получения подсказки нажмите F1 CAP NUM

Рис. 6.5. Результат выполнения запроса с упорядочиванием по сумме

Отбор максимальных (минимальных) значений

При большом объеме информации в результате запроса часто требуется посмотреть несколько наиболее больших или, наоборот, маленьких значений. Например, это нужно для получения списков самых дорогих услуг или товаров. В этом случае в конструкцию выврать добавляется ключевое слово первые. Например, если нас интересуют две квитанции с самыми маленькими выписанными суммами, то для этого можно использовать процедуру, приведенную в листинге 6.3.

На рис. 6.6 отражен результат работы этой процедуры в режиме 1С:Предприятие: показана информация (номер договора и сумма) о двух минимальных по сумме квитанциях.
Листинг 6.3. Процедура отбора двух минимальных по сумме квитанций

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ ПЕРВЫЕ 2 Договор, СуммаПоКвитанции
| КАК Сумма
| ИЗ
| РегистрСведений.Квитанции
| УПОРЯДОЧИТЬ ПО Сумма ВОЗР";
Результат = Запрос.Выполнить ().Выбрать ();
СписокОтобранныхДанных.Очистить ();
Пока Результат.Следующий () > 0 Цикл
СписокОтобранныхДанных.Добавить (Строка (Результат.Договор) +
" " + Строка (Результат.Сумма) + " руб.");
КонецЦикла;
```

КонецПроцедуры

🏪 1С:Предприятие (учебная версия) - Конфигур 💶 🗙
<u> <u> </u> </u>
🛮 🗋 📓 🕺 🐂 💙 🖬 🛅 🧟 м. м+ м- 🎇 🖕
🖹 Форма 📃 🗆 🗙
Отобрать информацию Список отобранных данных:
123/34 1 930 руб. 123/34 1 970 руб.
🖹 Форма
Для получения подсказки нажмите F1 САР NUM

Рис. 6.6. Результат выполнения отбора двух квитанций с минимальными суммами

Если бы нас интересовали, наоборот, две квитанции на максимальные суммы, то в тексте листинга 6.3 необходимо было бы изменить только конструкцию, связанную с упорядочиванием полученных данных. В этом случае текст запроса должен выглядеть следующим образом:

```
Запрос.Текст = "ВЫБРАТЬ ПЕРВЫЕ 2 Договор, СуммаПоКвитанции
| КАК Сумма
| ИЗ РегистрСведений.Квитанции
| УПОРЯДОЧИТЬ ПО Сумма УБЫВ";
```

Ключевое слово ИЕРАРХИЯ

Мы рассмотрели упорядочивание, которое в данной языковой конструкции весьма эффективно для объектов без групп (без иерархии). Перед рассмотрением следующего примера внесем небольшие изменения в справочник договоры.

Во-первых, необходимо указать, что он будет иметь иерархическую структуру. А, во-вторых, для выполнения данного примера следует создать в справочнике две группы (рис. 6.7). Понятно, что для этого следует перейти в режим 1С:Предприятие. Ранее внесенный договор учащегося Петрова перенесем в группу **Филиал 1**, в эту же группу добавим еще Алексеева (договор 123/35), а в группу **Филиал 2** запишем Афанасьева (договор 127/77).



Рис. 6.7. Создание групп в справочнике Договоры

Сконструируем запрос, который позволяет выбрать из справочника договоры название договора и фамилию плательщика, но с группировкой по филиалам. Результат выполнения подобного запроса должен выглядеть аналогично рис. 6.8. В этом случае договоры, заключенные в разных филиалах, будут сгруппированы в удобном виде. Подобное упорядочивание достигается за счет использования ключевого слова иерархия. В листинге 6.4 приведен измененный текст процедуры с запросом, который обеспечивает данный результат упорядочивания договоров (см. рис. 6.8).

Листинг 6.4. Процедура запроса с учетом иерархии

```
Процедура ОтобратьИнформациюНажатие(Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Наименование КАК Наим,
| ФИОплательщика
| ИЗ
| Справочник.Договоры
| УПОРЯДОЧИТЬ ПО Наименование ИЕРАРХИЯ";
```

```
Результат = Запрос.Выполнить().Выбрать();
СписокОтобранных.Данных.Очистить();
Пока Результат.Следующий() > 0 Цикл
СписокОтобранных.Данных.Добавить(Строка(Результат.Наим) +
" " + Строка(Результат.ФИОплательщика));
КонецЦикла;
КонецПроцедуры
```



Рис. 6.8. Пример использования ключевого слова ИЕРАРХИЯ в тексте запроса

Исключение в отборе повторяющейся информации

Ранее в листинге 6.2 мы привели текст процедуры, которая позволяла извлечь информацию о выписанных квитанциях (из регистра квитанции). В этом случае по результату запроса мы видим, по какому договору выписывалась квитанция и указанную в ней сумму. Однако если задача заключается в получении списка всех договоров, по которым выписывались квитанции, то рассмотренная процедура не подходит. Скажем, необходимо отделить договоры, по которым квитанции выписывались, от тех, по которым они еще не выписывались. В языке запросов существует ключевое слово различные, позволяющее отобрать только отличающиеся строки. Иначе говоря, в выборке результата выполнения запроса, включающего это ключевое слово, повторений не будет.

Вернемся к нашему примеру и будем выбирать из регистра квитанции договоры с использованием ключевого слова различные. Текст процедуры, в которой реализуется подобный отбор договоров, приводится в листинге 6.5, а на рис. 6.9 показан результат ее работы в режиме 1С:Предприятие.

Примечание

На компакт-диске, прилагаемом к книге, процедуры, относящиеся к этой главе, приводятся в различных обработках. Соответствие названий обработок листингам приведено в приложении в конце книги.

Листинг 6.5. Процедура отбора договоров, по которым выписывались квитанции

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ Договор
| ИЗ РегистрСведений.Квитанции
| УПОРЯДОЧИТЬ ПО Договор ВОЗР";
Результат = Запрос.Выполнить ().Выбрать ();
СписокОтобранныхДанных.Очистить ();
Пока Результат.Следующий () > 0 Цикл
СписокОтобранныхДанных.Добавить (Строка (Результат.Договор));
```

КонецЦикла;

КонецПроцедуры

🏪 1С:Предприятие (учебная версия) - Конфигур 💶 🗙
📱 🛨 айл Правка Операции Сервис Окна 1С:Предприятие (учебна
🛮 🗋 🔐 🛛 🔏 🍣 📲 🛅 🌌 м м+ м- 🎘 🖕
🖿 Форма 📃 🗆 🗙
Отобрать информацию
123/35 123/34
🔓 Форма
Для получения подсказки нажмите F1 САР NUM

Рис. 6.9. Отбор договоров, по которым выписывались квитанции

В результате (рис. 6.9) в списке договоров мы не увидим номеров, по которым еще не выписывались квитанции. Для того чтобы в этом убедиться, просто внесите еще одного нового клиента, и его в отобранных данных не будет.

Ключевое слово ГДЕ

До этого момента нам еще не встречалось ключевое слово языка запросов где. Оно позволяет задать условие отбора данных из исходных таблиц. В результате в запросе окажутся отобранными только те записи, для которых выполняется заданное условие. Применим эту возможность к процедуре, приведенной в листинге 6.5, с тем, чтобы отобрать только те договоры, по которым были оплаты.

До отметки оплат в регистр квитанции мы включили реквизит оплачено, который может принимать значения истина или ложь. И в запросе в листинге 6.6 конструкция с ключевым словом где позволяет оставить в отобранных данных только те договоры, у которых рассматриваемый реквизит установлен в значение истина.

Листинг 6.6. Процедура отбора договоров, по которым были оплаты

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ РАЗЛИЧНЫЕ Договор
| ИЗ
| РегистрСведений.Квитанции
| ГДЕ Оплачено=Истина
| УПОРЯДОЧИТЬ ПО Договор ВОЗР";
Результат = Запрос.Выполнить().Выбрать();
СписокОтобранныхДанных.Очистить();
Пока Результат.Следующий() > 0 Цикл
СписокОтобранныхДанных.Добавить (Строка (Результат.Договор));
КонецЦикла;
КонецПроцедуры
```

Эффект выполненных изменений продемонстрирован на рис. 6.10, где показано окно с результатами отбора и окно регистра сведений квитанции.

🏪 1С:Предприятие (учебная версия) - Конфигура	ация
<u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
I 🗋 🔟 🖉 🖧 🖻 📾 🥌 🖉 🕁 🖒 🖛 🗌	👻 🎽 🎹 🛅 🌠 🛛 M M+ M- 🎘 🗸
Р Форма	👖 Список Квитанции 📃 🗆 🗙
	Действия 🗸 😫 😫 🔩 🎽 🖦 🔛 🏹 🛅 🗃 - 🏹 🚬 🂙
	Период 🔹 Договор Сумма по Оплачено 🎼
Отобрать информацию	📨 10.08.2007 15:21:15 123/34 1 930 🗸
	🖾 10.08.2007 17:46:07 123/34 8 000 🗸 🔰
	☑ 07.09.2007 0:00:00 123/34 8 970 ✔
Список отобранных данных:	☑ 20.10.2007 0:00:00 123/35 10 000 ✔
123/35	☑ 27.08.2008 13:57:29 123/35 10 000 ✔
123/34	🖾 25.09.2008 10:22:42 187/77 6 000
	< >
🗅 Форма 🥤 Список Квитанции	
Для получения подсказки нажмите F1	CAP NUM

Рис. 6.10. Форма списка регистра Квитанции и результат отбора договоров

Использование логических операторов в запросе

В конструкции где можно использовать логические операторы (и, или, не), которые позволяют построить достаточно сложные логические выражения. Скажем, применительно к предыдущему примеру, нас могут интересовать договоры, по которым имеются оплаченные квитанции на сумму более 7000 рублей. Текст процедуры с необходимым для этого запросом приведен в листинге 6.7, а результат работы в режиме 1С:Предприятие — на рис. 6.11. Как видно, в отличие от рис. 6.10 в список отобранных данных за счет указанного условия попал только один договор.

Листинг 6.7. Процедура отбора договоров с использованием ключевого слова и

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ Договор
| ИЗ
| РегистрСведений.Квитанции
| ГДЕ Оплачено=Истина И СуммаПоКвитанции > 7000
| УПОРЯДОЧИТЬ ПО Договор ВОЗР";
Результат = Запрос.Выполнить ().Выбрать ();
СписокОтобранныхДанных.Очистить ();
Пока Результат.Следующий () > 0 Цикл
СписокОтобранныхДанных.Добавить (Строка (Результат.Договор));
КонецЦикла;
```

КонецПроцедуры

🏪 1С:Предприятие (учебная версия) - Конфигурация		
📱 🛨 айл Правка Операции Сервис Окна Справка		
I 🗋 🛁 🛛 X 🖻 🛍 🥌 🖾 🖕 🖒 🕐 🐘	🔻 🖗 🖗 🍟 🎆 🔜 🌠 M M+ M-	- 🎇
📑 Форма 📃 🗆 Х	🥤 Список Квитанции 🔔	$\Box \times$
	Действия • 🔤 🔄 🔩 🍇 🔤 🔜 🤲 🎢 🖆 🖷 - 🌾	\ Q
	Период • Договор Сумма по Оплачено	MA
	Image: Non-2007 15:21:15 123/34 1 930 ✓ Image: Non-2007 13:40:07 1 22/34 0 000 0 000	
	Image: Non-Object Image: Nojbect Image: Nojbect <th< td=""><td>_</td></th<>	_
Список отобранных данных:	20.10.2007 0:00:00 123/35 10 000 ✓	
123/35	☑ 27.08.2008 13:57:29 123/35 10 000 ✓	
	<	×
Родма Список Квитанции		
Для получения подсказки нажмите F1	CAP NL	JM <i> </i> //

Рис. 6.11. Результат выполнения запроса с использованием логического оператора

Приведем теперь пример запроса, в котором комбинируются конструкции или и и. Скажем, нас интересуют все квитанции, которые выписывались для оплаты

в 2007 году с началом интервала оплаты в сентябре или октябре. Соответственно, в интересующих нас записях реквизит годначала должен равняться 2007, а месяцначала может принимать значение 9 или 10. В листинге 6.8 приведена процедура, реализующая необходимую выборку информации в этом случае, а на рис. 6.12 показан вариант выполнения отбора в режиме 1С:Предприятие.

Листинг 6.8. Процедура отбора договоров с использованием логических конструкций

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос. Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ
| Договор, МесяцНачала, ГодНачала
  ИЗ
| РегистрСведений.Квитанции
       (МесяцНачала=9 ИЛИ МесяцНачала=10)
 ГЛЕ
| И ГолНачала=2007
| УПОРЯДОЧИТЬ ПО Договор ВОЗР";
Результат = Запрос.Выполнить().Выбрать();
СписокОтобранныхДанных.Очистить ();
Пока Результат.Следующий () > 0 Цикл
    СписокОтобранныхДанных.Добавить (Строка (Результат.Договор) + "
        Строка (Результат. МесяцНачала) + "
                                               " +
        Строка (Результат. ГодНачала));
КонецЦикла;
```

КонецПроцедуры

🏪 1С:Предприятие (учебная версия) - Конфигу	ация	
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка		
	🗸 🕅 🕅 🕞 🗸 📰 📰	餐 М М+ М- 🎘 🖕
🕒 Форма 📃 🗆 🗙	f Список Квитанции	_ 🗆 ×
Птобрать информацию	Действия • 😫 😫 🔩 🍇 📾 🤲 🏹 🎦 🖷 • 🕅	2 ?
	Период 🔹 Договор Сумма по Оплачено	Месяц начала Год начала 🛆
Список отобранных данных:	ID:08.2007 15:21:15 123/34 1 930 ✓	6 2 007
123/35 10 2 007	ID:08.2007 17:46:07 123/34	9 2 007
123/34 9 2 007	07.09.2007 0:00:00 123/34 8 970 🗸	11 2 007
	20.10.2007 0:00:00 123/35 10 000	10 2 007
	27.08.2008 13:57:29 123/35 10 000	1 2 008
۲	٢	×
🗋 Форма 🌱 Список Квитанци	4	
Для получения подсказки нажмите F1		CAP NUM

Рис. 6.12. Пример запроса с использованием нескольких логических операторов

При написании сложных логических конструкций следует обращать внимание на скобки, которые, как и в арифметических операциях, указывают порядок, в котором производится проверка выражения.

Агрегатные функции в запросах

Таким образом, мы на несложных примерах познакомились с технологией выборки информации из базы данных. Однако часто на практике требуется получить и разнообразную сводную информацию. В частности, по данным регистра квитанции управленческому персоналу могут понадобиться ответы на следующие вопросы:

- Какова итоговая сумма оплаты, произведенная каждым учащимся?
- Какова средняя сумма оплаты учащимися?
- Какова наибольшая сумма оплаты по каждому учащемуся?

Для реализации подобных запросов следует воспользоваться агрегатными функциями и группировками.

Примечание

В язык запросов включены агрегатные функции, которые используются при группировке результатов запросов, а также при подсчете итогов.

Любая агрегатная функция принимает в качестве аргумента какой-либо столбец, а возвращает единственное значение. В табл. 6.2 перечислены имеющиеся в системе 1С:Предприятие 8.1 агрегатные функции.

Результаты всех предыдущих запросов мы демонстрировали с помощью элемента управления *поле списка*, расположенного на форме обработки ИзвлечениеИнформации. В данной же ситуации более подходит табличный документ. Поэтому для демонстрации работы агрегатных функций, представленных в табл. 6.2, создадим новый объект — *отчет*, который назовем СводнаяВедомостьПоКвитанциям (рис. 6.13). Для табличного документа нам придется самостоятельно разработать макет.

Название	Описание
СУММА	Вычисляет сумму всех значений, содержащихся в указанном столбце. В качестве параметра функции можно указывать только поле, содержащее числовое значение
МАКСИМУМ	Находит наибольшее значение в указанном столбце
МИНИМУМ	Находит наименьшее значение в указанном столбце
СРЕДНЕЕ	Вычисляет среднее арифметическое значение по указанному столбцу
КОЛИЧЕСТВО	Подсчитывает количество значений, содержащихся в указанном столбце

Таблица 6.2. Агрегатные функции

В окне редактирования объекта конфигурации закладку Данные можно пропустить — реквизиты мы добавлять не будем, а на закладке Формы создадим новую форму произвольного вида и укажем, что она будет являться основной формой отчета (рис. 6.14). Для того чтобы формировать отчет при работе в режиме 1С:Предприятие, создадим на форме элемент управления — кнопку (рис. 6.15).

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🔲 🗵						
<u>Ф</u> айл <u>П</u> равка Ко	нфигурация Отлади	ка Администрирование <u>С</u> ервис <u>О</u> кна	1			
Справка						
I 🗋 🛋 🔛 🐰 I	` ```	₩ 🐂 🛛	- -			
🕸 Отчет Сводная	информацияПоК	витанциям .	_ 🗆 ×			
• Основные	14					
Данные	имя:	СводнаяИнформацияПоКвитанциям				
Формы	Синоним:	Сводная информация по квитанциям				
Макеты	Комментарий:					
Подсистемы						
Действия • <Назад Далее> Закрыть Справка						
Для получения подск	азки нажмите F1	CAP NUM	ru • //			

Рис. 6.13. Окно редактирования объекта конфигурации

🛃 Конфигуратор (у	чебная версия) - Конфигурация	
<u>∎ Ф</u> айл <u>П</u> равка Кон	фигурация Отладка Администрирование Сервис 🖸	кна Сп <u>р</u> авка
I 🗋 🚢 🔛 🐰 🖣	b 🛍 🧉 🎽 I 🕀 🎽 I	
🕸 Отчет Сводная	1нформацияПоКвитанциям	_ 🗆 ×
Основные	Основная форма отчета:	^
Данные	Форма	x Q
Макеты	Основная форма настроек:	
Подсистемы		x Q
Права	열 🛃 철 🕇 🖡 🛃	
Интерфейсы	🗗 📑 Формы	
прочее	Форма	
Действия	я • <Назад Далее> Закрыть	Справка
, Для получения подска	азки нажмите F1 CAP N	UM ru 🗸 🎢

Рис. 6.14. Закладка Формы окна редактирования отчета

Примечание

Отчетом мы воспользовались для того, чтобы использовать при разработке его подчиненный объект — форму.

Сам отчет будет формироваться с помощью табличного документа, и поэтому нам необходим макет табличного документа (рис. 6.16), с техническими действиями по созданию которого мы уже знакомы. На макете разместим три секции — Заголовок (название говорит само за себя), Шапка (для вывода поясняющей текстовой информации) и Строка (в которой будут выводиться значения переменных, определенных в тексте процедуры). Для всех ячеек с информацией установим обводку линиями (в этом случае следует обратиться к окну свойств ячеек), а для названий колонок установим еще и заливку цветом.



Рис. 6.15. Форма отчета в режиме 1С:Предприятие

🛵 Конфигуратор (учебная версия) - Конфигурация									
🙅 айл Правка Таблица Конфигурация Отладка Администрирование Сервис Окна Справка									
I □ 									- -
R1C1			▼ - =	=					_
🛕 Отчет С	Сводна	аяИнф	ормацияПоКвита	анциям: Макет				_ 🗆	×
		1	2	3	4	5	6	7	^
Заголовок	1								
	2		Отчет по к	витанциям	A				
	3								-
Шапка	4		Договор	Сумма	Максимум	Минимум	Среднее	Количество	
Строка	5		<Договор>	<Сумма>	<Максимум	<Минимум:	<Среднее>	<Количество>	
	6								
	7								
	8								÷
	9								ł
Для получени	ия подс	казки н	нажмите F1					CAP NUM ru*	111

Рис. 6.16. Макет отчета по квитанциям

Теперь перейдем к программной процедуре, где воспользуемся встречавшейся в *главе 1* конструкцией языка запросов — сгруппировать по. При ее использовании можно не просто выбирать записи из таблицы, а еще и группировать их определенным образом.

Примечание

Под словом *"сгруппировать"* подразумевается компоновка информации по группировочным полям и вычисление агрегатных функций по каждой группе.

Процедура для получения итогового отчета по оплаченным квитанциям приведена в листинге 6.9. В данном случае при вычислении сводной информации производится группировка по договорам (т. е. по учащимся). Наша задача в данном примере заключается в демонстрации работы агрегатных функций, приведенных в табл. 6.2. Непосредственно отчет формируется при нажатии кнопки Сформировать отчет.

Листинг 6.9. Процедура формирования отчета по квитанциям

```
Процедура СформироватьОтчетНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ Договор КАК Договор,
 СУММА (СуммаПоКвитанции) КАК Сумма,
    МАКСИМУМ (СуммаПоКвитанции) КАК Максимум,
 МИНИМУМ (СуммаПоКвитанции) КАК Минимум,
    СРЕДНЕЕ (СуммаПоКвитанции) КАК Среднее,
    КОЛИЧЕСТВО (*) КАК Количество
 L
    ИЗ
 РегистрСведений.Квитанции
    ГДЕ Оплачено=Истина
 | СГРУППИРОВАТЬ ПО Договор";
Выборка = Запрос.Выполнить ().Выбрать ();
ТабДок = Новый ТабличныйДокумент;
Макет = Отчеты. Сводная Информация ПоКвитанциям. Получить Макет ("Макет");
Область = Макет.ПолучитьОбласть ("Заголовок");
ТабДок.Вывести (Область);
Область = Макет.ПолучитьОбласть ("Шапка");
ТабДок.Вывести (Область);
Пока Выборка.Следующий() = Истина Цикл
    Область = Макет.ПолучитьОбласть ("Строка");
    Область.Параметры.Договор = Выборка.Договор;
    Область.Параметры.Сумма = Выборка.Сумма;
    Область.Параметры.Минимум = Выборка.Минимум;
    Область.Параметры.Максимум = Выборка.Максимум;
    Область.Параметры.Среднее = Выборка.Среднее;
    Область.Параметры.Количество = Выборка.Количество;
    ТабДок.Вывести (Область);
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Ложь;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать ();
КонецПроцедуры
```

Для внесенных в базу данных квитанций заполненный вариант отчета при работе в режиме 1С:Предприятие приведен (вместе со списком квитанций) на рис. 6.17.

🟪 1С:Предприятие (учебная версия) - Конфигурация 📃 🗖 🤰								×			
🛯 <u>Ф</u> айл <u>П</u> равка Операци	ни <u>С</u> ервис <u>О</u> кі	на Справка	1								
I 🗋 🚄 🔛 🐰 🖻 🕯	661	o et las			-	₩ *		1	M	M+ M- 🎇	š. –
🗼 Таблица										_ [] [x
											^
Отчет по к	витанция	м									
l	•									_	
Договор	Сумма	Максим	іум Мі	инимум	Средн	iee	Кол	ичесте	0		
123/35	20 000	10 0	000	10 000	1	0 0 0 0 0				2	
123/34	18 900	89	970	1 930		6 300				3	
		•				•					\sim
										>	
👖 Список Квитанции	M		-							_ U	×
Действия 👻 🔄 🚭	, 🔄 🖬 (++)	Щ ю П	[• 1 ₄	2							
Период 🗸	Договор С	умма по	Оплачен	о Месяц н	начала	Год нача	ла I	Месяц ок	онГ	од оконча И	~
10.08.2007 15:21:15	123/34	1 930	~		6	ź	2 007		6	2 007	
10.08.2007 17:46:07	123/34	8 000	~		9	2	2 007		10	2 007	
07.09.2007 0:00:00	123/34	8 970	~		11	2	2 007		7	2 007	
20.10.2007 0:00:00	123/35	10 000	~		10	2	2 007		12	2 007	
27.08.2008 13:57:29	123/35	10 000	~		1	2	2 008		5	2 008	
Форма	🕐 Слис	ок Квитанци	1.J	🗜 Tañ	пица						
Для получения подсказки	нажмите F1	онставиланци								CAP NUM	

Рис. 6.17. Отчет по квитанциям и состояние квитанций

Использование параметров в запросах

Во всех ранее рассмотренных запросах данной главы пользователю не требовалось устанавливать какие-либо параметры, а условия для отбора формировались непосредственно в тексте запроса. Если же посмотреть на любой из отчетов, разработанных специалистами фирмы "1С", то мы увидим, что параметры многочисленных условий пользователь может указывать в диалоге.

Для реализации подобной возможности запрос строится с использованием параметров. В качестве примера в листинге 6.10 приведена процедура, где от пользователя требуется ввод максимальной суммы для отбора интересующих квитанций. В этом случае в полученной выборке будут присутствовать все квитанции на сумму, не превышающую максимальную.

Саму процедуру расположим на форме обработки ИзвлечениеИнформации. Для добавления возможности ввода параметра нам потребуется поместить на форме поле ввода МаксимальнаяСумма (рис. 6.18), для которого установить числовой тип данных. Также необходимо скорректировать процедуру ОтобратьИнформациюНажатие, которая выполняется при нажатии кнопки Отобрать информацию. Обратите внимание, что в параметрическом запросе (листинг 6.10) параметр (в данном случае МаксимальнаяСумма) предваряется знаком амперсанда («).



Рис. 6.18. Размещение поля ввода для указания значения параметра в запросе

Листинг 6.10. Процедура с использованием параметра в запросе

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ Договор, СуммаПоКвитанции
| ИЗ РегистрСведений.Квитанции
| ГДЕ СуммаПоКвитанции <= &МаксимальнаяСумма";
Запрос.УстановитьПараметр ("МаксимальнаяСумма",
ЭлементыФормы.МаксимальнаяСумма.Значение);
Результат = Запрос.Выполнить ().Выбрать ();
СписокОтобранныхДанных.Очистить ();
Пока Результат.Следующий () > 0 Цикл
СписокОтобранныхДанных.Добавить (Строка (Результат.Договор) +
" "+Строка (Результат.СуммаПоКвитанции)+" руб.");
КонецЦикла;
КонецПроцедуры
```

Метод установитьПараметр () позволяет определить значение параметра для запроса. При этом первый параметр метода представляет собой название параметра запроса, а второй — значение, присваиваемое этому параметру. В данном случае второй параметр метода берется равным значению поля ввода МаксимальнаяСумма (которое пользователь заполняет перед выполнением запроса). На рис. 6.19 приведена информация, отобранная в результате выполнения запроса при работе в режиме 1С:Предприятие.

Примечание

Для установки параметра запроса используется метод УстановитьПараметр (ИмяПараметра, ЗначениеПараметра). Для параметра ИмяПараметра тип данных — строка, а параметр ЗначениеПараметра имеет произвольный тип данных. С помощью этого метода можно передавать переменные в запрос.



Рис. 6.19. Результат выполнения параметрического запроса в режиме работы 1С:Предприятие

Ключевое слово МЕЖДУ

В языке запросов имеется вспомогательное ключевое слово между для задания интервалов. Например, в запросе, приведенном в листинге 6.11, необходимо перед его выполнением установить значения параметров дата1 и дата2. Эти значения вводятся пользователем в поля ввода на форме (рис. 6.20).

📩 Конфигуратор (учебная версия) - Конфигурация
<u>∲</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна
Справка
I 🗋 🖬 🔏 👛 🧔 🐥 📜 🔽 🔽 🔽
🚟 Обработка Рис6_21: Форма 📃 🗆 🗙
Форма
Отобрать информацию Дата1: 🔳
Список отобранных данных: Дата2: 🖽
Для получения подсказки наж САР NUM ги 🕶 📩 0, 0 🕫 398 x 192 🥢

Рис. 6.20. Форма обработки в окне редактирования объекта конфигурации

В предыдущем примере на форме присутствовало поле ввода МаксимальнаяСумма. Его можно не убирать, а лишь отключить видимость (это одно из свойств) данного элемента управления. В этом случае в режиме 1С:Предприятие этого поля мы не увидим.

Заметим, что в конструкции рассматриваемого запроса (листинг 6.11) используется параметр период (более конкретно — измерение), который отражает дату и время внесения информации.

```
Листинг 6.11. Процедура с использованием ключевого слова между
```

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Договор,
| СуммаПоКвитанции КАК Сумма
| ИЗ РегистрСведений.Квитанции
| ГДЕ РегистрСведений.Квитанции.Период МЕЖДУ &Дата1 И &Дата2";
Запрос.УстановитьПараметр ("Дата1", ЭлементыФормы.Дата1.Значение);
Запрос.УстановитьПараметр ("Дата2", ЭлементыФормы.Дата2.Значение);
Результат = Запрос.Выполнить ().Выбрать ();
СписокОтобранныхДанных.Очистить ();
Пока Результат.Следующий () > 0 Цикл
СписокОтобранныхДанных.Добавить (Строка (Результат.Договор) +
" " + Строка (Результат.Сумма) + " руб.");
КонецЦикла;
КонецПроцедуры
```

Перейдем в режим 1С:Предприятие и перед выполнением запроса введем необходимый интервал дат. Выборка результата запроса вместе с содержанием регистра Квитанции показана на рис. 6.21.

🏪 1С:Предприятие (учебная версия) - Конфигурация	
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
I 🗋 🖬 🕺 🖓 🛍 📾 🔯 🖕 🖕 👘	💌 🖗 🖗 🖻 🖳 🦕 🛛 🎆 🦳 M M+ M- 🎇 🗸
🗋 Форма 📃 🗆 Х	f Список Квитанции 💦 _ 🗆 🗙
Отобрать информацию Дата1: 02.08.2007 🔳	Действия • 🔄 🔄 式 🎽 🔜 (++) 🏦 🛅 👘 • 7 🔆 🐥
Список отобранных данных: Дата2: 16.10.2007 🗐 123/34 1 930 руб. 123/34 8 900 руб. 123/34 8 970 руб.	Image: Second state of the second
Форма Список Квитанции Для получения подсказки нажмите F1	CAP NUM

Рис. 6.21. Выполнение запроса, использующего ключевое слово МЕЖДУ

В языке запросов существует ключевое слово имеющие, позволяющее накладывать условия на значения агрегатных функций. В других конструкциях языка запросов использовать агрегатные функции в условиях нельзя. Однако и в условии отбора со словом имеющие можно использовать агрегатные функции только для полей, по которым осуществляется группировка.

В листинге 6.12 приведен пример процедуры с запросом, в котором используется конструкция имеющие. Цель рассматриваемого запроса — отобрать договоры, для которых итоговая сумма по оплаченным квитанциям не превышает значения МаксимальнаяСумма (его мы вводим в качестве параметра с клавиатуры). На рис. 6.22 показан фрагмент регистра квитанции и окно формы с выборкой результата запроса.

陽 1С:Предприятие (учебная версия) - Конфигурация	
🛛 🙅 айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
I 🗋 🛁 🖌 🖻 🖷 🥌 🖉 Þ 🕹 🗰	💌 🖗 🐐 🍟 🎆 🛅 🧝 м м+ м- 🎇 🖕
🕒 Форма 💶 🗆 🗸	🥤 Список Квитанции 💦 💶 🗙
	Действия 🗸 🔄 🔄 式 🎽 🔜 💮 🏹 🏹 🎽
Отобрать информацию Максимальная сумма:	 Договор Сумма по квОплачено № 11123/34 1 930 ✓
Список отобранных данных: 25 000	□ 1 123/34 8 000 ✓ □ 0 122/24 8 970 4
123/35 20 000 руб.	Image: Signal and Si
123/34 18 900 руб.	<u> </u>
<	✓ 2 18//// 36 UUU ✓
🗋 Форма 🌈 Список Квитанции	
Для получения подсказки нажмите F1	CAP NUM ///

Рис. 6.22. Результат выполнения запроса с использованием ключевого слова ИМЕЮЩИЕ

Листинг 6.12. Запрос с использованием ключевого слова имеющие

```
Процедура Кнопка1Нажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ Договор,
| СУММА (СуммаПоКвитанции) КАК Сумма
| ИЗ РегистрСведений.Квитанции
| ГДЕ Оплачено=Истина
| СГРУППИРОВАТЬ ПО Договор
| ИМЕЮЩИЕ
| СУММА (СуммаПоКвитанции) <= &МаксимальнаяСумма";
Запрос.УстановитьПараметр ("МаксимальнаяСумма",
ЭлементыФормы.МаксимальнаяСумма.Значение);
Результат = Запрос.Выполнить ().Выбрать ();
СписокОтобранныхДанных.Очистить ();
```

```
Пока Результат.Следующий() > 0 Цикл
СписокОтобранныхДанных.Добавить(Строка(Результат.Договор) +
" " + Строка(Результат.Сумма) + " руб.");
КонецЦикла;
КонецПроцедуры
```

Функция расчета итогов в запросах

Формирование итоговой информации является одной из наиболее востребованных функций любой информационной системы. Язык запросов системы 1С:Предприятие 8.1 включает в себя очень мощные возможности расчета итогов. В этом случае пользователь может включить в результат запроса дополнительные строки, содержащие общие и промежуточные итоги по заданным полям и группировкам.

В листинге 6.13 приведена процедура, использующая запрос, который позволяет получить общие итоги по выписанным квитанциям. Как видно из выборки результата запроса (рис. 6.23), в этом случае в сформированном списке появляется дополнительная строка с итоговой суммой.

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🗆 🗙
🗄 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
I 🗋 💶 X ங 🛍 🛎 🔍 💙 I 🎹 🗎 🧟 M M+ M- 🎘 🧅
🔓 Форма 💶 🗆 🖌
Отобрать информацию
Список отобранных данных:
38 900 руб.
10.08.2007 15:21:15 123/34 1 930 py6.
10.08.2007.0.000 123734 8.370 pg6.
27.08.2008 13:57:29 123/35 10 000 py6.
20.10.2007 0:00:00 123/35 10 000 py6.
Форма
Для получения подсказки нажмите F1 САР NUM

Рис. 6.23. Использование функции расчета итогов в запросе

Листинг 6.13. Процедура, включающая в себя общие итоги по квитанциям

```
Процедура ОтобратьИнформациюНажатие(Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Договор,
| СуммаПоКвитанции КАК Сумма, Период
```

```
| ИЗ РегистрСведений.Квитанции
| ИТОГИ СУММА(Сумма) ПО Общие";
Результат = Запрос.Выполнить().Выбрать();
СписокОтобранныхДанных.Очистить();
Пока Результат.Следующий() > 0 Цикл
СписокОтобранныхДанных.Добавить(Строка(Результат.Период) +
" " + Строка(Результат.Договор) +
" " + Строка(Результат.Сумма) + " руб.");
КонецЦикла;
КонецПроцедуры
```

При обходе результата запроса итоговые строки можно отличить от обычных с помощью метода ТипЗаписи(). В листинге 6.14 приведена процедура, в которой итоговая строка по результату данного запроса выводится в отдельный элемент (надпись) на форме (рис. 6.24).

🏪 1С:Предприятие (учебная версия) - Конфигурация	a 📃	×
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка		
I 🗋 🔟 📓 🔏 📾 🥌 🖾 🗖 🗠 🖿	- 👻 📰 🧮 🧟 M M+ M- 💥	\$ _
🗅 Форма	👖 Список Квитанции	
	Действия • 👌 🔄 🔩 🧏 🔜 🙌 🏹 🌆 🖷 • 🌾	Q
Отобрать информацию	Период 🔹 Договор Сумма по Оплачен	0
Список отобранных данных:	Image:	_
10.08.2007 15:21:15 123/34 1 930 руб.	Image: Construction Const	
10.08.2007 17:46:07 123/34 8 000 py6.	☑ 20.10.2007 0:00:00 123/35 10 000 ✓	
07.09.2007 0:00:00 123/34 8 970 pg6. 20.10.2007 0:00:00 123/35 10 000 pg6. 27.08.2008 13:57:29 123/35 10 000 pg6.	27.08.2008 13:57:29 123/35 10 000 ✔	_
Итог 38 900		
🖹 Форма 🌈 Список Квитанции		
Для получения подсказки нажмите F1	CAP NUM	1

Рис. 6.24. Использование метода ТипЗаписи () при расчете итогов в запросе

```
Листинг 6.14. Пример процедуры с использованием метода ТипЗаписи ()

Процедура ОтобратьИнформациюНажатие (Элемент)

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ Договор,

| СуммаПоКвитанции КАК Сумма, Период

| ИЗ РегистрСведений.Квитанции

| ГДЕ Оплачено=Истина

| ИТОГИ СУММА (Сумма) ПО Общие";

Рез = Запрос.Выполнить ().Выбрать ();
```

```
СписокОтобранныхДанных.Очистить();
Пока Рез.Следующий() > 0 Цикл
Если Рез.ТипЗаписи() = ТипЗаписиЗапроса.ОбщийИтог Тогда
ЭлементыФормы.ИтогПоКвитанциям.Значение = Строка(Рез.Сумма);
Иначе
СписокОтобранныхДанных.Добавить(Строка(Рез.Период) +
" " + Строка(Рез.Договор) +
" " + Строка(Рез.Сумма) + " руб.");
КонецЕсли;
КонецЦикла;
КонецПроцедуры
```

Несложно получить итоги и по группировкам. В этом случае вычисляются значения агрегатных функций по выборкам с одинаковыми значениями полей, по которым производится группировка. В листинге 6.15 приведена процедура, в которой приводятся итоги оплаченных квитанций по договорам, а выполнение запроса в режиме работы 1С:Предприятие продемонстрировано на рис. 6.25.

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🔲 🗙
🗄 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
I 🗋 🛁 🖌 🦄 🛍 🥌 💽 🎽 I 🎹 🗐 🧝 🛛 м м+ м- 🎘 🧅
🔓 Форма 💶 🗆 🕹
?
Отобрать информацию
Список отобранных данных:
123/34 18 900 руб.
10.08.2007 15:21:15 123/34 1 930 руб.
10.08.2007 17:46:07 123/34 8 000 руб.
07.09.2007 0:00:00 123/34 8 970 руб.
123/35 20 000 руб.
20.10.2007 0:00:00 123/35 10 000 руб.
27.08.2008 13:57:29 123/35 10 000 руб.
P (Denvis
Для получения подсказки нажмите F1 САР NUM

Рис. 6.25. Отчет с включением итогов по группировкам

Листинг 6.15. Процедура, включающая итоги по группировкам

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ Договор,
| СуммаПоКвитанции КАК Сумма, Период
| ИЗ РегистрСведений.Квитанции
| ГДЕ Оплачено=Истина
| ИТОГИ СУММА (Сумма) ПО Договор";
```

```
Результат = Запрос.Выполнить().Выбрать();
СписокОтобранных.Анных.Очистить();
Пока Результат.Следующий() > 0 Цикл
СписокОтобранныхДанных.Добавить(Строка(Результат.Период) +
" " + Строка(Результат.Договор) +
" " + Строка(Результат.Сумма) + " руб.");
КонецЦикла;
КонецПроцедуры
```

Запрос с итогами по группировкам отличается от обычной группировки (свертки) с использованием предложения сгруппировать по с агрегатными функциями. В последнем случае в результат запроса не включаются исходные записи, а остаются только итоговые строки.

Объединение результатов нескольких запросов

В языке запросов имеется возможность объединять результаты нескольких запросов, при этом записи, полученные с помощью каждого из объединяемых запросов, будут собраны в один результат. При объединении каждый запрос собирает данные независимо, а такие операции, как упорядочивание результатов и расчет итогов, выполняются уже над результатом объединения запросов.

Поля результата запроса будут называться так, как описано в списке полей выборки первого из объединяемых запросов. Поля выборки остальных запросов сопоставляются с полями результата в соответствии с порядком их следования в списке полей выборки. Объединяемые запросы должны иметь одинаковое количество полей в списке полей выборки.

Рассмотрим ситуацию, когда кроме оплаты наличным путем через квитанции возможно еще и безналичное перечисление. В этом случае нам понадобится создать еще один регистр сведений — БезналичнаяОплата. Это будет независимый периодический (с периодом — одна секунда) регистр сведений. Заполнение основной закладки Данные в окне редактирования объекта конфигурации приведено на рис. 6.26.

Теперь мы должны построить запрос для извлечения оплаченных квитанций и банковских перечислений в задаваемом пользователем интервале дат (листинг 6.16). При объединении запросов ключевым является слово объединить. Перед ним должен быть расположен один запрос, а после него — другой.

Листинг 6.16. Пример процедуры с объединением запросов

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Договор,
| СуммаПоКвитанции КАК Сумма,Период
| ИЗ РегистрСведений.Квитанции
| ГДЕ (Период МЕЖДУ &Дата1 И &Дата2 ) И Оплачено=Истина
| ОБЪЕДИНИТЬ
| ВЫБРАТЬ Договор, ПеречисленнаяСумма КАК Сумма, Период
```

```
ИЗ РегистрСведений.БезналичнаяОплата
ГДЕ Период МЕЖДУ &Дата1 И &Дата2";
Запрос.УстановитьПараметр("Дата1", ЭлементыФормы.Дата1.Значение);
Результат = Запрос.Выполнить().Выбрать();
СписокОтобранныхДанных.Очистить();
Пока Результат.Следующий() > 0 Цикл
СписокОтобранныхДанных.Добавить(Строка(Результат.Период) + " " + Строка(Результат.Договор) + " " + Строка(Результат.Сумма) + " руб.");
КонецЦикла;
```

КонецПроцедуры



Рис. 6.26. Окно редактирования регистра Безналичная Оплата

🏪 1С:Предприятие (учебная версия) - Конфигурация		_ 🗆 🗙
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка		
ID → ■ X № ® <i>●</i> © > ⊄ x	🗸 🕅 🕅 🖻 🖳 🚛 📰 🛣 M M+ M- 🎘 🗸	
Рорма _	🦸 Список Безналичная оплата	_ 🗆 ×
	Действия - 🔄 🔄 🛃 🍇 🔛 (+) 🏹 🐚 🖷 - 🌾 🔃 🕐 🦷	
Отобрать информацию	Период • Договор Перечисленная сумма 2017.08.2007.0:00:00 127/77	1 000
Список отобранных данных:		
17.08.2007 0:00:00 127/77 1 000 py6.		
20.10.2007 0:00:00 123/35 10 000 py6.		~
10.08.2007 15:21:15 123/34 1 930 py6.	Действия 🕶 🔄 🛬 🛬 🔜 😝 👖 🕐 👖 🖭 🗤 🦷 🗘 🕐	
07.09.2007.07.00.00.123734-8.000.pg0.	Период 🔹 Посовор Симиз ро Орденено Месян ненас	Da For A
	10.08.2007 15:21:15 123/34 1 930 ✓	6
	☑ 10.08.2007 17:46:07 123/34 8 000 ✔	9
	🖂 07.09.2007 0:00:00 123/34 8 970 🗸	11
	📨 20.10.2007 0:00:00 123/35 10 000 🗸	10
Дата1: 01.08.2007 🔳 Дата2: 21.08.2008 🔳	☑ 27.08.2008 13:57:29 123/35 10 000 ✔	1
	٢	>
🛉 Форма 🌈 Список Безналичная оплата 🁔 С	писок Квитанции	
Для получения подсказки нажмите F1	CAP	NUM //

Рис. 6.27. Результат выполнения объединенного запроса

На рис. 6.27 показана информация, содержащаяся в регистрах квитанции и БезналичнаяОплата, а также результат выполнения запроса.

Более сложные конструкции языка запросов

В этом разделе мы познакомимся с рядом новых конструкций для построения запросов. Объединяющим моментом всех приведенных далее примеров является то, что они связаны с разработкой из *главы 1*. В целом для формирования запросов нам будет достаточно ранее созданных объектов конфигурации, однако для визуализации результатов потребуются также несколько интерфейсных форм с необходимыми элементами управления.

Итак, если вы выполняете описанные в книге действия на компьютере, то лучше всего в режиме конфигуратора открыть созданную информационную базу, разработанную в *главе 1*.

Примечание

На компакт-диске, прилагаемом к книге, все информационные базы, упоминаемые в тексте, размещены в отдельных папках. В конце книги приводится подробное содержание компакт-диска.

Отбор специалистов по указанной должности

Итак, приступим к первому запросу, связанному с разработанной конфигурацией из главы 1. В ней мы создали объект конфигурации — справочник Специалисты, в котором содержится список преподавателей нашего учебного центра. В этом справочнике имеется табличная часть МестаРаботы, в которой перечислены организации, где дополнительно работает каждый сотрудник нашего центра. Вместе с указанием организации в каждой строке табличной части справочника определяется еще и должность специалиста в этой организации.

Будем считать, что цель запроса заключается в том, чтобы отобрать специалистов (с указанием места работы), имеющих выбранную в элементе интерфейса (поле ввода) должность. Подобная (или аналогичная) задача вполне может возникнуть и в другой области деятельности.

Как и ранее, для визуализации результатов запроса мы создадим обработку с экранной формой произвольного вида. На форме расположим три элемента управления — поле ввода, кнопку и поле списка (рис. 6.28). Далее мы напишем программный код процедуры. Поэтому для исключения противоречий с листингом 6.18 для поля ввода установим имя должность, а тип значения выберем СправочникСсылка.Должность (это позволит нам с помощью данного элемента управления открыть для выбора справочник с уже внесенными должностями). Для поля списка установим имя Результатыютбора. Что касается кнопки, то ее имя можно установить в соответствии с рис. 6.28 — ОтобратьИнформацию.

📩 Конфигуратор (учебная версия) - Конфигурация	- 🗆 ×
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администриров	ание
I <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
	~
🚟 Обработка Рис6_29: Форма	_ 🗆 ×
Форма	
Должность:	
Отобрать информацию	
Результаты отбора:	
Для получения под] САР NUM ru + 1 + 0, 0 ↓ 386 × 3	277 //

Рис. 6.28. Форма обработки для визуализации результата запроса

Действия пользователя заключаются в предварительном выборе в поле ввода интересующей должности. После этого нажатие кнопки **Отобрать информацию** приведет к заполнению списка интересующей нас информацией о сотрудниках.

В листинге 6.17 показан текст процедуры, которая выполняется при нажатии на рассматриваемую кнопку. Здесь при осуществлении запроса к табличной части справочника обращение к обычным (вне таблицы) реквизитам справочника производится через поле Ссылка. Хотя запрос обращается к табличной части местаРаботы справочника Специалисты, основная информация о специалисте заключена в реквизите Наименование, который располагается вне табличной части. И для получения фамилии специалиста в тексте запроса используется конструкция Ссылка.Наименование.

Листинг 6.17. Запрос к табличной части справочника Специалисты

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Ссылка.Наименование
| КАК Наименование,
| Организация, Должность
| ИЗ Справочник.Специалисты.МестаРаботы
| ГДЕ Должность = &Должность";
Запрос.УстановитьПараметр ("Должность", ЭлементыФормы.Должность.Значение);
Результат = Запрос.Выполнить ().Выбрать ();
```

```
РезультатыОтбора.Очистить();
Пока Результат.Следующий() > 0 Цикл
РезультатыОтбора.Добавить(Строка(Результат.Наименование) +
" " + Строка(Результат.Организация) +
" " + Строка(Результат.Должность));
КонецЦикла;
КонецПроцедуры
```

В запросе, приведенном в листинге 6.17, в выборку отобранных данных один и тот же специалист может попасть два и более раз, если он работает в нескольких организациях на одинаковых должностях. Именно такая ситуация отражена на рис. 6.29: Иван Александрович Петров работает преподавателем в двух организациях (МТКИ и НМСИ).

陽 1С:Предприятие (учебная версия) - Конфигурация	<u>- 0 ×</u>
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
🗋 🔟 X 🍡 📾 🥌 🖄 🖕 🗶 🏧 🧮 🧝	M M+ M- 🎇 🖵
Рорма	_ 🗆 ×
Должность: Преподаватель x Отобрать информацию Результаты отбора:	
Петров Иван Александрович МТКИ Преподаватель Петров Иван Александрович НМСИ Преподаватель	
🔓 Форма	
Для получения подсказки нажмите F1	CAP NUM

Рис. 6.29. Результат отбора специалистов с указанной должностью

В листинге 6.18 приведена процедура, позволяющая в запросе отобрать только различных специалистов, имеющих указанную пользователем должность. Результат выполнения запроса для имеющихся в информационной базе данных приведен на рис. 6.30.

```
Листинг 6.18. Запрос с отбором различных специалистов с указанной должностью
```

```
Процедура ОтобратьИнформациюНажатие (Элемент)
```

```
Запрос = Новый Запрос;
```

```
Запрос.Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ
```

- | Ссылка.Наименование
- | КАК Наименование,
- | Должность

```
ИЗ Справочник.Специалисты.МестаРаботы
| ГДЕ Должность = &Должность";
Запрос.УстановитьПараметр("Должность", ЭлементыФормы.Должность.Значение);
Результат = Запрос.Выполнить().Выбрать();
РезультатыОтбора.Очистить();
Пока Результат.Следующий() > 0 Цикл
РезультатыОтбора.Добавить(Строка(Результат.Наименование));
КонецЦикла;
```

КонецПроцедуры



Рис. 6.30. Результат отбора специалистов с указанной должностью без повторений

Отбор курсов преподавателя

При работе с рассматриваемой информационной базой нам может потребоваться получить в выборке курсы с указанием их стоимости, которые ведет конкретный преподаватель (фамилия преподавателя аналогично должности должна выбираться из списка значений). Используемая ранее форма в данной ситуации для отображения результатов не совсем подходит. В этом случае можно пойти двумя путями — создать новую форму или немного подкорректировать имеющуюся (заменить некоторые элементы управления). Мы выберем второй путь, и у уже созданных элементов сделаем следующие изменения:

- поменяем имя поля ввода на значение Преподаватель;
- установим тип значения поля ввода СправочникСсылка. Специалисты.

После этого для реакции на нажатие кнопки **Отобрать информацию** необходимо на закладке **Модуль** разместить процедуру (заменить предыдущую), приведенную в листинге 6.19. Результат выполнения запроса в режиме 1С:Предприятие продемонстрирован на рис. 6.31.



Рис. 6.31. Отбор дисциплин указанного преподавателя

Листинг 6.19. Запрос для отбора курсов конкретного преподавателя

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Наименование,
| Стоимость, Преподаватель
ИЗ Справочник.Курсы
| ГДЕ Преподаватель = &Преподаватель";
Запрос.УстановитьПараметр ("Преподаватель",
       ЭлементыФормы.Преподаватель.Значение);
Результат = Запрос.Выполнить ().Выбрать ();
РезультатыОтбора.Очистить ();
Пока Результат.Следующий() > 0 Цикл
    РезультатыОтбора.Добавить (Строка (Результат.Наименование) +
            " + Строка (Результат.Преподаватель) +
            " + Строка (Результат. Стоимость));
КонецЦикла;
КонецПроцедуры
```

Соединение таблиц при организации запроса

До настоящего времени во всех использованных запросах выборка извлекалась из одной таблицы. Важной способностью языка запросов системы 1С является возможность обращения сразу к нескольким таблицам. При этом их можно соединять определенным образом. Например, при работе может потребоваться выбрать названия всех курсов с указанием стоимости, преподавателя и рейтинга, который имеет данный преподаватель.

В конструкции подобного запроса (листинг 6.20) используется ключевое слово соединение, позволяющее отобрать из справочников преподаватели и Курсы только те записи, в которых преподаватели в обоих справочниках совпадают (рис. 6.32).

Листинг 6.20. Запрос с использованием соединения двух таблиц

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ Спр1.Наименование КАК Курс,
| Спр1.Преподаватель КАК Преп,
| Спр1.Стоимость КАК Ст, Спр2.Рейтинг КАК Рейт
| ИЗ Справочник.Курсы КАК Спр1
| СОЕДИНЕНИЕ
| Справочник.Специалисты КАК Спр2
| ПО Спр1.Преподаватель=Спр2.Ссылка";
Результат = Запрос.Выполнить ().Выбрать ();
РезультатыОтбора.Очистить ();
Пока Результат.Следующий () > 0 Цикл
РезультатыОтбора.Добавить (Строка (Результат.Курс) +
" " + Строка (Результат.Преп) + " "
+ Строка (Результат.Ст) + "руб. Рейтинг: "
```

+ Строка (Результат. Рейт));

КонецЦикла;

КонецПроцедуры



Рис. 6.32. Результат соединения двух таблиц в запросе

Левое внешнее соединение

При выполнении предыдущего запроса в результат отбора попала часть интересующей нас сводной информации. Недостатком данного метода отбора информации является то, что курсы, для которых не указан преподаватель, в результате выборки не отражены. Так, на рис. 6.33 приведена ситуация, когда в одном из курсов (Web-дизайн) не указан преподаватель и, соответственно, в отобранной информации мы данный курс не видим. Соединение с таким поведением относится к классу *внутренних*.

🏪 1С:Предприятие (учеб	іная версия)	- Конфигурация			
🛛 <u>Ф</u> айл <u>П</u> равка Операции	<u>С</u> ервис <u>О</u> кн	ю Справка			
I 🗋 🚄 🝙 🐰 🖻 🖷	6	e 🖻 🦛	» •	I 📰 📄 🧟	м м+ м- 🕅 🎘 🖕
📄 Форма					_ 🗆 ×
Отобр	ать информац	ию			
Результаты отбора:					
Информатика (Excel Базы данных Петро РоwerPoint Петров Элект. почта Никол) Петров Ива ов Иван Алекс Иван Алексан, паева Ирина Вл	н Александрович андрович 2 200ру дрович 1 200руб. падимировна 1 1(1800руб. Рей јб. Рейтинг:7 Рейтинг:7 00руб. Рейтин	ітинг: 7 r: 5	
🗐 Справочник Курсы	×				_ 🗆 ×
Действия 🕶 🚊 🐳	🔁 🔤 🕸		♀ ?		
Код Названия 🔻 ,	Дата начала	Продолжитель	Стоимость	Преподаватель	^
🗕 3 PowerPoint	11.08.2007	3	1 200	Петров Иван Але	жсандрович
5 Web-дизайн	21.08.2007	5	2 000		
💻 2 Базы данных	11.08.2007	5	2 200	Петров Иван Але	жсандрович
ПИнформатик	07.08.2007	5	1 800	Петров Иван Але	жсандрович
- 4јзлект, почта	11.08.2007	2	1100	пиколаева Ирин	а владимир
🖻 Форма	🗐 Спрає	зочник Курсы			
Для получения подсказки на	ажмите F1				CAP NUM

Рис. 6.33. Результат внутреннего соединения при построении запроса

В языке запросов системы 1С:Предприятие 8.1 существует возможность еще и *внешних* соединений, которые могут быть левыми, правыми и полными.

Конструкция левое [внешнее] соединение означает, что в результат запроса надо включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Но, в отличие от внутреннего соединения, в результат запроса попадают и записи из первого источника (указанного *слева* от слова соединение), для которых не найдено соответствующих условию записей из второго источника. Таким образом, в результат запроса будут включены все записи из первого источника, и они будут соединены с записями из второго источника при выполнении указанного условия. Строки результата запроса, для которых не найдено соответствующих условию записей из второго источника, будут содержать специальное значение NULL в полях, формируемых на основании записей из этого источника.

Примечание

Значения NULL являются специальными маркерами, обозначающими неуказанные (отсутствующие) значения или значения, не имеющие смысла.

Пусть, например, необходимо выбрать названия всех курсов с указанием стоимости, преподавателя и рейтинга, который имеет данный преподаватель, однако при этом в результирующую таблицу необходимо включить курсы, для которых не указан преподаватель. Это реализуется с помощью конструкции левое [внешнее] соединение (листинг 6.21). Иллюстрация результата отбора (рис. 6.34) показывает, что в этом случае (в отличие от внутреннего соединения) в полученном списке присутствуют и курсы с неуказанными преподавателями.



Рис. 6.34. Результат выполнения запроса с использованием левого внешнего соединения

Листинг 6.21. Пример запроса с использованием левого внешнего соединения

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Спр1.Наименование КАК Курс,
| Спр1.Преподаватель КАК Преп,
| Спр1.Стоимость КАК Ст, Спр2.Рейтинг КАК Рейт
| ИЗ Справочник.Курсы КАК Спр1
| ЛЕВОЕ СОЕДИНЕНИЕ
| Справочник.Специалисты КАК Спр2
| ПО Спр1.Преподаватель=Спр2.Ссылка";
Результат = Запрос.Выполнить ().Выбрать ();
РезультатыОтбора.Очистить ();
Пока Результат.Следующий() > 0 Цикл
    РезультатыОтбора.Добавить (Строка (Результат.Курс) +
            " + Строка (Результат.Преп) + " " + Строка (Результат.Ст) +
        "руб. Рейтинг: " + Строка (Результат. Рейт));
КонецЦикла;
КонецПроцедуры
```

Правое внешнее соединение

Конструкция правое [внешнее] соединение означает, что в результат запроса надо включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результат запроса надо включить еще и записи из второго источника (указанного справа от слова соединение), для которых не найдено соответствующих условию записей из первого источника.

Таким образом, в результат запроса будут включены все записи из второго источника; они будут соединены с записями из первого источника при выполнении указанного условия. Строки результата запроса, для которых не найдено соответствующих условию записей из первого источника, будут содержать значение NULL в полях, формируемых на основании записей из этого источника. Правое внешнее соединение полностью аналогично левому, за исключением того, что таблицы поменялись местами.

Пусть, например, необходимо выбрать названия всех курсов с указанием их стоимости, преподавателя и рейтинга, который имеет данный преподаватель, однако при этом в результирующую таблицу необходимо включить и преподавателей, для которых не указан курс (еще не определена дисциплина, которую он будет вести). Это реализуется с помощью конструкции правое [внешнее] соединение (листинг 6.22). Иллюстрация результата отбора (рис. 6.35) показывает в этом случае, что в отличие от внутреннего соединения в полученном списке данных присутствуют и курсы с неуказанными преподавателями.

Листинг 6.22. Запрос с использованием правого внешнего соединения

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Спр1.Наименование КАК Курс,
| Спр2.Наименование КАК Преп,
| Спр1.Ссылка КАК СпрСсылка1, Спр2.Ссылка КАК СпрСсылка2,
| Спр1.Стоимость КАК Ст, Спр2.Рейтинг КАК Рейт
| ИЗ Справочник.Курсы КАК Спр1
| ПРАВОЕ ВНЕШНЕЕ СОЕЛИНЕНИЕ
| Справочник.Специалисты КАК Спр2
| ПО Спр1.Преподаватель=Спр2.Ссылка";
Результат = Запрос.Выполнить ().Выбрать ();
РезультатыОтбора.Очистить ();
Пока Результат.Следующий() > 0 Цикл
    Если Результат.СпрСсылка2.ЭтоГруппа = Ложь Тогда
        РезультатыОтбора. Добавить (Строка (Результат. Курс) +
            "
                " + Строка (Результат.Преп) + " " + Строка (Результат.Ст) +
            "
                    Рейтинг: " + Строка (Результат. Рейт));
    КонецЕсли;
КонецЦикла;
КонецПроцедуры
```

Для исключения попадания в результаты отбора названий категорий специалистов (техническое и гуманитарное направления) в процедуре мы воспользовались свойством ЭтоГруппа.

陽 1С:Предприятие (учебная версия) - Конфигурация	
🗄 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
I 🗋 🖬 🐰 🖻 📾 🥌 🗹 🗛 👘 🦉 I 🏧 🛅 🧝	M M+ M- 🎇 🖵
🔓 Форма	_ 🗆 ×
Отобрать информацию	
Результаты отбора:	
Информатика (Excel) Петров Иван Александрович 1 800 Рейтинг: 7 Базы данных Петров Иван Александрович 2 200 Рейтинг: 7 РоwerPoint Петров Иван Александрович 1 200 Рейтинг: 7 Иванова Ольга Петровна Рейтинг: 5	
Элект, почта Николаева Ирина Владимировна 1100 Рейтинг; 5	
Форма	
Для получения подсказки нажмите F1	CAP NUM //

Рис. 6.35. Пример выполнения запроса с правым внешним соединением

Полное внешнее соединение

Конструкция полное [внешнее] соединение означает, что в результат запроса надо включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результат запроса надо включить также еще и те записи из обоих источников, для которых не найдено соответствий.

Таким образом, в результат запроса будут включены все записи из обоих источников, и они будут соединены друг с другом при выполнении указанного условия. Листинг 6.23 и результат выполнения запроса в режиме 1С:Предприятие (рис. 6.36) иллюстрируют отбор курсов и преподавателей.



Рис. 6.36. Результат выполнения запроса с полным внешним соединением

Листинг 6.23. Пример запроса с использованием полного внешнего соединения

```
Процедура ОтобратьИнформациюНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Спр1.Наименование КАК Курс,
| Спр2.Наименование КАК Преп,
| Спр1.Ссылка КАК СпрСсылка1,
| Спр2.Ссылка КАК СпрСсылка2,
| Спр1.Стоимость КАК Ст, Спр2.Рейтинг КАК Рейт
| ИЗ Справочник.Курсы КАК Спр1
| ПОЛНОЕ ВНЕШНЕЕ СОЕДИНЕНИЕ
| Справочник.Специалисты КАК Спр2
| ПО Спр1.Преподаватель=Спр2.Ссылка";
Результат = Запрос.Выполнить().Выбрать();
```

```
РезультатыОтбора.Очистить ();
Пока Результат.Следующий() > 0 Цикл
    Если Результат. СпрСсылка1=NULL Тогда
        Если Результат.СпрСсылка2.ЭтоГруппа = Ложь Тогда
            РезультатыОтбора.Добавить (Строка (Результат.Преп) +
                " Рейтинг: " + Строка (Результат. Рейт));
        КонецЕсли;
  Иначе
        Если Результат.СпрСсылка2=NULL Тогда
            РезультатыОтбора.Добавить (Строка (Результат.Курс) +
                  " + Строка (Результат.Ст) + "руб.");
        Иначе
            РезультатыОтбора.Добавить (Строка (Результат.Курс) +
                  " + Строка (Результат.Преп) + " " + Строка (Результат.Ст) +
              "руб. Рейтинг: " + Строка (Результат. Рейт));
        КонецЕсли:
    КонецЕсли:
КонецЦикла;
КонецПроцедуры
```

Здесь в зависимости от полноты строки, выбранной с помощью запроса информации, изменяется формат отображения в элементе РезультатыОтбора. Например, при извлечении строки с информацией только о дисциплине (еще не указан преподаватель) не выводится лишнее обозначение относительно рейтинга специалиста. Аналогична ситуация и для преподавателей.

Ключевое слово ВЫБОР

Рассмотрим еще одну возможность языка запросов — операцию выбора. Для ее реализации предназначено соответствующее ключевое слово — вывор. Операция выбора позволяет получить одно из возможных значений в соответствии с указанными условиями.

Немного изменим конфигурацию для того, чтобы продемонстрировать практический эффект использования данной конструкции языка. Создадим в конфигураторе регистр сведений (непериодический и независимый), в который добавим одно измерение учащийся и один ресурс — Баллы. Фактически это означает, что в этом регистре напротив фамилии каждого учащегося будет фиксироваться количество баллов, которые он набрал по итогам, допустим, определенного периода. Для самого же регистра (рис. 6.37) выберем имя БаллыУчащихся.

Цель рассматриваемой доработки информационной системы заключается в том, чтобы сопоставить определенный интервал баллов конкретной оценке. Например, соответствие может выглядеть так, как показано в табл. 6.3.

Итак, после создания рассмотренного регистра сведений заполним его данными в режиме 1С:Предприятие. На рис. 6.38 показаны внесенные баллы учащихся. Это, можно сказать, первичные данные, которые потребуются нам для формирования отчета, содержащего напротив фамилии каждого учащегося оценку в соответствии с набранными баллами.

🛃 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 ≥			_ 🗆 🗙	
🛛 <u>Ф</u> айл <u>П</u> равка Конфигу	🛛 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка			
🗋 🛋 🖬 🐰 🖻 🕯	👌 🥌 🙆 🖕 🖕 🕻 🦣	* E		
🥤 Регистр сведений	БаллыУчащихся	Свойства: Р	есурс	ŦΧ
Основные	• = X= A I E	21 🛯 🗉 🛛	\times \vee	
▶ Данные		▼Основные:		^
Регистраторы	⊡ — ↓ Измерения	Имя	Баллы	
Формы	і і 🛶 Учащийся	Синоним	Баллы	
Макеты	— Ресурсы — 1 Баллы	Комментари		
Подсистемы	Реквизиты	Индексиров	Не индексировать	-
Права		Полнотексто	Использовать	-
		▼Тип ланны :	x-	~
Действия - <Наза	ц Далее> За	Имя объекта н	метаданных	
Для получения подсказки	нажмите F1			

Рис. 6.37. Окно редактирования объекта конфигурации — регистра сведений БаллыУчащихся

Таблица 6.3. Соответствие баллов и оценок

Номер интервала	Интервал по количеству баллов	Оценка
1	40—50	5
2	31—39	4
3	23—30	3
4	12—22	2
5	0—11	0

Теперь перейдем к основной части — к созданию отчета, который назовем отчетПоУчащимся. В окне редактирования объекта конфигурации на закладке **Формы** создадим форму произвольного вида, на которой расположим два элемента управления — кнопку и поле списка. Для кнопки выберем имя сформироватьОценки, а для поля списка — оценкиУчащихся. Пользователь в режиме 1С:Предприятие, нажимая кнопку, получает заполненный список (рис. 6.39). Сам текст процедуры приведен в листинге 6.24.

Примечание

В операции выбора (которая начинается с ключевого слова вывор) можно указывать сколько угодно альтернативных выборов, используя ключевые слова Когда и Тогда. В процессе выполнения запроса система последовательно проверяет каждое условие. Если указанное условие выполняется, то результатом выбора становится выражение, следующее за ключевым словом Тогда. Значение выражения, указанного после слова Иначе, используется в качестве результата операции выбора в том случае, если во всех ранее указанных условиях результат имел значение ложь. Завершается конструкция выбора словом КОНЕЦ.

🏪 1С:Предприятие (учебная версия) - Конс	фигурация	
<u>■ Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка		
I 🗋 🔟 X 🖻 🖻 🍜 🔍 📲 🛛	🗎 🌌 🛛 м м+ м- 🎇 🖕	
👖 Список Баллы учащихся	_ 🗆 ×	
Действия 🗸 🔄 🛬 式 🎽 🔜 👔 🐚 🎼	T - T 🖸 ?	
Учащийся	Баллы 🔼	
🖂 Пронин Олег Петрович	45	
🖂 Афонин Игорь Николаевич	41	
🔤 Федосеев Кирилл Федорович	31	
🖂 Сидоров Евгений Петрович	27	
🖂 Николаев Сергей Федорович	39	
	V	
г 🌈 Список Баллы учащихся		
Для получения подсказки нажмите F1	CAP NUM	

Рис. 6.38. Информация по баллам учащихся

🏪 1С:Предприятие (учебная версия) - Ко	нфигурация	<u>_ </u>
📱 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна (правка	
ID 🗳 🖬 🐰 🖻 🛍 🚳 🖻 🗇 o	× ×	🛅 🧟 м м+ м- 🎇 🖕
🖹 Форма 📃 🗆	🗙 🎁 Список Баллы учащихся	_ 🗆 ×
Сформировать оценки Оценки учащихся: Сидоров Евгений Петрович 3 Федосеев Кирилл Федорович 4 Николаев Сергей Федорович 4 Пронин Олег Петрович 5 Афонин Игорь Николаевич 5	Действия • 📔 🛬 ≼ 🍇 🖦 🕅 ` Учащийся 🖾 Пронин Олег Петрович 🖾 Афонин Игорь Николаевич 🖾 Федосеев Кирилл Федорович 🖾 Сидоров Евгений Петрович 🖾 Николаев Сергей Федорович	С т К С ? Баллы 45 41 31 27 39
🖹 Форма 🥤 Список Баллы учащихся		
Для получения подсказки нажмите F1 САР NUM		

Рис. 6.39. Использование ключевого слова ВЫБОР в запросе

Листинг 6.24. Пример запроса с использованием операции выбора

```
Запрос. Текст = "ВЫБРАТЬ Учашийся,
 1
   выбор
 I
    Когда Баллы >= 40
        Тогда 5
    Когда Баллы > 30
 Тогла 4
    Когда Баллы > 22
         Тогда З
    Когда Баллы > 11
        Тогда 2
    Иначе
            \cap
    КОНЕЦ КАК Оценка
 ИЗ РегистрСведений.БаллыУчащихся
  УПОРЯДОЧИТЬ ПО ОЦЕНКА ВОЗР";
 Результат = Запрос.Выполнить ().Выбрать ();
ОценкиУчащихся.Очистить ();
Пока Результат.Следующий() > 0 Цикл
    ОценкиУчащихся. Добавить (Строка (Результат. Учащийся) +
            " + Строка (Результат.Оценка));
КонецЦикла;
КонецПроцедуры
```

На рис. 6.39 приведен результат выполнения процедуры, выполненной при нажатии кнопки Сформировать оценки. Видно, что в списке учащихся напротив каждой фамилии появляется оценка, соответствующая набранным баллам.

Рассмотренная возможность может быть эффективно использована при составлении разнообразных отчетов, где требуется сопоставить большой разброс значений определенной характеристики более узкому набору оценок.

Встроенные функции языка запросов

В языке запросов системы 1С:Предприятие определены встроенные функции, которые могут использоваться в выражениях в списке полей выборки и в условии отбора в предложении где. Функций достаточно много, и они разрабатывались с учетом всех возможных практических ситуаций, которые могут встретиться при конструировании запросов. В сопроводительной документации, включаемой в поставку программного продукта 1С:Предприятие 8.1, содержится исчерпывающее описание всех функций. Некоторые из них ориентированы на общие задачи, а некоторые являются более узкоспециализированными. В этом разделе будут приведены пояснения по некоторым из них, на взгляд автора, наиболее используемым.

Функция ПОДСТРОКА

Назначение функции подстрока () заключается в выделении фрагмента строки. У этой функции три параметра. В качестве первого указывается исходная строка, из которой требуется выделить часть символов — подстроку. Понятно, что типом зна-
чения этого параметра является строка. Второй параметр определяет позицию символа в исходной строке, с которого начинается выделение части строки, а третий это длина выделяемой подстроки. Тип значения для второго и третьего параметров — число.

Продемонстрируем использование этой функции на несложном примере. Допустим, нам необходимо из справочника клиенты выбрать несколько начальных символов фамилии каждого клиента (для определенности первые 5 символов). В ряде практических ситуаций это несет краткую, но вполне достаточную информацию о клиентах. Подобный запрос организуется с помощью следующей конструкции: выбрать

Подстрока (Справочник. Клиенты. Наименование, 1, 5)

Один из возможных вариантов результата выборки показан в табл. 6.4.

Информация о клиентах
Афони
Никол
Прони
Сидор
Федос

Таблица 6.4. Выборка фрагментов наименований клиентов

Функция ДобавитьКДате

Функция добавить Кдате () позволяет добавлять к дате некоторую величину. В ней предусмотрены три параметра. Первый параметр представляет дату, к значению которой требуется добавить величину, определяемую вторым и третьим параметрами. Соответственно, тип значения первого параметра — дата. Второй параметр функции определяет единицы измерения, в которых задана величина приращения — третий параметр функции. В качестве второго параметра можно указывать одно из значений: Секунда, Минута, Час, День, Неделя, Месяц, Квартал, год, Декада, Полугодие, а тип значения третьего параметра — Число.

Например, можно использовать рассматриваемую функцию в запросе, приведенном в листинге 6.25.

Листинг 6.25. Запрос с использованием функции добавить Кдате ()

ВЫБРАТЬ

Подстрока (Справочник.Клиенты.Наименование, 1, 5) ДобавитьКДате (ДатаВремя (2007, 8, 14, 10, 50, 20), Месяц, 3)

В этом запросе мы дополнительно воспользовались функцией датаВремя (), которая принимает шесть параметров — год, месяц, число, часы, минуты, секунды. Ис-

пользование данной функции в качестве значения первого параметра другой функции добавитькдате () обеспечивает получение необходимой даты (с указанием еще и времени).

В табл. 6.5 представлена информация о фрагментах фамилий учащихся с указанием даты — результат выполнения запроса из листинга 6.25.

Дата представления сведений
14.11.2007 10:50:20
14.11.2007 10:50:20
14.11.2007 10:50:20
14.11.2007 10:50:20
14.11.2007 10:50:20

Таблица 6.5. Выборка, полученная в результате выполнения запроса из листинга 6.25

Функция РазностьДат

Часто в практических ситуациях требуется вычисление разности между двумя датами, например, между датой окончания и датой зачисления. И функция РазностьДат() предназначена для получения разности между датами. У нее три параметра. Первые два имеют тип дата, а третий определяет, в каких единицах измерения необходимо получить результат. Здесь возможны следующие варианты: Секунда, Минута, Час, День, Месяц, Квартал, Год.

В запросе, приведенном в листинге 6.26, извлекается число дней по указанной разности дат.

```
Листинг 6.26. Запрос с использованием функции РазностьДат ()
```

Справочник. Клиенты. Наименование,

РазностьДат (Справочник.Клиенты.ДатаЗачисления,

Справочник.Клиенты.ДатаОкончания, День)

С учетом данных, имеющихся в информационной базе, результат выполнения представленного запроса может выглядеть так, как показано в табл. 6.6.

Клиенты	Количество дней
Афони	50
Никол	27
Прони	20
Сидор	30
Федос	31

Таблица 6.6. Выборка количества дней индивидуального обучения учащихся

Функция ПРЕДСТАВЛЕНИЕ

Функция представление () предназначена для получения строкового представления значения произвольного типа. В качестве параметра функции может выступать выражение любого типа. Например, можно привести к строковому типу информацию о датах зачисления и окончания с помощью следующего запроса:

```
ВЫБРАТЬ
Справочник.Клиенты.Наименование,
ПРЕДСТАВЛЕНИЕ (Справочник.Клиенты.ДатаЗачисления),
ПРЕДСТАВЛЕНИЕ (Справочник.Клиенты.ДатаОкончания)
```

Оператор ПОДОБНО

Часто при построении запросов мы располагаем неточной информацией о том, что требуется найти, а какой-то ее частью. Например, мы знаем, что фамилия интересующего нас человека включает начальный фрагмент — Сидор. Ситуация достаточно распространенная — мы не помним точно фамилию человека. Она может быть Сидоров, или Сидорюк, или Сидоренко.

Поэтому в среде 1С:Предприятие часто бывает необходимо использовать интерфейс, аналогичный показанному на рис. 6.40. Здесь в окне ввода пользователь может внести любую часть фамилии, если даже он ее не полностью знает. При нажатии кнопки **Выбрать** в списке ниже отображаются существующие в справочнике фамилии учащихся.



Текст процедуры обработки нажатия кнопки приведен в листинге 6.27. Источником запроса является справочник, а отбор выполняется по полю Наименование.

Отбираются только те учащиеся, в фамилии которых присутствует значение параметра фрагмент. Условие отбора обеспечивается за счет следующей конструкции языка запроса:

ГДЕ Учащийся ПОДОБНО &Фрагмент

Здесь оператор подобно позволяет сравнить значение выражения, указанного слева от него, со строкой шаблона, указанной справа. Значение выражения должно иметь тип Строка.

Строка шаблона допускает использование ряда служебных символов (табл. 6.7). В приведенной процедуре значение параметра фрагмент устанавливается уже знакомым способом с использованием одного из служебных символов:

Запрос.УстановитьПараметр ("Фрагмент", "%"+ЭлементыФормы.Данные.Значение+"%")

Символ	Комментарий
8	Обозначает последовательность, содержащую любое количество символов
_(подчеркивание)	Обозначает один произвольный символ
[] один или несколько символов в квадратных скобках	Любой одиночный символ из перечисленных внутри квадратных скобок. В перечислении могут встречаться диапазоны, например, а-z, что означает произвольный символ, входящий в диапазон, включая его концы
[^]	Любой одиночный символ, кроме перечисленных следом за значком отрицания ^

Таблица 6.7. Служебные символы, применяемые в операторе подобно

Листинг 6.27. Отбор информации с указанной подстрокой

```
Процедура ВыборНажатие (Элемент)
```

Запрос = Новый Запрос;

```
Запрос.Текст = "ВЫБРАТЬ Наименование
```

| ИЗ Справочник.Клиенты

```
| ГДЕ Наименование ПОДОБНО &Фрагмент";
```

```
Запрос.УстановитьПараметр("Фрагмент", "%"+ЭлементыФормы.Данные.Значение+"%");
```

Результат = Запрос.Выполнить().Выбрать();

ОтобранныеУчащиеся.Очистить();

Пока Результат.Следующий() > 0 Цикл

ОтобранныеУчащиеся.Добавить (Строка (Результат.Наименование));

КонецЦикла;

КонецПроцедуры

Результат выполнения процедуры показан на рис. 6.41.

🏪 1С:Предприятие (учебная версия) - Ко 💶 🗙	
📱 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
🗄 🗋 🚢 🍣 🖥 🎆 🛅 🧟 M M+ M- 🎇 🗸	
🖹 Форма 💶 🗙	
Данные для поиска: d Отобранные учащиеся: Сидоров Евгений Петрович Николаев Сергей Федорович Федосеев Кирилл Федорович	Рис. 6.41. Результат отбора по указанной подстроке
Выбор	
Форма	
Для получения подсказки нажмите F1 САР NUM	

Результат использования запросов

Таким образом, мы познакомились с языком запросов. Разнообразные примеры продемонстрировали этот ключевой инструментальный объект, который используют разработчики для построения необходимых выборок и отчетов по базе данных. Мы, взяв за исходный материал созданные ранее информационные базы, фактически дополнили созданные ранее конфигурации средствами просмотра информации в удобном для пользователя виде.

Большинство примеров приведено в полном виде, включающем как текст запроса, так и саму процедуру, позволяющую запрос выполнить. Такое представление должно быть удобным для читателей. Конечно, конструкции языка запросов весьма разнообразны, и всю информацию на данную тему мы не охватили. Однако теперь читатели уже могут самостоятельно отправиться по лабиринтам программных ресурсов системы 1С:Предприятие, используя встроенную справочную документацию, а также ряд очень хороших изданий [1, 7, 8, 9].

С языком запросов мы встретимся и в последующих главах, где много внимания уделим построению отчетов. Но в целом, уже сейчас читатели могут самостоятельно строить запросы при решении собственных задач выборки информации из таблиц баз данных.

глава 7



Отчеты и система компоновки данных

При решении любых задач, связанных с количественным учетом, ключевым компонентом являются отчеты. В предыдущих главах на ряде примеров было рассмотрено создание отчетов без использования специально предназначенных для этого средств автоматизации или с их минимальным использованием. Так, мы сами формировали макеты для отчетов и создавали программные процедуры, которые в режиме 1С:Предприятие приводили к заполнению табличного документа в соответствии со структурой макета. Одним из основных достоинств системы 1С:Предприятие 8.1 являются эффективные возможности построения отчетов с минимальными усилиями со стороны разработчика. Наличие обеспечивающих это средств автоматизации повышает скорость разработки прикладных решений, а также предоставляет возможность самостоятельного формирования необходимых отчетов пользователям, не знакомым со встроенным языком программирования системы 1С:Предприятие.

Данная глава будет посвящена примерам использования системы компоновки данных. Эта технология отсутствовала в версии 1С:Предприятие 8.0, а ее появление в программном продукте 1С:Предприятие 8.1 позволяет пользователю разрабатывать отчеты без работы с программным кодом на встроенном языке. Система компоновки данных — это новое средство, которое предназначено для создания отчетов на основе декларативного описания. Механизм компоновки данных позволяет дать целостное описание отчета, используя только визуальные средства. Это отличает систему компоновки данных от построителя отчетов (основного средства автоматизации при создании отчетов в системе 1С:Предприятие 8.0).

При работе с системой компоновки данных на этапе визуального конструирования создается основа отчета — схема компоновки данных. Также создаются и настройки по умолчанию. В дальнейшем при интерактивной работе с отчетом в режиме 1С:Предприятие пользователь может изменить имеющиеся настройки отчета, а также создать собственные.

Система компоновки данных позволяет реализовывать новые возможности, отсутствующие у построителя отчета. Все они приведены в документации, прилагаемой к рассматриваемому программному продукту. Отметим основные черты системы компоновки данных:

- создание отчета без программирования;
- использование автоматически генерируемых форм отчета;
- настройка структуры отчета;
- совмещение в отчете нескольких таблиц;
- создание вложенных отчетов.

Прежде чем начать практические шаги, посмотрим на систему компоновки данных в целом. Она является набором элементов, каждому из которых отводится конкретное место при построении отчета. В итоге процесс создания отчета представляет собой переход от одного элемента к другому. Перечислим основные элементы системы компоновки данных:

- схема компоновки данных определяет суть данных, которые используются отчетом (откуда следует получать данные и как управлять их компоновкой);
- настройки компоновки данных представляют собой описание всего, что может быть настроено в установленной схеме компоновки данных (т. е. отбор, упорядочивание, параметры получения данных, параметры вывода данных и др.);
- макет компоновки данных задает готовое описание того, как должен формироваться отчет (он представляет собой результат применения конкретных настроек к схеме компоновки);
- элемент результата компоновки данных (выделение элемента результата компоновки данных связано с тем, что единого компонента результата компоновки данных нет, а существуют только его элементы).

Процесс компоновки данных состоит из нескольких этапов. Начальный этап называется созданием схемы компоновки данных. Этот этап чаще всего выполняется с использованием визуальных средств (конструктора схемы компоновки данных). Далее в книге мы будем использовать именно эти средства. Кроме того, можно использовать и программный текст на языке XML. Также возможно и использование программных объектов встроенного языка 1С:Предприятие 8.1.

Следующие этапы процесса компоновки данных связаны с редактированием настроек компоновки данных, подготовкой к исполнению (процесс формирования макета компоновки данных), исполнением компоновки данных, выводом результата компоновки данных (отчет может быть выведен в различных форматах).

Особенностью отчета, формируемого с использованием системы компоновки данных, является то, что он может включать различное сочетание таких элементов, как:

- группировка;
- таблица;
- ♦ диаграмма;
- вложенный отчет.

Теперь, после взгляда на систему компоновки данных в целом, рассмотрим ее на практических примерах. При этом в качестве исходной информации будем исполь-

зовать разработки, уже созданные в предыдущих главах. Для начала обратимся к конфигурации, которая была нами разработана в *главе 3*.

Отчет по остаткам товаров

Итак, начнем создание нового отчета, который будет относиться к информационной базе, разработанной в *главе 3*. Она связана с фиксированием поступлений товаров. В *главе 3* для хранения информации о товарах мы создали регистр накопления информацияотоварах. Данный регистр предназначен для отражения поступления и отгрузки товаров. При работе в режиме 1С:Предприятие информационная структура, созданная системой на основании данного объекта конфигурации, фиксирует все движения о поступлениях и отгрузках товаров.

Итак, в режиме конфигуратора откроем информационную базу, разработанную в *славе 3*. Первые шаги хорошо знакомы — необходимо создать новый объект конфигурации — отчет (рис. 7.1). На закладке **Основные** введем имя отчета отчет7_1. Такой выбор названия связан с тем, что далее в этой главе будет создан ряд отчетов, и подобный стиль названий отражает номер главы в совокупности с номером отчета. В реальных разработках рекомендуется давать отчетам более информативные названия.

🛃 Конфигуратор (учебная версия)	- Конфигурация									
<u>Ф</u> айл <u>П</u> равка Ко	нфигурация Отлад	ка Администрирование <u>С</u> ервис <u>О</u> кна									
Справка											
I 🗋 🖬 🔏 📾 📾 🖢 🖕 🛤 📃 💌 👻											
🖤 Отчет Отчет7_1 📃 🗆 🗙											
• Основные	Има	Oruer7.1									
Данные	KINDH.										
Формы	Синоним:	Отчет7 1									
Макеты	Комментарий:										
Подсистемы											
Права											
Интерфейсы											
Прочее	Основная схем	а компоновки данных:									
		×Q									
		Открыть схему компоновки данных									
Действия	≺Назад	Далее» Закрыть Справка									
Для получения подск	азки нажмите F1	CAP NUM ru V									

Рис. 7.1. Закладка Основные окна редактирования объекта Отчет 7 1

Теперь на той же закладке воспользуемся кнопкой **Открыть схему компоновки** данных. Для этого действия можно также использовать кнопку с изображением лупы напротив поля ввода с подписью **Основная схема компоновки данных**. Это

является первым шагом процесса создания схемы компоновки (ранее при создании отчетов мы ее не использовали).

В результате перед нами открывается окно конструктора макета (рис. 7.2). Установленный переключатель Схема компоновки данных отражает тип макета, который в данном случае используется. Заметим, что при выполнении практических действий значения параметров в большинстве окон, как правило, изменять не требуется — по умолчанию они установлены верно.

Конструктор ма	кета Х											
Имя:	Имя: ОсновнаяСхемаКомпоновкиДанных											
Синоним:	Ссновная схема компоновки данных											
Комментарий:												
Выберите тип макета:												
🕖 Табличный д	цокумент											
🔵 Текстовый д	цокумент											
🕖 Двоичные д	анные											
 Active docum 	nent											
💿 НТМL докум	ент											
🕖 Географичес	ская схема											
💿 Графическа:	я схема											
💿 Схема комп	оновки данных											
🔘 Макет офорі	мления компоновки данных											
Загрузить из фа	Загрузить из файла:											
	Готово Отмена Справка											

Рис. 7.2. Окно конструктора макета схемы компоновки данных

После нажатия кнопки **Готово** перед нами откроется окно конструктора схемы компоновки данных, которое содержит много закладок. На рис. 7.3 показана закладка **Наборы данных**. С помощью конструктора можно создавать или редактировать источники данных (в качестве которых в настоящее время может выступать только текущая информационная база). Также можно создавать и редактировать наборы данных, в качестве которых могут выступать:

- запрос (используется язык запросов и инструкции специального расширения языка запросов для системы компоновки данных);
- произвольный источник данных;
- объединение, механизм которого аналогичен объединению запросов средствами языка запросов.

Продолжим практические действия и, нажав кнопку Добавить набор данных (можно также воспользоваться контекстным меню, открывающимся по щелчку правой кнопкой по пиктограмме Наборы данных), выберем вариант Добавить набор данных — запрос. В результате окно на экране изменится (рис. 7.4), и в правой части мы увидим кнопку Конструктор запроса.

📩 Конфигуратор (уч	чебная версия) - Конфигурация									
🛛 <u>Ф</u> айл <u>П</u> равка Конс	фигурация Отла	дка Администрир	ование <u>С</u> ер	овис <u>О</u> кна С	правка						
🗋 🚄 🖬 🐰 🖻	6 6	5 C 🕅 M			▼ ¥	🗑 🛅 🔭 🐪	📮 💁 🖕				
💐 Отчет Отчет7_1	: ОсновнаяСх	емаКомпоновки,	Данных				_ 🗆 ×				
Наборы данных Св	язи наборов	Вычисляемые	Ресурсы	Параметры	Макеты	Вложенные схе	Настройки				
à • '≅							엄점				
Наборы да	1мя источника д	анных					^				
1 II											
Для получения подска	Для получения подсказки нажмите F1 [CAP] NUM] ги ▼										

Рис. 7.3. Закладка Наборы данных конструктора схемы компоновки данных

2	😽 Ko	нфигура	тор	(уче	бная	вер	сия) - Н	(онфи	гура	ция											_ 🗆	×
I	<u>Ф</u> ай	іл <u>П</u> равк	a K	онфи	урац	ия (Этладка	Адм	инистр	ирова	ние <u>С</u>	ервис	<u>О</u> кна	Спра	вка							
I		🛋 🖬	Ж	B _C	b.	ā	0 1	¢	33	29				-	8	a 1	5 🕅	?#4	P	<mark>™?</mark> ↓		
	0 🌾	тчет Отч	чет7	_1: 0)сно	вная	Схема	Комп	онові	иДан	ных										_ 🗆	×
	Набо	оры данны	ых	Связи	н набо	ров	данных	Выч	исляем	иые по	ля Р	есурсы	Пара	аметры	ы Ма	кеты	Вложе	енные	схем	ы На	астройки	1
	2 11	• × =				Пол	я:												2	12	🗳 ×	
	Ð	•Наборы,	данн Іабор	ых оДанн	њж1		Поле		Путь Заго.	повок	Огр Огр По.	раничен раничен Ус	ие пол ие рек Гр.,, У	я Ро в п	оль		Выраж Выраж упоряд	ение ения очива	Пр На э Па	оверка бор да раметр	а иерархі нных Э	И
						<																>
						3anj	000:										ľ	Ko	нстру	ктор з	апроса	
						<															1	×
	1					Авт	озаполн	ение	•													
1	1ля п	олучения	пода	казкі	инаж	мите	F1												CAP	NUM	ru ▼	//

Рис. 7.4. Результат добавления набора данных в схему компоновки

Теперь мы должны создать необходимый запрос для получения количества имеющихся товаров. Для этого щелчком по кнопке **Конструктор запроса** следует перейти к разработке запроса (ранее мы самостоятельно писали его текст на языке запросов, а теперь сформируем текст как результат визуальных действий).

На рис. 7.5 представлено окно конструктора запроса, в котором присутствуют три раздела:

- ♦ База данных;
- Таблицы;
- ♦ Поля.



Рис. 7.5. Окно конструктора запроса

В разделе **База данных** представлены имеющиеся таблицы для формирования запросов. Эти таблицы отражают все данные, имеющиеся в информационной базе. На основании информации, содержащейся в этих таблицах, строятся разнообразные отчеты.

Если раскрыть ветку Регистрынакопления, то кроме созданных нами регистров ИнформацияОТоварах и ИнформацияОЗаявках мы увидим еще несколько виртуальных таблиц, связанных с данными регистрами. Так, три виртуальные таблицы формируются системой на основании информации в регистре ИнформацияОТоварах:

- ИнформацияОТоварах.Остатки;
- ИнформацияОТоварах.Обороты;
- ИнформацияОТоварах.ОстаткиИОбороты.

Нас в данный момент интересуют остатки товаров. Поэтому перенесем в раздел **Таблицы** таблицу информацияОТоварах.Остатки. Теперь для запроса необходимо выбрать используемые поля. В рассматриваемой ситуации нам следует перенести два поля (товар, количествоОстаток) в раздел **Поля** (рис. 7.6). Окно конструктора запроса имеет еще ряд закладок, позволяющих формировать необходимые указания в запросе. Так, можно создавать группировки, задавать условия, назначать псевдонимы полям, задавать порядок и т. д. В последующих примерах этой главы мы с этими действиями встретимся, а пока, после того как необходимые поля для запроса выбраны, можно просто нажать кнопку **ОК**. В результате визуальные действия привели к формированию запроса, текст которого приведен в листинге 7.1.



Рис. 7.6. Отбор полей для запроса по заказам

Листинг 7.1. Запрос для получения информации о количестве товаров

```
ВЫБРАТЬ
```

```
ИнформацияОТоварахОстатки.Товар,
```

```
ИнформацияОТоварахОстатки.КоличествоОстаток
```

ИЗ

```
РегистрНакопления.ИнформацияОТоварах.Остатки
КАК ИнформацияОТоварахОстатки
```

В результате проделанных действий на закладке **Наборы** данных произойдут изменения. На рис. 7.7 показано заполнение полей, которое в этом случае будет выполнено автоматически.

2	🚽 Конфигуратор (учебн	ая в	ерсия) - Конфигураці	ия								
I	<u>Ф</u> айл <u>П</u> равка Конфигур	ация	. Отладка Администри	рование <u>С</u> ера	ис <u>С</u>	кна	Спра	вка				
=	- C 🔟 🖬 🐰 🖷 📾 🥌 🖻 🖕 🕅 🛪 🔽 🔽 🐨 🐨 🕅 🕅 🥱 🐅 🖪 😁 🖕											
	💐 Отчет Отчет7_1: Осі	нові	наяСхемаКомпоновки	ıДанных								
	Наборы данных Связи н	боры данных Связи наборов данных Вычисляемые поля Ресурсы Параметры Макеты Вложенные схемы Настройки										
	🛅 🕶 🎽	Пол	19.									1
	🖽 Наборы данных		Поле	Путь	Огра	аниче	ние п	рля	Роль	Выражение	Проверка иерархии:	Тип значения
	НаборДан			Заголовок	Orpa	ниче	ние р	ЭКВ	-	Выражения	Набор данных	Доступные
			КоличествоПстаток	Количество	110	90	I p	91.		упорядочива	Параметр	эпачения
		1	Konneerbooerarok	Количество					-			
			Организация	Организация	· ·			v				
				Организация	~	П	v	1				
		-	Товар	Товар					Измерение			
				Товар								
3anpoc:									🌇 Конструк			
			BMEPATE									
			Информаци:	яОТоварах(ста	тки	. Toe	ap,				
H			Информаци:	яОТоварах(ста	тки	. Кол	иче	ствоОстато	or		
			ИЗ РепистрНа:	коппения Б	fæðo	nwai	тиаО	Тов	anay Ocmar	ржи <mark>Как</mark> Ин	donwauwgOTosar	ахОстатки
			i cisso i pila			prict		.05	apan i oo fa.	MAN MA	*-buodwwo.opal	oun of the trap
1	іля получения подсказки н	ажм	ите F1								(CAP NUM ru* //

Рис. 7.7. Заполнение полей в схеме компоновки данных

Примечание

Поля для текущего набора данных будут сформированы системой автоматически только в случае, если в конструкторе схемы компоновки данных установлен флажок **Автозаполнение**.

Прокомментируем раздел **Поля** на рис. 7.7. Первый столбец с информацией называется **Поле**, и в нем отражается имя поля набора данных, для которого формируется описание. Учитывая, что для получения набора данных мы воспользовались запросом, содержание поля определяется именем поля в тексте запроса. Важно отметить, что информация в столбце **Поле** не допускает редактирования.

Путь — это имя поля набора данных, под которым к нему можно обращаться в других разделах конструктора схемы компоновки данных. Это значение можно устанавливать самостоятельно.

Заголовок — еще одно имя поля набора данных, под которым оно будет фигурировать в настройках в режиме 1С:Предприятие, а также в названии данного поля в шапке самого отчета. Значение в этом поле также может быть назначено разработчиком самостоятельно.

Настройка в столбце **Роль** показывает, каким образом поле будет использовано в дальнейшем в системе компоновки данных. Информация о том, что поле является измерением, используется при расчете итогов.

Сейчас наша задача заключается в том, чтобы получить информацию об остатках товаров. Для этого перейдем на закладку **Настройки**, где необходимо задать настройки отчета по умолчанию. А именно следует щелкнуть правой кнопкой мыши на элементе **Отчет** и в появившемся контекстном меню выбрать пункт **Новая группировка**.

Примечание

Закладка Настройки в нижней части окна имеет еще ряд дополнительных закладок, дающих разнообразные возможности влиять на формирование отчета.

Элемент **Группировка** позволяет отображать в отчете не только сгруппированную определенным образом информацию, но и детальные записи, полученные системой в результате обработки текста запроса. И если мы хотим отображать в отчете детальные записи, то в окне **Группировка** (рис. 7.8) следует раздел **Поле** оставить пустым.

Группировка 🛛 🗙								
Поле:	x							
Тип:	Без иерархии							
	ОК Отмена							

Рис. 7.8. Окно для указания группировочного поля

Перейдем на дополнительную закладку **Выбранные поля** (рис. 7.9), где в число выбранных полей следует добавить два поля: товар и количествоОстаток. Добавление полей в раздел **Поля** проще всего произвести, дважды щелкнув по нужному полю в разделе **Доступные поля**. Также для этого можно воспользоваться перетас-

киванием названий из раздела Доступные поля. Третий способ добавления — использование кнопки Добавить из пиктографического меню, расположенного в правой части окна.

После таких несложных визуальных действий уже можно проверить формирование отчета (рис. 7.10). Для этого следует перейти в режим 1С:Предприятие, в разделе меню **Отчеты** обратиться к отчету с нужным названием и в открывшемся окне нажать кнопку **Сформировать**.

Г								1
M	Конфигуратор (учеоная версия) - конф	игурация						-
-	<u>Ф</u> айл <u>П</u> равка Конфигурация Отладка Адмі	инистрирование	<u>С</u> ервис <u>О</u>	кна Справка	э			
	D 🛋 🖬 🕺 🖓 🖬 📾 🥌 🖉 🕁 😋	M M		- 3	a 7a T	5 🕅 🔭	?#** ~*	
	💐 Отчет Отчет7_1: ОсновнаяСхемаКомп	юновкиДанны	к					
	Наборы данных Связи наборов данных Выч	исляемые поля	Ресурсы	Параметры	Макеты	Вложен	ные схе	M
	⊡-Ш) Отчет 							
	Настройки: Отчет <Детальные записи>							
	Параметры данных Выбранные поля Отбо	ор Сортировка	Условное с	оформление	Пользов	ательски	ю поля	[
	Доступные поля	Поле						
	————————————————————————————————————	⊡…Выбран	ные поля Говар					
	⊞… I Системныеноля 	✓ : ►	количествоц	ЈСТАТОК				
	Для получения подсказки нажмите F1				CAP	NUM	u•	1

Рис. 7.9. Заполнение закладки Выбранные поля

🏪 1С:Предприятие (уче)	бная версия) - Ко	онфигурация	
🛛 <u>Ф</u> айл <u>П</u> равка Таблица	Операции <u>С</u> ервис	: <u>О</u> кна Сп <u>р</u> авка	
🗋 📫 🖬 🖓 🛍	a 🖸 🖕	👻 i 📰 🛅 🧟	м м+ м- 🎇 🗸
🕸 Отчет Отчет71			_ 🗆 ×
Действия - 🕨 Сформир	овать 📑 🦄 Конст	руктор настроек	Настройки 🍟
Товар	Количество остаток		^
Мякая мебель	6		
Стенка (мод. 15-7)	11		
Кухня Уют-З	6		
٢			>
🗊 Отчет Отчет71	📇 Документ	ы Поступление	
Для получения подсказки н	ажмите F1		CAP NUM

Рис. 7.10. Формирование отчета по остаткам товаров

Использование системы компоновки данных позволяет расширить возможности формирования отчетов без внесения изменений в конфигурацию. Рассмотрим, как это выглядит на практике. Будем считать, что в нашем отчете необходимо просмотреть суммарную информацию о том, сколько всего в настоящий момент имеется товаров (необходимо просуммировать все остатки товаров). Для этого не требуется переходить в режим конфигуратора и перерабатывать имеющийся отчет, а достаточно лишь в окне, приведенном на рис. 7.10, перейти на закладку **Настройки** и в списке выбранных полей снять флажок напротив поля **Товар** (рис. 7.11).

Теперь осталось щелкнуть по кнопке **OK** и далее сформировать уже новую версию отчета (рис. 7.12). Важно, что такие изменения в формировании отчета (изменение вида отчета) мы выполнили непосредственно в режиме 1С:Предприятие. На практике с помощью подобных действий даже из очень простых отчетов можно получать разнообразную итоговую информацию.

🖤 Настройки отчета Отчет7 1	□ ×
⊡-ШР <mark>Отчет</mark> 	₩ • • •
Настройки: Отчет Отчет	
Параметры Выбранные Отбор Сортировка Условное о Пользовате Другие н	аст
Доступные поля — Количество остаток — Товар — Системные поля — Параметры Поле — Выбранные поля — Товар — Количество остаток — Состемные с с с с с с с с с с с с с с с с с с с	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
ΟΚ Ο)тмена

Рис. 7.11. Настройка отчета для получения сводной информации

🏪 1С:Предприятие (учебная версия) - К	онфиг 📕	
<u> <u> </u> </u>	е (учебная ве	ерсия) - М
🗋 📫 🕌 💥 🖬 🗐 🙀 1	M M+ M-	₩
🕸 Отчет Отчет71	_ 1	⊐ × ∣
Действия - 🕨 Сформировать		» ▼
Количество остаток 23		~
🗒 Отчет Отчет71 🛛 📇 Документ	ы Поступлен	ие
Для получения подсказки нажмите F1	CAP NUM	1 //

Рис. 7.12. Отчет по суммарным остаткам товаров

Использование конструкторов настроек

Если охарактеризовать проделанную работу по созданию отчета с помощью системы компоновки данных в первом разделе этой главы, то можно выделить две составляющие:

- создание набора данных для отчета, что было сделано с помощью конструктора запросов;
- создание необходимых настроек для отчета, где были использованы ручные действия.

Однако кроме ручного создания настроек можно воспользоваться и *конструктором настроек*. В этом разделе мы создадим настройки для отчета по остаткам товаров (он будет аналогичен предыдущему отчету), но при этом воспользуемся конструктором настроек.

Итак, вернемся к моменту, когда с помощью визуальных средств был сформирован запрос к базе данных (см. рис. 7.7). Теперь на закладке **Настройки** следует вызвать контекстное меню, которое открывается щелчком правой кнопкой на разделе **Отчет**. В этом меню необходимо воспользоваться пунктом **Конструктор настроек**. В результате перед нами откроется окно, показанное на рис. 7.13. На первом шаге требуется выбрать тип отчета, который мы собираемся создать. В данном случае нам необходим **Список**. Примеры других вариантов отчета (**Таблица** и **Диаграмма**) будут рассмотрены далее в этой главе. Надо заметить, что в предыдущем разделе, когда создание настроек производилось вручную, отчет был также сформирован в виде списка.



Рис. 7.13. Работа с конструктором настроек (шаг 1)

Примечание

В информационной базе на компакт-диске рассматриваемый отчет находится под именем Отчет7_2.

Вернемся к разработке и, отметив вариант Список (рис. 7.13), нажмем кнопку Далее.

Следующее окно, которое мы увидим на экране, представлено на рис. 7.14. Здесь от нас требуется определить поля, выводимые в отчет. И из левого списка доступных полей следует перенести два поля: Товар и КоличествоОстаток.

Примечание

Для изменения порядка расположения полей в отчете следует пользоваться кнопками **Вверх** и **Вниз**.

Выполнив этот шаг, нажмем кнопку Далее (см. рис. 7.14), что приведет к появлению на экране следующего окна, показанного на рис. 7.15. На этом шаге нам предоставляется возможность сформировать в отчете поля группировки. Например, если в базе данных имеется несколько записей, касающихся одного и того же товара, то в этом случае в отчете можно получить суммарные данные по этому товару. Однако, учитывая, что таблица информацияОтоварах.Остатки является виртуальной и информация по остаткам товаров уже сгруппирована, этот этап создания отчета следует пропустить — сразу нажмем кнопку Далее (см. рис. 7.15).



Рис. 7.14. Работа с конструктором настроек (шаг 2)

В результате на экране мы увидим следующее окно (рис. 7.16). Здесь при необходимости упорядочивания данных в будущем отчете требуется указать конкретные поля. В данном случае разумно упорядочить информацию по товарам по возрастанию (от А до Я). Для этого достаточно перенести в правый список (из левого) поле **Товар**. Как видно, кнопка **Далее** неактивна, что говорит о том, что данное окно диалога последнее и осталось только щелкнуть по кнопке **ОК**. В результате мы получим настройки, аналогичные предыдущему отчету, но на этот раз они были сделаны конструктором. Если перейти в режим 1С:Предприятие, то перед нами откроется уже знакомый отчет (см. рис. 7.10).

Конструктор настроек компоновки данных				×
Группировки Момениатура Склад группировок отображаются в отчете.				
Доступные поля		Поле	Тип группировки	
🖃 — 🗕 Товар				
— Код				*
— – Наименование				
Пометка9даления				
Предопределенный	>			
	>>			
	<			
	<<			
< Назад Д	lалее >	ОК	Отмена Справ	ка

Рис. 7.15. Работа с конструктором настроек (шаг 3)

Конструктор настроек компоновки данны <u>Ne Howencarype</u> <u>1 10:Пред8.7</u> <u>3 10:Пред8.1</u> отчете.	ж Толя уг	торядочивания задают	х г порядок следования данных в
Доступные поля		Поле	Направление сортиро 🔺
∰… — Товар ∰… § КоличествоОстаток	>	— Товар	По возрастанию
< Hasan		Далее > ОК	Отмена Справка

Рис. 7.16. Работа с конструктором настроек (шаг 4)

Отчет в виде таблицы

Предыдущие примеры касались использования одного формата отчета — вывода данных в виде списка. Рассмотрим формирование отчета в виде таблицы в этом случае данные находятся на пересечении определенного столбца и конкретной строки. Будем считать, что нас интересует информация о поступлениях товаров от организаций. И по горизонтали в отчете сформируем перечень организаций, а по вертикали — перечень товаров. Содержательная часть таблицы будет касаться количества конкретных товаров от определенных организаций.

Продолжим работу с информационной базой из *главы 3* и создадим новый отчет — Отчет7_3. При этом, как обычно, воспользуемся кнопкой **Открыть схему** компоновки данных. На первом этапе необходимо создать запрос для получения информации о поступивших товарах. Фактически нас в данном случае интересует приход товаров. Поэтому перенесем в раздел **Таблицы** таблицу ИнформацияОТоварах.Обороты и выберем необходимые поля (рис. 7.17):

- ♦ Товар;
- Организация;
- ♦ КоличествоПриход.



Рис. 7.17. Окно конструктора запроса

В результате проделанные визуальные действия привели к формированию запроса, текст которого показан в листинге 7.2.

Листинг 7.2. Запрос для получения информации о поступлении товаров

ВЫБРАТЬ

ИнформацияОТоварахОбороты.Организация,

ИнформацияОТоварахОбороты.Товар,

ИнформацияОТоварахОбороты.КоличествоПриход

ИЗ

РегистрНакопления.ИнформацияОТоварах.Обороты КАК ИнформацияОТоварахОбороты Группировка информации в отчете реализуется с помощью закладки **Ресурсы**. Для указания в отчете конкретного ресурса необходимо перенести его в раздел **По**ле (рис. 7.18). Для этого достаточно выполнить двойной щелчок мышью по полю

🗄 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка
D 🛋 🖬 🐰 🔚 🛍 🥌 🙋 🖕 d 间 🛤 🛛 🍹 ! 🕀 🐥 📃 📃 💌
II 🕼 🔐 🕋 📰 🔿 🧅
💐 Отчет Отчет7_3: ОсновнаяСхемаКомпоновкиДанных 💶 🗖 🗙
Наборы данн Связи набор Вычисляемы Ресурсы Параметры Макеты Вложенные Настройки
Доступные поля Поле Выражение Рассчитывать по - КоличествоПриход > - КоличествоПриход Сумма[КоличествоПр] - Товар >>
Щ/ Отчет Отчет 7_3 🛃: ОсновнаяСхемаКомпон

Рис. 7.18. Установки на закладке Ресурсы



Рис. 7.19. Окно конструктора настроек отчета

в разделе Доступные поля. Можно воспользоваться кнопкой, расположенной на разделительной полосе. В колонке **Выражение** содержится выражение, по которому будет вычисляться значение ресурса. По умолчанию используется выражение сумма, которое нас устраивает, но доступны и другие функции работы с данными.

Следующий этап заключается в создании настроек отчета, для чего воспользуемся уже знакомым конструктором настроек. В данном случае нам требуется указать вариант **Таблица** (рис. 7.19), после чего в следующем окне отметить все имеющиеся поля для отчета (рис. 7.20).

В следующем окне (рис. 7.21) определим поля для группировки данных в отчете. После этого укажем упорядочивание по товару (рис. 7.22).



Рис. 7.20. Выбор полей отчета

Конструктор настроек компоновки да	нных			×
колонок отчета. Поля группир	овок (отображаются в отче	яте.	^
Доступные поля		Строки:		
🖽 — 🗕 Организация	÷	Поле	Тип группировки	F
́— Товар	>>	🖛 Товар	Без иерархии	
	<		4	
	<<	Kazaurur		- 11
	\sum	Поле	Тип соцерировки	
	>>	- Организация	Без иерархии	
				L
	<			
	<<	Tofauur		
	>	Таолицы. Поле	Тип соцопновки	
	>>	110/10	Типтруппировки	r
			1	L
	<			
	<<			~
< Назад	Д	алее > ОК	Отмена Справ	вка

Рис. 7.21. Выбор полей для группировки

В результате проделанных действий после нажатия кнопки **ОК** на экране появится новое окно с автоматически созданными настройками (рис. 7.23).

Теперь разработка отчета завершена, и можно перейти в режим 1С:Предприятие для формирования отчета по имеющимся данным (рис. 7.24).

Конструктор настроек компоновки дан	ных			×
№ Номенклатура Выберите поля упорядочивания. Поля упорядочивания задают порядок следования 1 10: Пред. 8.0 данных в отчете. 3 10: Пред. 8.1 стичете.				
Доступные поля		Поле	Направление сортир 🔺	
🖽 – Организация		— Товар	По возрастанию	
🕂 🗕 Товар			+	
🗄 🖷 🔋 КоличествоПриход				
	>			
	>>			
				~
< Назад	,	Далее > ОК	Отмена Справи	ka

Рис. 7.22. Установки на закладке упорядочивания данных

🛃 Конфигуратор (учебная версия) - Конфигурация 📃 🔲 🛽	X
🛛 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка	
I 🗋 🖬 🔏 🚳 🖆 de 🖬 🛤 🛛 🐥 🐥 📜 🔍	•
Ĩ₩ ₩ ∰ Ĩ Þ , I Ж К Ц 🗄 • <u>◊</u> • <u>▲</u> • <u>∠</u> • E = = = 2	
🗟 Отчет Отчет7_3: ОсновнаяСхемаКомпоновкиДанных 📃 🗗 🕽	×
Наборы данных Связи наборов данных Вычисляемые поля Ресурсы Параметры Макеты Вложенные схемы Настройки	
Стиет Стиет Стороки Сторок	
 Кастройки: Отчет Отчет 	:
Параметры данных выоранные поля отоор сортировка эсловное оформление пользовательские п другие настроики	
Доступные поля	
1/ Отчет Отчет7 3 🔊: ОсновнаяСхемаКомпон	
Для получения подсказки нажмите F1 САР NUM ги -	/

Рис. 7.23. Автоматическое формирование настроек отчета

👺 1С:Предприятие (учебная версия) - Конфигурация 📃 🗖 🗙						
<u>∲</u> айл <u>П</u> равка Таблица (<u>Ф</u> айл <u>П</u> равка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка					
I 🗋 🛋 🖬 🕺 🖌 🖻	🛎 🗟 🖕 c	M +	📰 📄 🧟 м	м+ м- 🞇 🖕		
🖤 Отчет Отчет7 3				_ 🗆 ×		
Действия 🗸 🕨 Сформир	овать 🥂 🔥 Конст	руктор настроек	Настройки	1 1 ?		
Товар	Тико	Юниор	Итого	^		
	Количество	Количество	Количество			
	приход	приход	приход			
Кухня Уют-З	2	17	19			
Мяткая мебель	5	40	45			
Стенка (мод. 15-7)	23	12	35			
Итого	30	69	99			
<				×		
🚉 Отчет Отчет71	🛄 Отчет Отче	ет7 2	💵 Отчет Отчет7	3		
Для получения подсказки на:	жмите F1		C	AP NUM /		

Рис. 7.24. Отчет в виде таблицы

Отчет в виде диаграммы

После рассмотрения отчетов в виде списка и таблицы осталось привести пример использования в отчете диаграммы. А именно, разработаем отчет в варианте, представленном на рис. 7.25.



Рис. 7.25. Отчет в виде диаграммы



Рис. 7.26. Окно конструктора запроса

Начальные шаги не комментируем, а первый ключевой момент связан с формированием запроса для получения информации о поступлениях товаров (рис. 7.26).

На следующем этапе сделаем уже встречавшуюся в предыдущем примере (см. рис. 7.19) установку на закладке **Ресурсы**. Далее воспользуемся конструктором настроек (рис. 7.27).

Конструктор н	астроек компоновки данных	×
Тип отчета:		
	О Список. Данные по всем измерениям отчета выводятся списком.	
	О Таблица. Данные выводятся по измерениям расположенным как по горизонтали, так и по вертикали.	
	 Диаграмма. Данные выводятся в виде диаграммы. 	
	< Назад Далее > ОК Отмена Справка	

Рис. 7.27. Окно конструктора настроек

В следующем окне укажем, что для отчета нам потребуются три поля:

- 🕈 Товар;
- Организация;
- ♦ КоличествоПриход.

После этого очередной шаг работы с конструктором настроек представлен на рис. 7.28. Здесь необходимо определить поле Организация в качестве серий, а поле товар в качестве точек. Фактически на этом этап разработки отчета завершен, а в режиме 1С:Предприятие его мы уже видели (см. рис. 7.25).

Конструктор настроек компоновки данны	ях			×
Диаграммы Выберите поля по которым будет диаграммы. Поля группировок от	г произ тображ	водиться группировка аются в отчете.	а диаграмм, серий и точек	
Доступные поля		Серии:		
🖽 — Организация		Поле	Тип группировки	+
🗄 🗝 Товар	1	📼 Организация	Без иерархии	21
				*
	<			
	<<			
		Точки	/	
	5	Поле	Тип группировки	+
		— Товар	Без иерархии	24
				+
	<			
	<<			
		Диаграммы:		
		Поле	Тип группировки	+
	5			24
				+
	<			
	<<			
< Назад	ļ	laлее > ОК	Отмена Справи	ka

Рис. 7.28. Окно для определения серий и точек диаграммы

Формирование отчета по поступлениям с итоговой информацией

Рассмотрим еще один из вариантов отчета по поступлениям товаров, в котором будут реализованы следующие требования:

- отчет будет организован в виде списка;
- определение пользователем интервала по дате для отбора информации;
- включение в отчет итоговых показателей по каждому товару;
- упорядочивание информации в отчете.

Приступим к созданию нового отчета, который назовем Отчет7_5. Начальные шаги традиционны — щелчок по кнопке Открыть схему компоновки данных и добавление нового набора данных (запроса). В окне конструктора запроса перенесем таблицу ИнформацияОТоварах.Обороты и выберем из нее три поля таблицы: Организация, Товар и КоличествоПриход (см. рис. 7.26). Для того чтобы данные в отчете были упорядочены необходимым образом, следует сделать установки на закладке **Порядок** (рис. 7.29). Теперь можно щелкнуть по кнопке **OK**, что позволит сформировать запрос на языке запросов (листинг 7.3).

Листинг 7.3. Запрос для получения информации о поступивших товаров

ВЫБРАТЬ

ИЗ

```
ИнформацияОТоварахОбороты.Организация КАК Организация,
ИнформацияОТоварахОбороты.Товар КАК Товар,
ИнформацияОТоварахОбороты.КоличествоПриход КАК КоличествоПриход
```

РегистрНакопления.ИнформацияОТоварах.Обороты

КАК ИнформацияОТоварахОбороты

УПОРЯДОЧИТЬ ПО

Товар,

Организация,

КоличествоПриход

Ē	🖥 Конструктор	запроса *								×
	Таблицы и поля	Группировка	Условия	Дополнительно	Объединения/Псевд	Порядок	Компоновка данных	Характе	ристин	ки
	Поля			Поле		Сортир	овка		+	^
	🕀 🚍 Все поля	a		;Товар		Возрас	тание			
				Организация		Возрас	тание			
				КоличествоПр	риход	Возрас	тание			
				`						- 1
			>>							- 11
										- 11
										\sim
	Запрос				< Назад Дал	iee >	ОК Отмен	a Ci	правка	•

Рис. 7.29. Установки на закладке Порядок

В создаваемый отчет вместе с детальной информацией необходимо обеспечить включение итоговых показателей по поступлению каждого товара. Для этого перейдем на закладку **Ресурсы** и выберем в качестве ресурса количествоПриход (рис. 7.30).

Теперь осталось перейти на закладку **Настройки**, где следует создать группировку по товарам, включающую вложенную группировку для отображения детальных записей (рис. 7.31).

На этом разработка отчета в режиме конструктора завершена. В режиме 1С:Предприятие сформированный отчет по имеющимся данным представлен на рис. 7.32. Получившийся отчет весьма информативен. Так, мы видим — сколько всего различных товаров поступило, при этом в случае необходимости можно увидеть детализацию по конкретным поставщикам.

Однако в такой редакции разработки в отчетную информацию попадают сведения обо всех поступлениях. Реально же часто требуется просмотреть информацию в определенном периоде по дате. Для этого в нашем отчете системой автоматически предусмотрены параметры:

- Начало периода;
- Конец периода.

🚵 Конфигуратор (учеб	ная версия) - Конф	игурация		. 🗆 🛛		
🛛 <u>Ф</u> айл <u>П</u> равка Конфигура	ция Отладка Админис	трирование <u>С</u> ервис <u>О</u> к	на Справка			
I 🗋 🛋 🖬 🐰 🕒 🛍	🍯 🗟 🖕 🕹 🐥	₩ <mark>*</mark>				
I 🔄 🖳 🕋 🔳 🖡 🥊						
💐 Отчет Отчет7_5: Осно	внаяСхемаКомпонов	экиДанных1		_ 🗗 🗙		
Наборы д Связи наб	Вычисляе Ресурсь	і Параметры Макеты	Вложенны Н	Настройки		
Доступные поля	Поле	Выражение	Рассчитывать г	10		
– КоличествоПриход	🔪 🗕 КоличествоП	ри Сумма(Количество				
🗕 Организация						
🗕 Товар	>>					
	<					
				~		
🖏 Отчет Отчет7_5 🛛 👌: ОсновнаяСхемаКомпон						
Для получения подсказки нажмите F1 САР NUM ги -						

Рис. 7.30. Установки на закладке Ресурсы

🚵 Конфигуратор (учебная версия) - Конфигурация	
🛛 🛨 айл Правка Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
I 🗋 🛁 🐰 🖻 🖷 🧉 🖄 🖕 I 🤀 💥 I	
	» ▼
💐 Отчет Отчет7_5: ОсновнаяСхемаКомпоновкиДанных1 📃 🖞	P ×
Наборы да Связи набо Вычисляем Ресурсы Параметры Макеты Вложенные Настр	ойки
☐-Щ Отчет ☐-У == Товар └У == <Детальные записи>	₩
Настройки: Отчет Параметры Выбранные Отбор Сортировка Условное оф Пользовател Другие наст	r
Доступные поля Поле 9	불
В Воранизация В В Организация	è
🖽 📲 КоличествоПриход 🔽 Товар .	•
В. В. СистемныеПоля ✓ В. В. КоличествоПриход	- 11
В. ПараметрыДанных	•
	1
для получения подсказки нажмите г і САР NOM 10.▼	11

Рис. 7.31. Установки на закладке Настройки

Для того чтобы выполнить их установку, следует щелкнуть по пиктограмме **Настройки** (см. рис. 7.32) и далее перейти на закладку **Параметры данных**. Например, можно указать даты в соответствии с рис. 7.33.

🏪 1C:	🏪 1С:Предприятие (учебная версия) - Конфигурация 💦 🔲 🔀												
∏ <u>Ф</u> айл	🛛 <u>Ф</u> айл <u>П</u> равка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка												
I 🗋 🛓	I 🗋 🛁 🕌 🔏 🗠 🖕 🐥 📰 📰 🗮 🗮 🗶 м. м+ м- 🎘 🖕												
🕸 O1	🔮 Отчет Отчет7 5 🛛 💶 🗙												
Дейст	гвия - 🕨 Сформироват	ь 📑 🔨 Конструктор	о настроек	Настройки	» ▼								
	Товар	Количество			^								
	Организация	приход											
P	Кухня Уют-З	19											
	Тико	2											
	Юниор	17											
P	Мягкая мебель	45											
	Тико	5											
	Юниор	40											
P	Стенка (мод. 15-7)	35											
	Тико	23											
L	Юниор	12											
	Итого	99											
政 Отч	💷 Отчет Отчет71 🔹 Отчет Отчет72 💷 Отчет Отчет73 💷 Отчет Отчет75												
Для по	лучения подсказки нажмите	• F1		CAP NUM	Для получения подсказки нажмите F1 САР NUM								

Рис. 7.32. Отчет по поступлениям товаров с группировкой данных

🛃 Конфигуратор (учебная версия) - Конфи	пурация						
🛛 🛨 айл Правка Конфигурация Отладка Админист	трирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка						
🗋 🔟 📕 🕺 🐴 👘 🥌 🖕 🕁 🖻 🌆							
Ĩ 🔩 🐥 Ĩ Ж <i>К</i> Ц 🖽 • 💁 • 🗛 • 💆 • ≡		» •					
💐 Отчет Отчет7_5: ОсновнаяСхемаКомпонов	киДанных1 .	_ @ ×					
Наборы да Связи набо Вычисляем Ресур	осы Параметры Макеты Вложенные Н	астройки					
⊟•∰ Отчет ⊡•⊽≕ Товар ≟⊽≕ <Детальные записи>		₩ ₩					
Настройки: Отчет Параметры Выбранные Отбор Сортиров	зка Условное оф Пользовател Другие	наст					
Параметр	Значение Дата						
Иначало периода	Произвольная дата 14.10.2009 0:00:00						
Конец периода	Произвольная дата <mark>ЗО ТО 2009 0:00:00</mark>	ð					
🌐 Отчет Отчет7_5 🌏: ОсновнаяСхемаКомпон							
Для получения подсказки нажмите F1	CAP NUM	ru •					

Рис. 7.33. Задание параметров отчета на закладке Настройки

ୱ 1С:Предприятие (учебная версия) - Конфигурация 💦 🔲 🔲 🔀										
<u>∎</u> айл	<u>Ф</u> айл <u>П</u> равка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка									
I 🗋 🖕	D 🗅 🖬 X 🖻 🛍 🚔 Q 5 – 🐉 🎆 🛅 🌊 M. M+ M- 🎘 ↓									
🕸 От	🕼 Отчет Отчет75 💶 🗸 🗌 🗙									
Дейст	вия 🗸 🕨 Сформироват	гь 🕴 🐴 Конструктс	р настроек	Настро	йки	» ▼				
	Товар	Количество								
	Организация	приход								
P	Кухня Уют-З	19								
	Тико	2								
L	Юниор	17	_							
P	Мякая мебель	45								
	Тико	5	-							
	Юниор	40								
日日	Стенка (мод. 15-7)	35								
	Тико	23	-							
L	Юниор	12				-				
	Итого	99								
💷 Отче	🗈 Отчет Отчет71 📑 Отчет Отчет72 📑 Отчет Отчет73 📑 Отчет Отчет75									
Для пол	учения подсказки нажмит	e F1		CAP	NUM					

Рис. 7.34. Отчет по интервалу дат

Теперь если сформировать отчет (рис. 7.34), то мы получим сведения только по тем поступлениям, которые ограничены указанными датами.

Формирование в отчете двух таблиц

Рассмотрим еще один вариант отчета (Отчет7_6), когда вместе с детализированной информацией в отчет выводятся итоговые показатели по товарам. Отличием от предыдущего примера (см. рис. 7.34) будет формирование двух таблиц. При создании нового отчета следует выполнить шаги, аналогичные тем, которые были выполнены при разработке отчета Отчет7_5. Изменения коснутся закладки **Настройки** (рис. 7.35).

В режиме 1С:Предприятие созданный отчет представлен на рис. 7.36. Мы видим здесь две таблицы — отдельную с детализацией товаров по организациям и общую с итоговыми показателями.

Отчет по документам Заявка

Разработаем отчет, который будет включать в себя информацию по имеющимся заявкам на товары. Данный отчет (Отчет7_7) предназначен для включения в него перечня имеющихся в базе данных документов Заявка.

В окне конструктора запроса нам потребуется таблица заявка (рис. 7.37), из которой для запроса необходимо отобрать следующие поля:

- ♦ Номер;
- 🔶 Дата;
- 🔶 ФирмаЗаказчик.

📩 Конфигуратор (учебная версия) - Конфигурация	×
🗄 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка	
i 🗋 🔟 🖌 🗞 📾 🥌 🖄 🖕 🖕 🗶 🙀 🙀 👘 👘	» ▼
₩ <mark></mark> , ЖКЧ	» ▼
👌 Отчет Отчет7_6: ОсновнаяСхемаКомпоновкиДанных _ 🗖 🗴	×
Наборы Связи на Вычисля Ресурсы Парамет Макеты Вложенн Настройки	и
🗗 🗗 🕎 OTVET 📑 📑	
Герерание записи> Фетальные записи> Фетальные записи> Фетальные записи> Фетальные записи> Фетальные записи> Фетальные записи>	
Настройки: Отчет Отчет	
Параметр Выбранны Отбор Сортировка Условное Пользоват Другие на	
Доступные поля Поле 🔤	
П В Панизация В Выбранные поля	
В Повар V По	
🕀 🖿 СистемныеПол 🔽 🛄 КоличествоПриход	
🗄 📲 ПараметрыДа	
🗊 Отчет Отчет7_6 😽: ОсновнаяСхемаКомпон	
Для получения подсказки нажмите F1 САР NUM гит	//

Рис. 7.35. Установки на закладке Настройки

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🗖 🔀									
🛛 <u>Ф</u> айл <u>П</u> равка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка									
[] 🔟 🔲 🐰 🖻 🕼 😂 🖸 🖕 🚆 🎆 🗐 🧝 м м+ м- 🎘 🖕									
🖺 Отчет Отчет7 6 💶 🗸 🔲									
Действия - 🕨 Сформир	овать Сонстр	уктор настроек	Настройки						
Товар	Организация	Количество приход		^					
Кухня Уют-З	Тико	2							
Мяткая мебель	Тико	5							
Стенка (мод. 15-7)	Юниор	12							
Кухня Уют-З	Юниор 1			H					
Стенка (мод. 15-7)	Тико	23							
Мяткая мебель	Юниор	40							
Итого		99							
Товар	Количество приход								
Кухня Уют-З	19								
Мяткая мебель	45								
Стенка (мод. 15-7)	35								
Итого	99								
				\sim					
<				>					
			05						
💷 Отчет О 💷 Отчет О	. 🗓 Отчет О 🗓	Отчет О 💷 Отч	ет О 💷 О	тчет О					
Для получения подсказки на	ажмите F1		CAP N	UM //					

Рис. 7.36. Отчет с двумя таблицами для детальной и групповой информации

🎬 Конструктор запроса *								□ ×
Таблицы и по Группировка Услови	ия До	ополнительно	Объединения	Π	орядо	к Комі	поновка	Характерист
E 🐝		월 👞 🛬	K S			열 -	× *	
База данных		Таблицы		^		Поля		
⊡	>	🖃 🐺 Заяв	ка		>		Заявка.Но	мер
🕂 📥 Документы	>>		сылка		>>		Заявка. Да	ana
🗄 🐷 Поступление			ІометкаУдаления			L	Заявка.Фі	ирмаЗаказчик
🖪 🕀 🐨 🐷 Заявка		H	омер					
🗄 🐨 🐷 Отгрузка		Д	lata					
🗄 🚰 РегистрыНакопления	<	Π	роведен		<			
	11	🗼 🖷 🖕 🙀	^р ирмаЗаказчик					
		Ē ≣ ∐	ереченьТоваров					
		I I I I I I I I I I I I I I I I I I I	редставление					
			оментВремени	~				
κ >								
3anpoc		< Hasa	ад Далее >			JK 🔤	Отмена	а Справка

Рис. 7.37. Отбор полей для запроса при формировании отчета по документам

Далее перейдем на закладку **Порядок** и укажем, что результат запроса должен быть упорядочен по убыванию дат документов (рис. 7.38).

_									
Ē	Конструкт	ор запроса							×
	Таблицы	Группиров	Условия	Дополнит	Объедине	Порядок	Компонов	Характери	
	Поля			Поле		Сорт	ировка		^
	; — Ном	ер		Дата		Убы	вание		
	— — Фир	маЗаказчик							
	🗄 📲 Bcei	поля							
			>						
			>>						~
	<							>	
	Запрос (Назад Далее > ОК Отмена Справка								3

Рис. 7.38. Установка порядка вывода результатов запроса

Других установок для запроса делать не будем, и щелчок по кнопке **ОК** приведет к формированию текста необходимого запроса (листинг 7.4).

```
Листинг 7.4. Запрос для получения перечня документов
```

ВЫБРАТЬ

```
Заявка.Номер,
Заявка.Дата КАК Дата,
Заявка.ФирмаЗаказчик
ИЗ Документ.Заявка КАК Заявка
УПОРЯДОЧИТЬ ПО Дата УБЫВ
```

В запросе используется объектная таблица документ.Заявка. После формирования текста запроса перейдем на закладку **Настройки** и создадим новую группировку в структуре отчета. Укажем, что в группировке будут выводиться детальные записи из информационной базы, и перенесем необходимые поля в раздел **Выбранные поля** (рис. 7.39).

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🔀								
<u>Ф</u> айл <u>П</u> равка Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка								
🗋 🔟 📓 🔏 📾 🖆 🖕 📾 🛤 📃 🔽 🕷 🐥								
₩ [*] ж к ц ⊡ • <u>≫</u> • <u>A</u> • <u>/</u> • ≡ ≡ ≡ ∅								
💐 Отчет Отчет7_7: ОсновнаяСхемаКомпоновкиДанных 💶 🗆 🗙								
Наборы Связи н Вычисл Ресурсы Параме Макеты Вложен Настро								
4*								
Настройки: Отчет Отчет								
Выбранные Отбор Сортировка Условное оф Пользовател Другие наст								
Доступные поля Поле 😋								
Дата 🕒 Выбранные поля								
— Номер 🖌 — Цата 🖀								
Ш ш та тирмазаказ (♥) помер ш ш та тирмазаказчик								
🕽 Отчет Отчет7_7 🛛 🛃: ОсновнаяСхемаКомпон								
Для получения подсказки нажмите F1 САР NUM ru▼								

Рис. 7.39. Установки на закладке Настройки для получения информации о документах

🧏 1С:Предприятие (учебная версия) - Конфигурация 💦 🖃 🔀										
<u>Ф</u> айл <u>П</u> равка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка										
I 🗋 🔟 🚽 X 🖻 🛍 😅 🔯 5 🐥 I 🎹 🛅 🧝 M M+ M- 🎘 🖵										
🕎 Отчет Отчет7 7	📑 Отчет Отчет77 💶 🗕 🗆									
Действия 🗸 🕨 Сформиј	оовать 📑 🔨 Ко	нструктор настр	оек Настройки 🍟							
Дата	Номер	Фирма заказчик								
08.09.2008 0:00:00	000000002	Тико								
03.09.2008 11:13:49	00000003	Меда								
19.08.2008 21:34:44	000000001	Юниор								
[4]			×							
Отчет Отчет? 7 Лоя подучения подсказки нажинте F1 САР NUM										

Рис. 7.40. Формирование отчета по заявкам

Далее осталось перейти в режим 1С:Предприятие и сформировать отчет по имеющимся документам Заявка в информационной базе (рис. 7.40).

Отчет по заявкам

Рассматриваемый в данном разделе отчет позволит получить информацию о количестве позиций заявок по фирмам-заказчикам. Новый объект конфигурации будет называться — отчет7_8. В окне конструктора запроса выберем две таблицы:

- объектную таблицу справочника Фирмы;
- виртуальную таблицу ИнформацияОЗаявках.Обороты.

Теперь из данных таблиц в раздел Поля перенесем три поля (рис. 7.41):

- 🔶 Фирмы.Ссылка;
- ИнформацияОЗаявкахОбороты.Товар;
- ♦ ИнформацияОЗаявкахОбороты.КоличествоОборот.



Рис. 7.41. Окно конструктора запроса

В отличие от предыдущих примеров этой главы в запросе мы используем две таблицы, которые необходимо связать. Ранее при рассмотрении запросов мы использовали различные варианты соединения таблиц, а также синтаксические конструкции языка запросов, позволяющих это реализовать. В данном случае подобные установки реализуются визуальными действиями. Так, на закладке Связи следует убедиться в установке необходимой связи между двумя таблицами (рис. 7.42). Значение измерения фирма регистра накопления должно быть равно ссылке на элемент справочника фирмы. Обязательно требуется проверить установку флажка Все у таблицы справочника. В этом случае из справочника будут выбраны все элементы справочника, и этим элементам будет поставлено в соответствие значение оборота по данной организации из регистра. Таким образом, в результате запроса в отчете будут присутствовать все фирмы, а для ряда из них (которые делали заявки на товары) будут указаны значения оборотов.

_		
Ī	Конструктор запроса	×
	Таблицы и поля Связи Группировка Условия Дополнительно Объединения/П Порядок Компоновка да Характеристики 😪 🖶 🥃 🍋	
	№ Таблица 1 Все Таблица 2 Все П Условие связи 1 Э Фирмы У МнформацияОЗаявкахОбороты П У ИнформацияОЗаявкахОбороты.Фирма = Фирмы.Ссылка	лка
	Запрос Казад Далее ОК Отмена Справка]

Рис. 7.42. Установка связей между двумя таблицами в запросе

Перейдем на закладку Объединения/Псевдонимы, где установим псевдоним Организация вместо Фирмы. Ссылка (рис. 7.43), т. к. при работе в пользовательском режиме слово Организация намного информативнее.

🎬 Конструктор запроса *		□ ×				
Таблицы и поля Связи Груп	пировка Условия Дополнительно Объединения/	Псев Порядок Компоновка данных Характеристики				
월월월 11 -	1 I					
Имя Бездубл.	Имя поля	Sanpoc 1				
Запрос 1	——— Организация	···· Фирмы.Ссылка				
	Товар	Информация0Заявках0бороты. Товар				
	КоличествоОборот	Информация0Заявках0бороты.Количество0				
Запрос	< Назад	Далее > ОК Отмена Справка				

Рис. 7.43. Закладка Объединения/Псевдонимы

На закладке **Порядок** укажем (рис. 7.44), что результат должен быть отсортирован сначала по фирмам, а затем по значению поля количествоОборот в порядке убывания.

🛙 Конструктор заг	ipoca *									×
Таблицы и поля Св	язи Группиров	ка Ус	ловия	Дополнительно	Объединения/Псев	Порядок	Компоновка данных	Характер	оистик	ч
Поля			Поле			Сортиров	<a< td=""><td></td><td>+</td><td>^</td></a<>		+	^
у 🛶 Товар			·····0	уни-Организация Возрастание			•	2.1		
🗄 📲 Все поля			КоличествоОборот			Убывание			+	
		>								
		>>								
		<								
		~~								\sim
Запрос					< Назад Да	nee >	ОК Отмена	a Cn	равка	1

Рис. 7.44. Установка порядка вывода результатов запроса

На этом визуальные действия по конструированию запроса завершены, и осталось щелкнуть по кнопке **ОК**, чтобы получить текст запроса (листинг 7.5).

Листинг 7.5. Запрос для получения информации о сделанных заказах

```
ВЫБРАТЬ

Фирмы.Ссылка КАК Организация,

ИнформацияОЗаявкахОбороты.Товар,

ИнформацияОЗаявкахОбороты.КоличествоОборот КАК КоличествоОборот

ИЗ

Справочник.Фирмы КАК Фирмы

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ИнформацияОЗаявках.Обороты КАК

ИнформацияОЗаявкахОбороты

ПО (ИнформацияОЗаявкахОбороты.Фирма = Фирмы.Ссылка)

УПОРЯДОЧИТЬ ПО

Организация,

КоличествоОборот УБЫВ
```

В данном случае в запросе используется уже знакомое левое соединение таблиц, о котором мы говорили в *главе* 6.

Перейдем на закладку **Параметры** (рис. 7.45), где уже указаны два параметра. Это связано с тем, что система компоновки данных самостоятельно анализирует текст запроса и указывает те параметры, которые, скорее всего, будут использоваться в отчете. Присутствующие здесь началоПериода и КонецПериода являются параметрами виртуальной таблицы РегистрНакопления. Информация ОЗаказах. Обороты. Первый параметр представляет начало периода расчета итогов, второй — конец периода. В результате исходная таблица будет содержать обороты, рассчитанные в переданном интервале дат.

📩 Конфигуратор (учебная версия) - Конфигурация							
🗄 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка							
I 🗋 🖬 😹 📾 📾 🔯 🖕 🖬 🛤 🛛 🐥 🐥 📜 🖓 🌪 I							
1 🐜 🎼 🕋 🖬 🕨 🗸							
💐 Отчет Отчет7_8: ОсновнаяСхемаКомпоновкиДанных1							
Наборы данных Связи наборов д Вычисляемые поля Ресурсы Параметры Макеты Вложенные схемы	Настройки						
월번철 1 ↓							
Имя Заголовок Доступные типы Доступные зна Д Значение Выражение	B O						
НачалоПериода Начало периода Дата							
КонецПериода Конец периода Дата							
💷 Отчет Отчет7_8 🛛 🔊: ОсновнаяСхемаКомпон							
Для получения подсказки нажмите F1 САР NUM ги т							

Рис. 7.45. Закладка для определения параметров отчета

Важно отметить, что тип данных дата кроме непосредственно даты содержит еще и время. Разумеется, при указании интервала дат присутствие времени является излишним. Для необходимой коррекции (исключения конкретного времени в указании даты) нужно дважды щелкнуть в графе Доступные типы параметра началопериода и с помощью пиктограммы с изображением многоточия открыть окно Редактирование типа данных (рис. 7.46). Здесь в качестве состава даты необходимо выбрать вариант — дата.



Рис. 7.46. Окно для редактирования типа данных

Особенностью работы системы является то, что по умолчанию время в дате установлено 00:00:00. В связи с этим, если пользователь задаст период расчета, скажем, до 20.02.08, то данные на этот день (время создания которых позже 00:00:00) в расчет не войдут. Поэтому при задании интервала по дате при формировании отчета это надо учитывать.

В [9] приведен пример, в котором с помощью программных конструкций реализуется включение в результат информации по указанной дате.

Перейдем теперь на закладку **Настройки**, где добавим группировку без указания поля группировки. Здесь же на дополнительной закладке **Выбранные поля** (рис. 7.47) укажем поля Организация, Товар и КоличествоОборот.

Теперь уже можно перейти в режим 1С:Предприятие и посмотреть отчет в работе (рис. 7.48).

Дополним отчет установкой интервала по дате. Для этого с помощью кнопки **Настройки** перейдем к установке настроек отчета и установим интервал по дате (рис. 7.49).

Теперь, зафиксировав настройки с помощью кнопки **ОК**, уже можно сформировать необходимый отчет в интересующем интервале дат.
🚵 Конфигуратор (учебная версия) - Конфигурация	l ×
II <u>Ф</u> айл <u>П</u> равка Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
I □ 	
Ĩ₩₩₩ ₩₩ ₩₩₩₩₩₩ <u>₩</u> ₩ <u>₩</u> ₩₩	•
💐 Отчет Отчет7_8: ОсновнаяСхемаКомпоновкиДанных1 🗕 🗖	×
Наборы данных Связи наборов д Вычисляемые поля Ресурсы Параметры Макеты Вложенные схемы Настро	йки
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	
	×
Настройки: Отчет Отчет Параметры данных Выбранные поля Отбор Сортировка Условное оформ Пользовательск Другие настройи	ки
Доступные поля Поле — КоличествоОборот ⊟Выбранные поля — Организация ✓ — Товар ✓ — СистемныеПоля ✓ — ПараметрыДанных ✓	TLTI TLTI
т тур Отчет Отчет7_8 😵: ОсновнаяСхемаКомпон	
Для получения подсказки нажмите F1 САР NUM ги т	///

Рис. 7.47. Настройки отчета по заказам

🏪 1С:Пред	приятие (учебная версия)	- Конфигурация
<u>∎</u> айл <u>П</u> ра	авка Таблица Операции <u>С</u> ер	рвис <u>О</u> кна Сп <u>р</u> авка
I 🗋 🛋 🖬	🗶 🖻 🛍 🛋 🖸 🗅) 🖻 🐘 🧳 📰 🛅 🧟 M M+ M- 🎇 🗸
政 Отчет	Отчет7 8	_ 🗆 ×
Действия 🕶	🕨 🕨 Сформировать 📑	Конструктор настроек Настройки 🔄 🖳 🤶
Организа	ация Товар	Количество 🔨
Меда	Стенка (мод. 15-7)	7
Тико	Мякая мебель	5
Тико	Стенка (мод. 15-7)	3
Юниор		
Лада		
		M
🗒 Отчет От	чет 78 🔛 🚵 Докул	иенты Отгрузка
Для получен	ия подсказки нажмите F1	CAP NUM

Рис. 7.48. Отчет по заказам в работе

🖤 Настройки отчета Отчет7 8					□ ×
ि-∰ Отчет ⊽ === <Детальные записи>					³ ³ ³ ³ ³ ³ ³ ³ ³
Настройки : Отчет Отчет Параметры да Выбранные поля	Отбор Сортировка	Условное офо	Пользователь	Другие нас	стро
		Значение	Лата		
✓ Начало периода		Произвольная дата	27.05.2008 0:	00:00	- S
🗸 Конец периода		Произвольная дата	24.07.2009 0:1	00:00	2
					ð
				OK	Отмена

Рис. 7.49. Настройка интервала по дате в отчете по заказам

Отчеты по информационной базе главы 5

Перейдем теперь к рассмотрению конфигурации, разработанной в *главе 5*. В ней была предложена возможность формирования квитанций для оплаты за обучение с использованием формы ввода. В связи с этим последующие действия переносятся в конфигурацию, разработанную в *главе 5*.

Отчет по оплаченным квитанциям

Для развития созданной в *главе 5* разработки добавим отчет по оплаченным квитанциям. Начнем создание нового отчета (Отчет7_9 на рис. 7.50). В окне конструктора запроса (рис. 7.51) воспользуемся таблицей Квитанции (из ветки РегистрыСведений). В раздел Поля перенесем:

- ♦ Договор;
- СуммаПоКвитанции;
- ♦ МесяцНачала;
- ♦ ГодНачала;
- МесяцОкончания;
- ♦ ГодОкончания.

Далее перейдем на закладку Условия и добавим условие, связанное с полем оплачено (рис. 7.52).

На закладке **Порядок** установим порядок сортировки сначала по договорам, а затем по началу интервала оплаты (рис. 7.53). Теперь визуальное проектирование

запроса завершено, и можно нажать кнопку **ОК**, что приводит к формированию запроса (листинг 7.6).

🛃 Конфигуратор (учебная версия) -	- Конфигурация
	-	•
🕸 Отчет Отчет7_	9	_ 🗆 ×
• Основные	Имя:	Отчет7 9
Данные	C	
Формы	Синоним:	UT4et7 9
Макеты	Комментарий:	
Подсистемы		
Права		
Интерфейсы		
Прочее	Основная суем:	
	OCHOBINA CACING	
		UсновнаяLхемаКомпоновкиДанны × ч
1		Открыть схему компоновки данных
Действия	• КНазад	далее> закрыть Справка
🛄 Отчет Отчет7_9	💷 Отчет О	Ітчет7_9st 🗒 Отчет Отчет7_8
Для получения подск	азки нажмите F1	CAP NUM ru -

Рис. 7.50. Окно редактирования объекта конфигурации



Рис. 7.51. Визуальное конструирование запроса

Листинг 7.6. Запрос для получения информации о квитанциях

ВЫБРАТЬ

```
Квитанции.Договор КАК Договор,
Квитанции.СуммаПоКвитанции,
```

```
Квитанции.МесяцНачала КАК МесяцНачала,
Квитанции.ГодНачала КАК ГодНачала,
Квитанции.МесяцОкончания,
Квитанции.ГодОкончания
ИЗ
РегистрСведений.Квитанции КАК Квитанции
ГДЕ
Квитанции.Оплачено = &Оплачено
УПОРЯДОЧИТЬ ПО
Договор, ГодНачала, МесяцНачала
```

🎬 Конструктор запроса	•				□ × □
Таблицы и Группиров	Условия Дополн	ит Объедине	Порядок	Компонов	Характери
Поля ⊞~¶ Квитанции					^
	1 Квита	ие нции.Оплачено = &О	плачено		
<					Σ
Запрос	< Назад	Далее >	OK	Отмена	Справка

Рис. 7.52. Добавление условия для отбора квитанций

🎬 Конструктор запроса *						□ ×
Таблицы и Группировка Усл	ловия	Дополните	Объединен	Порядок	Компоновк	Характери
Поля		Поле		Сортирови	(a	^
🐖 🥼 СуммаПоКвитанции		Договор		Возрастан	ние	
— МесяцОкончания	ГодНачала		Возрастание			
ГодОкончания	МесяцНачала		Возрастание			
🗄 🖷 🚍 Все поля						
	>					
	>>					
		1				×
Запрос	Γ	< Назад	Далее >	OK	Отмена	Справка

Рис. 7.53. Добавление сортировки

Нас интересуют оплаченные квитанции, поэтому на закладке Параметры следует установить значение истина для поля Оплачено (рис. 7.54).

Осталось выполнить установки на закладке Настройки (рис. 7.55), и отчет готов (рис. 7.56).

🛃 Конфигура	тор (учебная вер	сия) - Конфигурация				
💐 Отчет Отч	іет7_9: Основная	СхемаКомпоновкиДа	нных			_ 🗆 ×
Наборы даннь	их Связи наборов д	анных Вычисляемые г	оля Ресурсы Парами	етры Макеты	Вложенные схемы	Настройки
철 털 철	t I					
Имя	Заголовок	Доступные типы	Доступные знач Д 3	Значение	Выражение	B Or
Оплачено	Оплачено	Булево		истина		
📑 Отчет Отчет	7_9	7 Отчет Отчет7_9st	📑 Отчет Отчет7	_8	💐: ОсновнаяСхе	маКомпон
Для получения	подсказки нажмите	F1			CAP NU	JM ru≁ //

Рис. 7.54. Установка значения для параметра Оплачено

📩 Конфигуратор (учебная версия) - Конфигурация	
🛛 🕂 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка	
Ĩ₩, [*] ĨЖКЧ • <u>*</u> • <u>*</u> •] ≡ ≡ ≡ ∭	»
💐 Отчет Отчет7_9: ОсновнаяСхемаКомпоновкиДанных	_ 🗆 ×
Наборы да Связи наб Вычисляе Ресурсы Параметры Макеты Вложенны	Настройки
⊟-Ш Отчет 	₩- ₩ ₩ - ×
Настройки: Отчет Отчет Параметры Выбранные Отбор Сортировка Условное о Пользовате Др Доступные поля Поле ГодНачала Поле Поле Поле Поле Поле Поле Поле Поле Поле Поле Поле Поле Поле Полезовате Др Поле Полезовате Др Поле Полезовате Др Поле Полезовате Др Поле Полезовате Др Поле Полезовате Др Поле Посе Пос	угие наст • • • •
— СуммаПоКвитан 🗹 — ГодОкончания	
Ш… ЦистемныеПоля VI : = СуммаПоКвитанции Ш… ПараметрыДанн	
💷 Отчет Отчет7_9 🛛 👌: ОсновнаяСхемаКомпон	
Для получения подсказки нажмите F1 САР NU	M ru▼ //

Рис. 7.55. Установки на закладке Настройки

🏪 1С:Предп	риятие (уч	ебная веро	сия) - Конфигу	урация	E				
<u>Ф</u> айл <u>П</u> равка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка									
I 🗋 🛋 🖬	🗋 🛄 🚽 🔏 🛍 🚔 🙆 🖕 😋 🗛 👘 🎽 📰 📰 🧝 м м+ м- 🞇 🖕								
関 Отчет Отч	чет7 9				_	. 🗆 🗙			
Действия 🕶	🕨 Сформир	овать 🔤	Конструктор наст	роек Настрой	іки 🖳 📑 🛛	?			
Договор	Месяц	Год	Месяц	Год	Сумма по				
	начала	начала	окончания	окончания	квитанции				
123/34	8	2 007	10	2 007	6 000				
123/34	11	2 007	1	2 008	6 000				
123/77	7	2 007	7	2 007	1 970				
123/77	8	2 007	9	2 007	4 300				
123/77	10	2 007	10	2 007	2 150				
123/77	11	2 007	1	2 008	6 450				
123/77	2	2 008	6	2 008	11 000				
<						×			
 Отчет Отчет Для получения г 	79 подсказки на:	жмите F1			CAP	NUM			

Рис. 7.56. Отчет по оплаченным квитанциям

Отчет по договорам

Составим отчет для получения информации об итоговой оплате по договорам. Имя отчета выберем — отчет7_10. Начальные шаги традиционны для данной главы, и первые важные шаги сделаем в окне конструктора запроса (рис. 7.57). В данном случае нам для отчета достаточно одной таблицы — регистра сведений Квитанции.

🎬 Конструктор запроса		□ ×
Таблицы и поля Группировка Условия इ. 👯	Дополнительно Объединения/Псевд 🔄 🔩 🎽 🖬 🔍 Таблицы	Порядок Компоновка данных Характеристики
 ⊕-Э Справочники ⊖- РегистрыСведений ⊕-• Тарифы ⊕-• Квитанции ⊕-• Квитанции.СрезПервых ⊕-• Квитанции.СрезПоследних 	 Квитанции Период Договор СуммаПоКвитанции Оплачено МесяцНачала ГодНачала КосяцОкончания 	> Квитанции.Договор >> Квитанции.СуммаПоКвитанции
Запрос	< Назад Да	лее > ОК Отмена Справка

Рис. 7.57. Окно конструктора запроса

На закладке **Группировка** укажем в качестве группового поля квитанции. Договор (рис. 7.58), а на закладке **Условия** добавим условие для поля Оплачено. В результате мы получим текст запроса (листинг 7.7).

🎬 Конструктор	запроса									□ ×
Таблицы и поля	Группировка	Условия	Допо	олнительно	Объединения/Г	Псевд	Порядок	Компоновка данны	ых Характер	ристики
Поля				Групповое	поле					^
⊞~≣ Все пол	A		>	👗 Кві	итанции. Договор					
			>>							
			<							
			<<							
				-						
				Суммируем	юе поле				Функция	
			_	🚺 Кв	итанции.Cymmal li	оКвитанц	ии		Сумма	
			>>							
			<							
<										>
2					(1 1			04 0		
запрос					< Назад	Дал	iee >		ена Сп	равка

Рис. 7.58. Выбор поля для группировки в запросе

Листинг 7.7. Запрос для получения информации об оплате

ВЫБРАТЬ

```
Квитанции.Договор,
```

```
СУММА (Квитанции.СуммаПоКвитанции) КАК СуммаПоКвитанции
```

ИЗ

```
РегистрСведений.Квитанции КАК Квитанции
```

ГДЕ

```
Квитанции.Оплачено = &Оплачено
```

🛃 Конфигурато	р (учебная верс	:ия) - К	онфигурация					_		ĸ
🛛 <u>Ф</u> айл <u>П</u> равка	Конфигурация От	тладка	Администриров	ание	<u>С</u> ервис !	<u>О</u> кна (Справк	a		
I		•								
💐 Отчет Отчет	г7_10: Основная	аСхема	КомпоновкиД	аннь	ях					
Наборы данных	Связи наборов да	анных	Вычисляемые п	оля	Ресурсы	Парам	иетры	Макеты	Вло	ж
월 달 집 1	1.4									
Имя	Заголовок	Дост	упные типы	Дост	упные зна	чД	Значе	ние		В
Оплачено	Оплачено	Буле	80			-	истин	a		Π
💷 Отчет Отчет7_1	10 📑	7: Осн	ювнаяСхемаКом	пон						
Для получения по	дсказки нажмите I	F1					CAP	NUM ruh	•	11

Рис. 7.59. Задание значения параметра запроса

📩 Конфигуратор (учебная версия) - Конфигурация	<u>_ ×</u>
🛛 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка	
💐 Отчет Отчет7_10: ОсновнаяСхемаКомпоновкиДанных 🗕	
Наборы данн Связи наборо Вычисляемы Ресурсы Параметры Макеты Вложенные с Нас	тройки
	1
	-
	*
Настройки: Отчет Отчет	
Параметры да Выбранные по Отбор Сортировка Условное офо Пользователь Другие нас	πро
Доступные поля	1 😑 📗
Потовор	. 📜 📗
— Сумманоквитанции V — Договор	
Ште ПараметрыДанных	1
	+
💷 Отчет Отчет7_10 🛛 💩: ОсновнаяСхемаКомпон	
Для получения подсказки нажмите F1 САР NUM	ru • //

Рис. 7.60. Установки на закладке Настройки

Теперь в окне схемы компоновки данных на закладке **Параметры** выберем значение истина для параметра Оплачено (рис. 7.59).

Осталось выполнить стандартные установки на закладке **Настройки** (рис. 7.60), после чего можно перейти в режим 1С:Предприятие и получить готовый отчет (рис. 7.61).

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🔍
📱 🕂 айл 🏾 Правка Таблица Операции Сервис 🛛 кна Справка
🛯 🗋 📮 🐰 🖻 🛍 🧔 💝 📲 🛅 餐 м м+ м- 🎘 🖕
🖤 Отчет Отчет7 10 🛛 💶 🗙
Действия 🗸 🕨 Сформировать 🦄 Конструктор настроек 🎇
Договор Сумма по квитанции 123/34 12 000 123/77 25 870
👜 Отчет Отчет710
Для получения подсказки нажмите F1 САР NUM //

Рис. 7.61. Отчет по оплате договоров

Отчет по квитанциям с группировкой

Рассмотрим создание отчета, представленного на рис. 7.62. Он позволяет увидеть как суммарную оплату, так и детализацию оплат за обучение.

🏪 1C:I	Предприятие	е (учебная версия)	- Конфигу	рация		
<u>Ф</u> ай)	п <u>П</u> равка Та	блица Операции <u>С</u> е	рвис <u>О</u> кн	а Сп <u>р</u> авка		
	🛋 🖬 🔏 🛙	h 🖻 🛋 🗟 🗅	c m	» ▼ ■ ■	🙎 🛛 м м	1+ M- 🎇 🖵
🕸 O1	гчет Отчет7	11				_ 🗆 ×
Дейст	гвия 🕶 🕨 Со	формировать 🛛 📩 К	онструкто	р настроек	Настройки	🖻 🕺 👗
	Договор	Период		Сумма по квитанции		^
	Итого 123/34			12.0	00	
ΠΥ	120/04	10.08.2007 17:46:0)7	60	00	
		27.08.2008 8:58:45	5	6.0	00	
Q.	123/77			25 87	70	
		10.08.2007 15:22:1	2	1 97	70	
		30.08.2007 12:58:1	5	4 30	00	
		30.08.2007 12:59:4	18	21	50	
		12.09.2007 8:25:47		64	50	
L		12.09.2007 8:47:31		11 0	00	
						~
<						>
💷 Отч	ет Отчет711					
Для по	лучения подск	азки нажмите F1			CA	P NUM

Рис. 7.62. Отчет с включением группировки и детальных записей

Учитывая уже сформированный опыт работы с системой компоновки данных, мы не будем детально описывать этапы создания отчета, а лишь поясним ключевые шаги, которые необходимо выполнить. Так, на рис. 7.63 показано окно конструктора запроса, который нам потребуется создать.



Рис. 7.63. Окно конструктора запроса

В качестве условия отбора данных, как и ранее, будем учитывать только оплаченные квитанции, а на закладке **Настройки** выполним установки в соответствии с рис. 7.64. После этого в режиме 1С:Предприятие мы получим готовый отчет (см. рис. 7.62), и для его формирования потребовались минимальные действия с нашей стороны.

📩 Конфигуратор (учебная версия) - Конфигурация	_ 🗆 ×
🛛 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка	
I D 🛋 🖬 X 🖻 🛍 🛋 🙆 🖕 C 💙 I 🏵 ♥ I	
₩ <mark>*</mark> ж к ч • • <u>*</u> • <u>A</u> • <u>Z</u> • ≡ ≡ ≡ 0	»
💐 Отчет Отчет7_11: ОсновнаяСхемаКомпоновкиДанных	_ 🗆 ×
Наборы д Связи наб Вычисляе Ресурсы Параметры Макеты Вложенны П	Настройки
□-₩ <mark>Отчет</mark> □-♥━= Договор └♥━= Период	
Настройки: Отчет Отчет	
Параметры выоранные Отоор Сортировка эсловное о Пользовате Друг	ие наст
Доступные поля	- 1
政 Отчет Отчет7_11 🔹: ОсновнаяСхемаКомпон	
Для получения подсказки нажмите F1 САР NUM	1 ru <i>▼ //,</i>

Рис. 7.64. Закладка настроек отчета

Отчеты по информационной базе главы 1

Рассмотрим еще несколько примеров отчетов, которые можно создать с использованием системы компоновки данных. Они будут касаться конфигурации, разработанной в *главе 1*. В ней был предложен учет слушателей учебных курсов. В связи с этим последующие действия переносятся в конфигурацию, разработанную в *главе 1*.

Отчет по учащимся

На рис. 7.65 приведен отчет, который позволяет просмотреть как общие оплаченные суммы, так и увидеть детализацию сделанных оплат. Для начала нам потребуется с помощью конструктора запросов сформировать необходимый запрос (рис. 7.66). Здесь мы воспользуемся документом ВнесениеОплаты и его табличной частью. В листинге 7.8 приведен текст данного запроса, который позволит извлечь необходимую информацию.

🏪 1C:	Предприятие (учебная вер	сия) - Конфигурация						
<u>Ф</u> айл	🛛 🛨 айл Правка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка							
	[] 🔄 🔔 🔛 🖄 🚔 🙆 🖕 😋 😫 👘 🦉 🔛 🦉							
🕸 01	🔍 Отчет. Отчет с системой компоновки 🛛 _ 🗆 🗙							
Дейст	твия • Сформировать	*Конструктор настроек	Настройки	h 🕂 ?				
	Клиент		Внесенная	^				
	Курс	Дата	сумма					
	Николаев Сергей Федорович		1 700					
	Базы данных	31.08.2008 12:59:08	1 000					
	Базы данных	31.08.2008 12:59:49	700					
P	Сидоров Евгений Петрович		3 250					
	Базы данных	31.08.2008 11:48:15	500					
	PowerPoint	31.08.2008 11:48:15	700					
	Информатика (Excel)	31.08.2008 11:48:15	1 200					
	Информатика (Excel)	17.07.2009 11:52:55	700					
	Базы данных	17.07.2009 11:52:55	50					
L	PowerPoint	17.07.2009 11:52:55	100					
	Итого		4 950					
				×				
<				>				
政 Отч	ет. Отчет с системой ко 💷 (Этчет Отчет с системой ко						
Для по	олучения подсказки нажмите F	1		CAP NUM				

Рис. 7.65. Отчет по учащимся

🎬 Конструктор запро	са								□ ×
Таблицы и поля Связи	и Группировка	Условия	Дополнительно	Объединени	я/Псе	вдони	Порядок	Компоновка данны	х Характеристики
長 徽		얼 🛃 🎗	E 🙀 🔍			얼굴	, × =		
База данных		Таблицы				Поля			
🕀 🗐 Справочники			есениеОплаты				ВнесениеО	платы.Клиент	
🕂 🛁 Документы	>	⊡≣ B⊦	есениеОплатыОп.	патаКурсов	>		ВнесениеО	платыОплатаКурсов.	Курс
	>>	÷… —	Ссылка		>>		ВнесениеО	платыОплатаКурсов.	ВнесеннаяСумма
			НомерСтроки			···· _	ВнесениеО	платы. Дата	
		÷	Курс						
		····	ВнесеннаяСумма	э					
	<				<				
	11				11				
<									>
3anpoc				< H	азад	Дa	алее >	ОК Отме	на Справка

Рис. 7.66. Окно конструктора запроса

Листинг 7.8. Запрос для получения информации о слушателях курсов

ВЫБРАТЬ

ВнесениеОплаты.Клиент, ВнесениеОплатыОплатаКурсов.Курс, ВнесениеОплатыОплатаКурсов.ВнесеннаяСумма, ВнесениеОплаты.Дата Документ.ВнесениеОплаты.ОплатаКурсов КАК ВнесениеОплатыОплатаКурсов ЛЕВОЕ СОЕДИНЕНИЕ Документ.ВнесениеОплаты КАК ВнесениеОплаты ПО ВнесениеОплатыОплатаКурсов.Ссылка = ВнесениеОплаты.Ссылка

Осталось выполнить установки на закладке Настройки (рис. 7.67), и отчет готов (см. рис. 7.65).

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🗵
<u>Ф</u> айл <u>П</u> равка Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
▝▙▝▙▝▆▝▝▖▝▓▕▝▖▝▓ヽ▙ヽZヽ▏▆▝▖▆▕▓▏
💐 Отчет ОтчетССистемойКомпоновки: ОсновнаяСхемаКомпоновкиДанных 💶 🗖 🗙
Наборы данных Связи наборов д Вычисляемые поля Ресурсы Параметры Макеты Вложенные схемы Настройки
⊡·І́́́́́́́́́́́́́́́́́́́́́́́́́́́́́́́́́́́́
Настройки: Отчет Отчет Выбранные поля Отбор Сортировка Условное оформление Пользовательские поля Другие настройки Доступные поля — Дата — Ала — Клиент — Клиент — Курс — Вибранные поля — Кирс — Кирс — Кирс
Для получения подсказки нажмите F1 САР NUM ги т

Рис. 7.67. Окно настроек отчета

Отчет по курсам

На рис. 7.68 приведен отчет, который позволяет просмотреть информацию по оплатам за курсы (видно, кто и сколько внес денежных средств). Для начала нам

ИЗ

потребуется с помощью конструктора запросов сформировать необходимый запрос (рис. 7.69). Запрос аналогичен тому, который нам встретился в предыдущем разделе.

🏪 1С:П	редприятие (учебная версия) - Кон	фигурация		<u>_ D ×</u>			
∏ <u>Ф</u> айл	Правка Таблица Операции <u>С</u> ервис	<u>О</u> кна Сп <u>р</u> авка					
I 🗋 🚄	🗋 🛁 📓 🔏 📾 🖾 🖸 🖕 🖒 👌 🖒 🗰 🧮 🎬 🎬 🎬 🌌 🛛 M M+ M- 🎇 🧅						
🕸 От	чет Отчет713			_ 🗆 ×			
Дейст	вия 🗸 🕨 🕨 Сформировать 🛛 🛝 Конст	груктор настроек.	Настройки	h 1 ?			
	Күрс	Внесенная		^			
	Клиент	сумма					
0 P	PowerPoint	800					
l I	Сидоров Евгений Петрович	800					
P	Базы данных	2 250					
	Сидоров Евгений Петрович	550					
L	Николаев Сергей Федорович	1 700					
P	Информатика (Excel)	1 900					
L	Сидоров Евгений Петрович	1 900					
	Итого	4 950					
政 Отчет Отчет7 12 👜 Отчет Отчет7 13							
Для пол	учения подсказки нажмите F1			CAP NUM			

Рис. 7.68. Отчет по оплатам за курсы

🌃 Конструктор запроса			x
Таблицы и поля Связи Группировк База даннык Ваза даннык Сравочники Славочники ВнесениеОплаты ВнесениеОплаты Внесения	а Условия Дополнительно Объединен Таблицы — ВнесениеОплатыОплатаКурсов — Ссылка — НомерСтроки — Курс — ВнесеннаяСумма	ния/Псевдони Порядок Компоновка данных Характеристи Поля ВнесениеОплатыЮплатаКурсов.Курс ВнесениеОплаты.Клиент Клиент	CK
Запрос	< +	Назад Далее> ОК Отмена Справка	9

Рис. 7.69. Окно конструктора запроса

Выполним установки на закладке Настройки (рис. 7.70), после чего отчет готов (см. рис. 7.68).

Таким образом, в этой главе мы рассмотрели систему компоновки данных, которая является мощным ресурсом для создания отчетов. Однако сказать, что система компоновки данных приводит к полному вытеснению других технологий формирования отчетов, нельзя. Реально в большинстве ситуаций она обеспечивает существенную эффективность, однако в ряде случаев удобнее воспользоваться уже знакомыми возможностями без использования такого сложного инструмента автоматизации.

🛃 Конфигуратор (учебная версия) - Конфигурация	- O ×
📱 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка	
I 🗋 🖬 🐰 🖻 💼 🙆 🖕 d 🔍 🦓 🦓 I	
Ĭ₩ [*] ₩ K Ҷ - <u>*</u> • <u>A</u> • <u>A</u> • <u>F</u> = = =	»
💐 Отчет Отчет7_13: ОсновнаяСхемаКомпоновкиДанных	_ 🗆 ×
Наборы д Связи наб Вычисляе Ресурсы Параметры Макеты Вложенны 🥤	Настройки
B-V == Kypc 	
Настройки: Отчет Отчет	
Выбранные поля Отбор Сортировка Условное оформ Пользовательск Другие н	настройки
Доступные поля Поле	n 🔁 📗
Парана Кира Парана Кира	
на прости прос	
Ш… ПистемныеПоля ✓ Ш	
	_↓
💷 Отчет Отчет7_12 💷 Отчет Отчет7_13 🛃: ОсновнаяСхемаКомпи	он
Для получения подсказки нажмите F1 САР NUM	1 ru <i>▼ ///</i>

Рис. 7.70. Окно настроек отчета

глава 8



Автоматизация работы с данными

К настоящему моменту мы разобрали большое количество практических примеров, дающих представление о технологии разработки конфигураций для платформы 1С:Предприятие 8.1. В предыдущих главах мы работали с такими объектами конфигурации, как перечисления, справочники, документы, отчеты, регистры сведений и регистры накопления. Рассмотрев большое количество программных процедур, мы познакомились с объектами системы 1С:Предприятие, с их свойствами и методами. Большое внимание было уделено подчиненным объектам конфигурации — формам, табличным частям, макетам. Также мы рассмотрели систему компоновки данных. Все это должно сформировать у читателей цельное представление о разработке прикладных решений на платформе 1С:Предприятие 8.1.

Однако в данную систему заложены практически неисчерпаемые ресурсы для разработчиков, и к настоящему моменту уже выпущено несколько изданий, раскрывающих секреты мастерства профессионального разработчика [1, 2, 4, 7—9]. В данной главе мы рассмотрим ряд технических возможностей, которые представляются наиболее интересными в практическом плане.

Большое внимание будет уделено построению разнообразных отчетов и обмену информацией с популярными офисными приложениями (Microsoft Word и Microsoft Excel). Ранее в одном из примеров была рассмотрена возможность внесения изменений с помощью процедуры на встроенном языке 1С в уже имеющийся документ, созданный в приложении Microsoft Word. Здесь данная тема получит развитие — мы познакомимся с рядом ситуаций, в которых требуется интеграция различных офисных приложений. Изучение программных процедур позволит читателям успешно справиться и с решением собственных аналогичных задач.

В *главе* 5 мы рассматривали задачу учета оплаты за обучение в организации, предоставляющей платные образовательные услуги. Можно сказать, что тематика разработок данной главы будет в определенной степени похожа.

Учет начислений преподавателям

Первая задача, которую мы рассмотрим, будет связана с необходимостью учета начислений преподавателям за проведенные занятия. Известно, что во всех учеб-

ных заведениях и центрах подготовки существует несколько видов (категорий) занятий. Мы ограничимся для определенности лекциями, практическими занятиями и консультациями (могут быть и другие, но в последующем примере остановимся на таком наборе). Эти занятия, как правило, по-разному тарифицируются (под тарификацией подразумевается часовая стоимость занятия для преподавателя). В этом случае преподаватель, проведя некоторое количество часов учебной нагрузки определенного вида, должен получить сумму, являющуюся результатом умножения количества часов на тариф. Для этого в информационной базе следует учитывать различные виды занятий, проведенные сотрудниками. Разрабатываемые в данной главе объекты конфигурации, как и прежде, будут предназначаться для ввода исходных данных, а также получения разнообразной итоговой информации по работе учебного центра.

Хотя тематика разработки сохраняется, но мы при создании конфигурации основной акцент сделаем на особенностях, которые в предыдущих разделах еще не встречались.

Информация о тарифах

Начнем создание новой конфигурации, которая будет включать в себя необходимые объекты и программные процедуры, что в итоге обеспечит автоматизацию учета в учебном центре. Нам понадобится описать сведения о тарифах (их названиях и стоимости). Для этого мы создадим два объекта конфигурации — перечисление ВидыЗанятий и регистр сведений тарифы.

На рис. 8.1 показано окно редактирования объекта конфигурации для работы с перечислением ВидыЗанятий.

🛃 Конфигуратор (учебная версия)	- Конфигурация	_ 🗆 🗙				
<u>Ф</u> айл <u>П</u> равка Ко	нфигурация Отлад	ка Администрирование	<u>С</u> ервис <u>О</u> кна				
Справка							
🗋 🛋 🖬 🐰 🛛	à 🔒 🧉 🗟	5 C 🗿 🛤	» •				
🔛 Перечисление	ВидыЗанятий		_ 🗆 ×				
• Основные	Има	Bugu 2sumuč					
Данные	KIMDA.	Бидыранятии					
Формы	Синоним:	Виды занятий					
Макеты	Комментарий:						
Подсистемы							
Деиствия 👻	назад Дал	ее>Закрыть	Справка				
Для получения подск	азки нажмите F1	CAP	NUM ru -				

Рис. 8.1. Закладка Основные окна редактирования перечисления ВидыЗанятий

Значения, составляющие перечисление ВидыЗанятий, являются названиями тарифов, и следует заполнить закладку Данные так, как показано на рис. 8.2.



Рис. 8.2. Значения перечисления ВидыЗанятий

Теперь для тарифов занятий (различные виды занятий по-разному тарифицируются) мы создадим периодический регистр сведений тарифы. Это название уже встречалось в *главе 5*, но поскольку новая конфигурация независима, какой-то путаницы здесь возникать не должно. Ранее подразумевалась тарификация оплаты за обучение, а сейчас рассматривается тарификация оплаты преподавателей.

На практике тарифы с течением времени меняются. В нашем случае будем считать, что они изменяются не чаще одного раза в месяц. Чтобы отразить это, надо в окне редактирования объекта конфигурации (рис. 8.3) выбрать в списке **Периодичность** значение **В пределах месяца**. Здесь же следует для режима записи в регистр установить вариант **Независимый**. После этого можно перейти на закладку **Данные** и определить измерения и ресурсы регистра (рис. 8.4). Создадим одно измерение — ВидЗанятия (тип данных — ПеречислениеСсылка.ВидыЗанятий). Другим измерением регистра является Период, что связано со сделанной нами установкой для поля **Периодичность**, но он создается автоматически. В качестве ресурса примем — ВеличинаТарифа (тип данных — Число).

Из конфигуратора перейдем в режим 1С:Предприятие и в автоматически созданной экранной форме внесем в регистр сведений информацию о тарифах (рис. 8.5).

Примечание

Возможна ситуация, когда требуется организовать изменения в тарифах чаще, чем ежемесячно. В этом случае следует просто изменить значение периодичности в окне редактирования регистра сведений.

Конфигуратор (учебная версия) - Конфигурация Файл Правка Конфигурация Отладка Администрирование Сервис Окна						
I Справка	6 5 5	c 🕅 M 🔹 🗸				
🥤 Регистр сведений	і Тарифы	_ 🗆 ×				
 Основные Данные 	Имя:	Тарифы				
Регистраторы	Синоним:	Тарифы				
Формы	Комментарий:					
Макеты		·				
Подсистемы						
Права	Периодичность:	В пределах месяца 💌				
Интерфейсы	Режим записи:	Независимый				
Обмен данными Прочее	Обмен данными Прочее Основной отбор по периоду 🗸					
Действия - (Назад Далее) Закрыть Справка						
Для получения подсказк	и нажмите F1	CAP NUM ru •				

Рис. 8.3. Закладка Основные окна редактирования регистра сведений Тарифы

🛃 Конфигуратор (учебн		
🛛 <u>Ф</u> айл <u>П</u> равка Конфигуј	рация Отладка Администриро	вание <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
I 🗋 🛋 🖬 🐰 🖻 🖻	1 🛋 🗟 5 C 🖪 A	🔹 🕅 🖍 🐂 🍟
🥤 Регистр сведений 1	Гарифы	Свойства: Измерение 🛛 🕈 🗙
Основные		₽↓ 🛃 🎦 🗙 🗸
🕨 Данные	8881+8	Имя ВидЗанятия
Регистраторы	⊡ — Измерения	Синоним Вид занятия
Формы	Вид Занятия	Комментарий
Макеты	ВеличинаТарифа	
Подсистемы	Реквизиты	Сеновной от Г
Права		Запрет незаг
Интерфейсы		
Обмен данными		Индексирова Не индексировать 💌
Прочее		Полнотексто Использовать 🔻
		▼Типданных:
		Тип ПеречислениеСсылка.ВидыЗан 💌 💆
Действия 👻	<Назад Далее>	Имя объекта метаданных
Для получения подсказки н	нажмите F1	

Рис. 8.4. Закладка Данные окна редактирования регистра сведений Тарифы

Этап подготовительной работы по созданию необходимых объектов конфигурации еще не завершен, и перед разработкой программных процедур нам потребуется несколько вспомогательных справочников.

🏪 1С:Предприятие (уче	<mark>陽</mark> 1С:Предприятие (учебная версия) - Конфигурация 💶 🗙					
🛛 <u>Ф</u> айл <u>П</u> равка Операци	и <u>С</u> ервис <u>О</u> кна Сп	равка				
I 🗋 🛋 🖬 🛛 🕹 🖻	🚆 📰 🖹 🧣	M M+ M- 🎇 🗸				
🥤 Список Тарифы		_ 🗆 ×				
Действия 🕶 🔄 🚽 式	🎽 🔜 🔛 🏹 🏾	🛅 🖷 - 🏹 📘 🖉 🐥				
Период 🔹	Вид занятия	Величина тарифа \land				
01.08.2008	Консультации	145				
01.08.2008	Лекции	190				
01.09.2008	Практика	180				
01.09.2008	Консультации	150				
01.09.2008	Лекции	200				
🖌 Список Тарифы						
Для получения подсказки н	нажмите F1	CAP NUM				

Рис. 8.5. Внесение данных в регистр сведений Тарифы в режиме 1С:Предприятие

Вспомогательные справочники

Во-первых, необходимо создать справочник преподаватели, предназначенный для перечисления фамилий (с инициалами) преподавателей нашего учебного центра. Этот объект конфигурации не будет содержать дополнительных реквизитов, и каких-то особенностей при работе с формой справочника нам не потребуется. Один из вариантов заполненной формы списка, генерируемой по умолчанию, приведен на рис. 8.6.



Рис. 8.6. Заполнение справочника Преподаватели в режиме 1С:Предприятие

Другой справочник, который мы назвали группы, отражает названия учебных групп (рис. 8.7). В нем кроме кода и наименования в качестве реквизита добавлено еще и количество — для отражения количества учащихся в каждой группе. Как и при заполнении первого справочника, мы используем форму списка, генерируемую системой по умолчанию. На рис. 8.8 приведен один из вариантов заполнения справочника с названиями групп.

🛃 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🗙							
🖩 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка							
D 🛋 🖬 X 🖻 f	i 🛋 🖸 i to i di 🕅 🖉	H4	- 1	R R			
菌 Справочник Группы				_ & ×			
Основные	Длина кода	Свойства: Ре	квизит	 			
Иерархия		21 🛐 🍯 🗆	X V				
Владельцы	— Тип кода — — Тип	•Основные:		^			
🕨 Данные	ОЧисло	Имя	Количество				
Нумерация	© Cranera	Синоним	Количество				
Формы	• строка	Комментарий					
Макеты	열롱철수부		_				
Подсистемы	Количество	Использован	Для элемента				
Ввод на основании		Индексирова	Не индексироват	ъ			
Права		Полнотекстов	Использовать				
Интерфейсы	<u> </u>	▼Тип данных	-				
	🛄 Табличные часті	Тип Число		▼			
	ействия 🔹 <Назад	Синоним объек	та метаданных	¥			
Для получения подсказки нажмите F1							

Рис. 8.7. Закладка Данные окна редактирования справочника Группы

🏪 1С:Предприятие	(учебная версия) - К	онфигурация 📕	
🛛 <u>Ф</u> айл <u>П</u> равка Опер	рации <u>С</u> ервис <u>О</u> кна (Справка	
🗋 🚄 🔐 🐰 🖻	🔒 🌷 🔳 🗎	🜠 🛛 м м+ м-	₩
🗐 Справочник Гру	ппы	_	
Действия 🛛 🔤 불	🛃 😼 🕅 To 🖷	- 🌾 🖸 ?	
Код	Наименование 🔹	Количество	^
- 000000001	07-ДВ-1		20
- 00000002	07-ДВ-2		25
- 000000003	07-ФМ-1		30
			~
и Поправочник Группы			
Для получения подска:	зки нажмите F1	CAP NU	M /

Рис. 8.8. Заполнение формы списка справочника Группы в режиме 1С:Предприятие

Документ Учет

После создания и заполнения информацией справочников перейдем к разработке документа для фиксирования начисленных преподавателям сумм за проведенные занятия. Для этого в режиме конфигуратора создадим новый объект конфигурации — документ учет. В области шапки разместим только один дополнительный реквизит датаРасчета, который будет использоваться для внесения информации о дате, на которую производится расчет начисляемых сумм. Сами суммы будут размещаться в табличной части документа. Это связано с тем, что дата создания документа и дата расчета могут отличаться, а тарифы в нашей ситуации привязаны к дате.

Использование реквизита датаРасчета позволит нам в дальнейшем (в программной процедуре) извлекать из регистра сведений тарифы информацию о величине тарифа на тот месяц, который указан в этом реквизите.

Для отражения основной информации об оплате в окне редактирования объекта конфигурации (рис. 8.9) необходимо создать табличную часть (с именем Список) и разместить в ней несколько реквизитов:

- Преподаватель (информация о преподавателе, тип данных СправочникСсылка.Преподаватели);
- Группа (информация о группе, тип данных СправочникСсылка. Группы);

🚵 Конфигуратор (учебная ве	ерсия) - Конфигурация 📃 🔲 🗙
<u>Ф</u> айл <u>П</u> равка Конфигурация	Отладка Администрирование <u>С</u> ервис <u>О</u> кна
Справка	
I 🗋 🛋 🖬 🐰 🌇 🚳 📾	i 🖻 🖕 🖬 🛤 🛛 🚆 🍟
🐱 Документ Учет	_ 🗆 ×
Основные	包装灌土工具
▶ Данные	
Нумерация	ДатаРасчета
Движения	
Последовательности	
Журналы	
Формы	· · · · · · · · · · · · · · · · · · ·
Макеты	⊟•
Подсистемы	— III Список
Ввод на основании	— Преподаватель
Права	пруппа
Интерфейсы	— видзанятия — ЧислоЧасов
Обмен данными	— Тариф
Прочее	Сумма
Действия 👻 <Назад	Далее> Закрыть Справка
Для получения подсказки нажми	те F1 CAP NUM ru т

- ВидЗанятия (ТИП Данных ПеречислениеСсылка.ВидыЗанятий);
- Числочасов (ТИП данных число);
- Тариф (ТИП данных Число);
- Сумма (ТИП Данных Число).

В связи с тем, что в дальнейшем нам придется реагировать на интерактивные действия пользователя в программе, форма документа, генерируемая по умолчанию, здесь нам не подойдет. Поэтому несложными техническими действиями создадим пользовательскую форму документа (рис. 8.10).

📩 Конфигуратор (учебная версия) - Конфигурация	<u>- 0 ×</u>
🛛 🙅 айл Правка Форма Конфигурация Отладка Администрирование Сервис Окна Справка	
I 🗋 💶 X 🖻 🛍 🗉 🔯 🖕 🖆 📾 א 📃 🖓 🖓 👘 🎘 ?	an →
🚟 Документ Учет: ФормаДокумента	□ ×
Учет	\sim
Действия - 🛃 📧 🔩 🚰 🛃 🕐 💭	
Номер:	
Дата:	
Дата расчета:	
알클ુ볼 및 철 문 한 분 CD	
N Преподаватель Группа Вид занятия Число часов Тариф Сумма	
аы <Поле ввода>	оле в
	>
С.) ОК Записать За	Крыть
П	

Рис. 8.10. Форма документа Учет

Процедуры обработки событий на форме документа

В форму документа нам необходимо заложить удобный интерфейс для работы пользователя. Поэтому перейдем к программированию событий, связанных с расположенными на форме элементами управления. Но сначала поясним, как будет выглядеть в целом процесс заполнения данного документа пользователем в режиме 1С:Предприятие.

Сотрудник учебного центра при создании нового документа типа учет сначала заполняет расположенное в области шапки поле Дата расчета, которое определяет, на какую дату мы собираемся производить расчет начислений преподавателям. После этого вносится информация в табличную часть. При вводе очередной строки прежде всего необходимо заполнить поле Преподаватель (для этого автоматически открывается список преподавателей из соответствующего справочника). Затем заполняется поле Группа, для чего также используется соответствующий справочник. Для ввода данных в следующее поле ВидЗанятия также выбирается значение из ранее созданного перечисления.

После заполнения этих трех полей требуется реализовать определенную автоматизацию. А именно, при выборе вида занятия в поле тариф должно автоматически подставляться значение тарифа на дату, ближайшую к дате расчета (точнее, на дату не позже даты расчета). Для этого мы должны оформить соответствующим образом обработку события приизменении для поля ВидЗанятия. В листинге 8.1 приведена программная процедура, реализующая поиск и подстановку нужных данных.

Листинг 8.1. Процедура заполнения поля тариф при выборе вида занятия

```
Процедура СписокВидЗанятияПриИзменении(Элемент)
ТабличнаяЧасть = ЭлементыФормы.Список.ТекущиеДанные;
СтруктураОтбора = Новый Структура;
СтруктураОтбора.Вставить("ВидЗанятия", ТабличнаяЧасть.ВидЗанятия);
Ресурс = РегистрыСведений.Тарифы.
ПолучитьПоследнее(ДатаРасчета,СтруктураОтбора);
ТабличнаяЧасть.Тариф = Ресурс.ВеличинаТарифа;
КонецПроцедуры
```

Для отбора ближайшего к дате расчета значения тарифа используется метод получитьПоследнее(), который требует некоторого комментария. Вообще для получения информации о значении ресурса регистра сведений можно использовать метод получить(). В этом случае мы можем выбрать конкретное значение ресурса, соответствующее набору измерений (в данной ситуации у нас измерение одно) и значению периода. Например, это можно выполнить с помощью следующего программного кода:

Ресурс = РегистрыСведений. Тарифы. Получить (ДатаРасчета, СтруктураОтбора)

Использование подобной конструкции вполне допустимо, но при условии, что для каждого месяца в регистр сведений занесено значение тарифа. Однако в реальной ситуации в течение продолжительного времени тарифы могут не меняться, и в регистр сведений их повторно не вносят. В связи с этим, если на указанный месяц (ДатаРасчета) информации в регистре сведений нет, то при использовании метода Получить () мы выберем пустое значение. Метод ПолучитьПоследнее () позволяет извлечь из регистра последний по времени имеющийся тариф (не позднее указанной в качестве параметра даты) для выбранного вида занятий. Это как раз нам и требуется для рассматриваемой задачи.

Таким образом, в процедуре, представленной в листинге 8.1, вносится правильное значение тарифа на дату расчета, и от пользователя требуется ввести только количество отработанных часов. После этого на форме документа (в табличной части) должна автоматически рассчитываться сумма, которую следует начислить преподавателю. Для этого требуется соответствующим образом оформить процедуру обработки события приизменении для поля числочасов. В листинге 8.2 представлена эта несложная процедура.

Листинг 8.2. Процедура для расчета поля Сумма

```
Процедура СписокЧислоЧасовПриИзменении(Элемент)
ТабличнаяЧасть = ЭлементыФормы.Список.ТекущиеДанные;
ТабличнаяЧасть.Сумма = ТабличнаяЧасть.Тариф*ТабличнаяЧасть.ЧислоЧасов;
КонецПроцедуры
```

Итак, программирование формы выполнено. Теперь можно перейти в режим 1С:Предприятие и заполнить несколько документов учет. Один из вариантов заполнения формы документа показан на рис. 8.11. При практических действиях обеспечивается запланированный сервис — значения тарифов на дату расчета заполняются автоматически, а суммы рассчитываются после указания пользователем количества отработанных часов.



Рис. 8.11. Заполненная форма документа Учет в режиме работы 1С:Предприятие

Однако процедура установки тарифов (см. листинг 8.1) не позволяет реализовать функциональность интерфейса в случае, если при заполненной табличной части мы изменим дату расчета. В этом случае необходимо обеспечить автоматический пересчет информации табличной части документа. А именно, должны быть пересчитаны значения тарифов и сумма по каждому преподавателю. Для реализации такого функционала следует оформить соответствующим образом процедуру (листинг 8.3), выполняемую при изменении в поле датаРасчета.

Листинг 8.3. Процедура для перерасчета табличной части

```
Процедура ДатаРасчетаПриИзменении (Элемент)
Для Каждого ТекСтрокаСписок Из Список Цикл
СтруктураОтбора = Новый Структура;
СтруктураОтбора.Вставить ("ВидЗанятия", ТекСтрокаСписок.ВидЗанятия);
Ресурс = РегистрыСведений.Тарифы.
ПолучитьПоследнее (ДатаРасчета, СтруктураОтбора);
ТекСтрокаСписок.Тариф = Ресурс.ВеличинаТарифа;
ТекСтрокаСписок.Сумма = ТекСтрокаСписок.Тариф*ТекСтрокаСписок.ЧислоЧасов;
КонецЦикла;
КонецПроцедуры
```

Теперь, если перейти в режим 1С:Предприятие, можно убедиться, что данные в табличной части документа пересчитываются при изменении даты расчета (рис. 8.12).

🏪 1С:Предпри	ятие	(учебная	версия	а) - Конфигу	рация			- D ×
🛛 <u>Ф</u> айл <u>П</u> равка	Опер	рации <u>С</u> ер	вис <u>О</u> н	жна Сп <u>р</u> авка				
I 🗋 🚢 🔛	XÈ	6 6		b e l a	» •	📰 🗎 🧝	M M+	м- 🎇 🖕
🐱 Учет: Учет	(Нов	ый) *						_ 🗆 ×
Действия 🕶 🔤	0	🔄 🔁		?				
Номер:	00000	00001						
Дата:	12.09	.2008 0:00	:00					
Дата расчета:	07.08	.2008				E		
월 달 등 3	5	(† †		2				
N Преподав	атель	Группа	В	ид занятия	Число часов	Тариф	Сумма	
📕 🚺 Бородин В	3.A.	07-ДВ-2	ſ	екции	3	3 190	כ	570
2 Синицын л	4.B.	07-ФM-1	K	онсультации	45	5 140)	6 300
3 Акатьев Д	д.Ю.	07-ДВ-2	ſ	екции	10) 190	ו	1 900
						OK	Записать	Закрыты
🚬 Документы Учет 🛛 🐱 Учет: Учет (Новый) * 🌱 Список Тарифы								
Для получения п	Для получения подсказки нажмите F1 САР NUM //							

Рис. 8.12. Заполненная форма документа Учет в режиме работы 1С:Предприятие

Просмотр актуальных тарифов

Документ уже достаточно функционален, но сервисные доработки формы пока не завершены. Наша цель в данном разделе — организовать просмотр информации по тарифам. Для этого добавим на форму два элемента управления — кнопку и поле списка (рис. 8.13). Эти элементы управления в данном случае предназначены для того, чтобы пользователь прямо на форме документа мог посмотреть последние тарифы (на указанную дату расчета) по различным категориям занятий. При нажатии кнопки **Последние тарифы** поле списка заполняется информацией из базы данных.

🍌 Конфигуратор (учебная версия) - Конфигурация 📃	
🛛 🙅 айл Правка Форма Конфигурация Отладка Администрирование Сервис Окна Справка	
I 🗋 💶 X 🖻 🛍 📾 🖻 🕇 📽 🕅 🛤 📃 🔽 🐨 🖓 👘 🎘 🍡	• *
🚟 Документ Учет: ФормаДокумента	
Учет	
Действия - 🛃 💽 🍇 ? 💭	
Номер: Последние тарифы	
Дата:: Ш Срез тарифов:	51
Дата расчета:	
а одаватель Группа Вид занятия Число часов Тариф Сумма	
Поле ввода> аы «Поле вв аы «Поле ввода>	
<	>
III Тії Служебные сообщения	
Для получения подсказки нажмите F1 САР NUM ru ▼ 118, 108 ↓1384 x 24	1

Рис. 8.13. Добавление элементов управления на форме документа Учет

Имена этих добавленных элементов установим следующими:

- ПоследниеТарифы (кнопка);
- СрезТарифов (поле списка).

Для реализации просмотра информации о тарифах нам потребуется написать процедуру обработки события, связанного с нажатием созданной кнопки (листинг 8.4).

Листинг 8.4. Процедура вывода информации о тарифах на дату расчета

```
Процедура ПоследниеТарифыНажатие (Элемент)
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ Срез.ВидЗанятия КАК Вид,
| Срез.ВеличинаТарифа КАК Величина
| ИЗ РегистрСведений.Тарифы.СрезПоследних(&ДатаОкончания)
| КАК Срез";
Запрос.УстановитьПараметр("ДатаОкончания", ЭлементыФормы.ДатаРасчета.Значение);
Результат = Запрос.Выполнить().Выбрать();
СрезТарифов.Очистить();
```

```
Пока Результат.Следующий() > 0 Цикл
СрезТарифов.Добавить (Строка (Результат.Вид) +
" " + Строка (Результат.Величина) + " руб.");
КонецЦикла;
КонецПроцедуры
```

Приведенная процедура содержит некоторые новые детали и поэтому требует небольшого комментария. В системе 1С:Предприятие 8.1 для регистров сведений существует метод Срезпоследних (), который возвращает таблицу значений, заполненную информацией о последних актуальных значениях ресурсов регистра.

В связи с тем, что получение среза последней по дате информации встречается достаточно часто, разработчиками системы 1С:Предприятие была включена возможность использования виртуальной таблицы для получения такого среза.

В данном случае вместо запроса к основной таблице регистра сведений мы воспользовались запросом к виртуальной таблице:

ИЗ РегистрСведений. Тарифы. СрезПоследних (&ДатаОкончания)

Примечание

Виртуальная таблица среза последних значений измерений не хранится в базе данных, а строится в момент обращения к ней.

Таким образом, наш документ (рис. 8.14) почти готов к использованию, и осталось лишь организовать автоматический учет денежных сумм по данным докумен-

🏪 1С:Предпри	ятие (у	чебная версия)	- Конфигураци	я					IX
<u>∎ Ф</u> айл <u>П</u> равка	Опера	ции <u>С</u> ервис <u>О</u> кн	на Справка						
I 🗋 🚢 🔚 ,	X 🖻	6 5 0 1	d 8		» ₹	1 📰 🛅 🌠	M 1	м+ м- 🅬	- 3
🐱 Учет: Учет	r 00000	0001 от 12.09.2	2008 18:33:37 *			-		_ 🗆	×
Действия 🕶 🔤		😼 i 🔁 🛃 i 🤇	0						
Номер:	000000	001				Послед	ние тарі	ифы	
Дата:	12.09.2	008 18:33:37		I		Срез тарифов	:		
Дата расчета:	12.09.2	008		E		Консультации	н 150 ру	16. 🤷	
엄벌로						Лекции 200	рус.	~	
N Преподав	затель	Группа	Вид занятия	Число часов		Тариф	Сумма		
Бородин	B.A.	07-ДВ-2	Лекции		5	200		10	00
2 Бородин	B.A.	07-ДВ-2	Лекции		7	200		14	00
3 Акатьев	Д.Ю.	07-ФМ-1	Консультации		10	150		15	00
						OK	Записа	ять Закры	ль
斗 Документы У	чет	🐱 Учет:	Учет 000000001	.:37 * 🎁 Спис	юк	Тарифы			
Для получения п	одсказк	и нажмите F1					C/	AP NUM	//

Рис. 8.14. Просмотр тарифов в форме документа Учет

там. Для этого, во-первых, нужно создать регистр накопления, в котором будут регистрироваться внесенные суммы. А во-вторых, требуется обеспечить движения по данному регистру при проведении документа учет.

Оборотный регистр накопления

Регистры накопления имеют две разновидности — регистры остатков и регистры оборотов. В предыдущих главах для задач учета количественной информации мы использовали и те, и другие регистры. Для регистров остатков система 1С:Предприятие создает несколько виртуальных таблиц — таблицу остатков, таблицу оборотов и совместную таблицу остатков и оборотов.

Оборотный регистр отличается от регистра остатков тем, что для него система создает только одну виртуальную таблицу — таблицу оборотов.

Для начала на закладке **Основные** в окне редактирования объекта конфигурации необходимо установить основную информацию (рис. 8.15), определяющую регистр начисление. Обратите внимание на то, что вид данного регистра — обороты. Далее на закладке **Данные** (рис. 8.16) следует ввести измерение Преподаватель (тип данных — СправочникСсылка.Преподаватели) и ресурс Сумма (тип данных — число). Работа с регистром (по записи информации по начислению денежных средств) должна быть организована с помощью документа Учет.

🛃 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🗙						
🛛 🛨 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка						
🗋 📫 🖬 🐰 🐚	6 6 1	C 🗿 🗛 📃		▼		
🛊 Регистр накоплен	ия Начисление			_ 🗆 ×		
• Основные		<u></u>				
Данные	имя:	Начисление				
Регистраторы	Синоним:	Начисление				
Формы	Комментарий:					
Макеты						
Подсистемы						
Права	Вид регистра	Обороты		-		
Интерфейсы						
Обмен данными						
Действия 👻	<Назад	Далее> Зак	срыть Сп	равка		
Для получения подсказки	и нажмите F1		CAP NUM	ru • //		

Рис. 8.15. Закладка Основные окна редактирования регистра Начисление

Движения по регистру Начисление

В режиме 1С:Предприятие работать с созданным регистром для учета начисленных сумм пока рано, поскольку механизм записи информации в регистр еще не



Рис. 8.16. Закладка Данные окна редактирования регистра Начисление

обеспечен. Дело в том, что движения по регистру должны выполняться при проведении документа учет.

В режиме конфигуратора вернемся в окно редактирования документа учет и перейдем на закладку Движения, где укажем в качестве регистра накопления для движений только что созданный регистр (рис. 8.17). Для формирования движений воспользуемся конструктором (кнопка — Конструктор движений). В результате на экране мы увидим уже знакомое окно (рис. 8.18), где от нас потребуется подставить необходимые выражения для полей регистра (наши инструкции конструктору).

После щелчка по кнопке **ОК** конструктор движений автоматически генерирует необходимый программный код (листинг 8.5).

Листинг 8.5. Процедура формирования движений по регистру Начисление

```
Процедура ОбработкаПроведения (Отказ, Режим)
Для Каждого ТекСтрокаСписок Из Список Цикл
Движение = Движения.Начисление.Добавить();
Движение.Период = Дата;
Движение.Преподаватель = ТекСтрокаСписок.Преподаватель;
Движение.Сумма = ТекСтрокаСписок.Сумма;
КонецЦикла;
// записываем движения регистров
Движения.Начисление.Записать();
КонецПроцедуры
```

Конструктор движений обеспечил завершенность документа — теперь при записи документа в информационную базу формируются движения по регистру. На

🛃 Конфигуратор (учебна: Файл Правка Конфигура	я версия) - Конфигурация ция Отладка Администриров	
Справка		
D 🐴 🖬 🕺 🛍 🛍	🧉 🗋 🖕 e 🚺 🛤	» *
🐱 Документ Учет		_ 🗆 ×
Основные Данные Нумерация • Движения Последовательности Журналы Формы	Проведение: Оперативное проведение: Удаление движений: Спорт Регистры накопли	Разрешить Разрешить Удалять движения ав ения
Макеты Подсистемы Ввод на основании Права Действия 🔻 <	Конструктор движен Начисление Назад Далее>	ний 🔄 Закрыть Справка
Для получения подсказки на:	жмите F1	CAP NUM ru -

Рис. 8.17. Закладка Движения окна редактирования документа Учет

Конструктор движен	ия регистров			□ ×
불불불↑↓				
Регистры	а Начисление	Реквизи	ты документа	
		 Ном Дата Тек0 Тек0 Тек0 Тек0 Тек0 Тек0 Тек0 	ер аРасчета СтрокаСписок.П СтрокаСписок.В СтрокаСписок.В СтрокаСписок.Ч СтрокаСписок.С	реподаватель руппа идЗанятия ислоЧасов ариф умма
Табличная часть: Спи	асок.		< <u>Н</u> азад	Далее>
Поле	Выражение		<u>З</u> аполнить	выражения
↓ Преподаватель ј Сумма	ТекСтрокаСписок.Преподавател ТекСтрокаСписок.Сумма	њ	Оуистить в	зыражения
			OK	
		Отм	Отмена	
			Спра	авка

Рис. 8.18. Окно конструктора движений документа Учет

этом работа в конфигураторе по созданию документа учет завершена. Документ готов к использованию, и, перейдя в режим 1С:Предприятие, можно сформировать (с проведением при записи) несколько документов типа учет. В результате в регистре начисление будут сформированы соответствующие записи (рис. 8.19). Каждая строка табличной части документа найдет свое отражение в регистре. Теперь можно составить отчет на основании информации, которая содержится в таблицах, формируемых системой для сохранения данных регистра Начисление.

🏪 1С:Предприятие (учебная версия) -	Конфи	гурация			
<u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна	Справк	a			
I 🗋 🛋 🖬 🐰 🐂 🛍 i 🛎 🔯 🖕	2 20	۱	» •	🗄 🛅 🧝 м. м	(+ M- 🎇 🗸
ি Список Начисление					_ 🗆 ×
Действия - 🛛 🚓 🎢 🖆 🖷 - 🌾 🛛 🔃	?				
Регистратор	Номе	Актив	Период 🔹	Преподаватель	Сумма 🛆
 Учет 00000001 от 12.09.2008 19:00: 	1	~	12.09.2008 19:00:08	Бородин В.А.	1 000
• Учет 000000001 от 12.09.2008 19:00:	2	~	12.09.2008 19:00:08	Синицын А.В.	2 400
• Учет 000000001 от 12.09.2008 19:00:	3	~	12.09.2008 19:00:08	Акатьев Д.Ю.	2 280 📃
			~	^	
					~
🚼 Список Начисление					
Для получения подсказки нажмите F1				CAF	P NUM

Рис. 8.19. Форма списка регистра Начисление

Отчет по начислению

После разработки интерфейса для электронной регистрации документов необходимо разработать для сотрудников учебного центра и итоговый отчет. Поэтому наша очередная задача заключается в создании выходной формы, на которой будет отображаться информация о начисленных суммах в течение указанного пользователем временного интервала.

В конфигураторе начнем создание нового отчета, для которого выберем имя отчетпоначислению. На закладке Макеты воспользуемся конструктором выходной формы, и в качестве таблицы для построения запроса укажем начисление. Обороты (рис. 8.20).

Пользователь при работе с отчетом должен с клавиатуры указать интервал дат. Об этом необходимо информировать конструктор. Поэтому в пиктографическом меню над разделом **Таблицы** (рис. 8.20) следует щелкнуть по пиктограмме **Параметры виртуальной таблицы**. Теперь в открывшемся диалоговом окне (рис. 8.21) необходимо задать параметры, определяющие начало и окончание интересующего нас временного интервала. Символ & перед именем параметров указывает на то, что это параметр запроса.



Рис. 8.20. Закладка Таблицы и поля окна отчета ОтчетПоНачислению

Параметры виртуальной таблицы		
Началоl Гериода	&Датаначала	
КонецПериода	&ДатаОкончания	
Периодичность	• x	
Условие		
	ОК Отмена Справка	

Рис. 8.21. Окно для выбора параметров виртуальной таблицы

На закладке **Порядок** установим упорядочивание по сумме (в порядке убывания), а на закладке **Отчет** снимем установку **Использовать построитель отчета**. Теперь осталось оформить закладку **Выходная форма** в соответствии с рис. 8.22.

Все интерактивные действия диалога с конструктором выходной формы выполнены, и осталось щелчком по кнопке **ОК** сформировать объекты и программный код, которые в совокупности реализуют в режиме 1С:Предприятие отчет по начислению. В листинге 8.6 приведены программные процедуры, обеспечивающие заполнение табличного документа на основе автоматически формируемого макета.

Листинг 8.6. Процедуры для формирования отчета по начислению

Процедура ДействияФормыОтчетПоНачислениюСформировать (Кнопка)

- //{ {КОНСТРУКТОР_ВЫХОДНЫХ_ФОРМ_ПРОЦЕДУРА_ВЫЗОВА (ОтчетПоНачислению)
- // Данный фрагмент построен конструктором.
- // При повторном использовании конструктора внесенные вручную
- // изменения будут утеряны!!!

```
ТабДок = ЭлементыФормы.ПолеТабличногоДокумента;
ОтчетПоНачислению (ТабДок, ДатаНачала, ДатаОкончания);
ТабДок.ОтображатьЗаголовки = Ложь;
//}}КОНСТРУКТОР ВЫХОДНЫХ ФОРМ ПРОЦЕДУРА ВЫЗОВА
КонецПроцедуры
Процедура ОтчетПоНачислению (ТабДок, ДатаНачала, ДатаОкончания) Экспорт
Макет = ОтчетОбъект.ПолучитьМакет ("ОтчетПоНачислению");
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
     НачислениеОбороты.Преподаватель,
     ПРЕДСТАВЛЕНИЕ (НачислениеОбороты. Преподаватель),
     НачислениеОбороты.СуммаОборот КАК СуммаОборот
IN3
     РегистрНакопления.Начисление.Обороты («ДатаНачала,
       «ДатаОкончания, , ) КАК НачислениеОбороты
| УПОРЯДОЧИТЬ ПО
     СуммаОборот УБЫВ";
Запрос.УстановитьПараметр("ДатаНачала", ДатаНачала);
Запрос.УстановитьПараметр ("ДатаОкончания", ДатаОкончания);
Результат = Запрос.Выполнить ();
OбластьЗаголовок = Макет.ПолучитьОбласть ("Заголовок");
ОбластьПодвал = Макет.ПолучитьОбласть ("Подвал");
ОбластьШапкаТаблицы = Макет.ПолучитьОбласть ("ШапкаТаблицы");
ОбластьПодвалТаблицы = Макет.ПолучитьОбласть ("ПодвалТаблицы");
ОбластьДетальныхЗаписей = Макет.ПолучитьОбласть ("Детали");
ТабДок.Очистить ();
ТабДок.Вывести (ОбластьЗаголовок);
ТабДок.Вывести (ОбластьШапкаТаблицы);
ВыборкаДетали = Результат.Выбрать ();
Пока ВыборкаДетали.Следующий() Цикл
    ОбластьДетальныхЗаписей.Параметры.Заполнить (ВыборкаДетали);
    ТабДок. Вывести (Область Детальных Записей);
КонецЦикла;
ТабДок.Вывести (ОбластьПодвалТаблицы);
ТабДок.Вывести (ОбластьПодвал);
//}}КОНСТРУКТОР ВЫХОДНЫХ ФОРМ
КонецПроцедуры
```

Перейдем в режим 1С:Предприятие, и на основе уже введенной информации заполним отчет. На рис. 8.23 приведена заполненная форма отчета по имеющимся в информационной базе данным. Перед формированием отчета (кнопка Сформировать) предварительно требуется на форме установить интересующий нас интервал дат, за который собираемся получить отчет по начисленным преподавателям суммам.

🛃 Конфигуратор (учеб	и́ная версия) - Конфигур	ация			
🛛 🙅 айл Правка Конфигурация Отладка Администрирование Сервис Окна Справка					
I 🗋 🝙 🔏 📾 📾 🖆 😋 🙀 🛤 📃 🔽 🐨 🐨 🛣 👘 🔭 🗸					
💈 ОтчетПоНачислению * 💦 🗌 🗙					
Таблицы и по Группи	ровка Условия Дополни	пель Объединения	Порядок	Итоги Построитель Отчет Выходная	фо
Форма Форма настр	юйки построителя отчета				
Название выходной формы: ОтчетПоНачислению					
С Модуль О Модуль Формы					
Заголовок выходной формы: Отчет по начислению • Новая форма					
Заголовок кнопки:	Сформировать			ОтчетПоНачислению	
Параметры: 1					
Имя параметра	Редактировать в форме	Подпись	Тип параметр	ра Элемент управления	
ДатаОкончания	✓	Дата окончания	Дата	аы Поле ввода	
ДатаНачала	✓	Дата начала	Дата	аы Поле ввода	
Запрос (Назад Далее) ОК Отмена Справка					
Для получения подсказки нажмите F1 САР NUM ги •					

Рис. 8.22. Закладка Выходная форма окна конструктора отчета выходной формы

陽 1С:Предприятие (учебная версия) - Конфигурация	
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
I 🗋 🛁 🖌 🖻 📾 📾 🗠 🕁 🖒 🦛	🙄 🗄 🎹 🧮 🦧 м м+ м- 🎇 🖕
👜 Отчет по начислению	_ 🗆 ×
Сформировать	
Дата окончания 15.09.2008 0:00:00	
Дата начала 10.09.2008 0:00:00	
Отчет по начислению	^
Преподаватель СуммаОборот	
Бородин В.А.	5 200
Синицын А.В.	2 400
Акатьев Д.Ю.	2 280
Итчет по начислению	
Для получения подсказки нажмите F1	CAP NUM //

Рис. 8.23. Отчет по начислению в режиме работы 1С:Предприятие

Реализация множественного подбора

Рассмотрим небольшое сервисное улучшение документа Учет. Дело в том, что при создании очередной строки табличной части первое поле, которое пользователь

должен заполнить, — поле преподаватель. При этом происходит автоматическое открытие формы списка справочника преподаватели. При вводе следующей строки действия повторяются — пользователь выбирает из справочника следующего преподавателя и т. д. Мы хотим реализовать более удобный вариант работы. А именно сделать так, чтобы пользователь мог в форме справочника отметить сразу нескольких преподавателей, которые затем всей группой добавились бы в табличную часть документа.

Во-первых, добавим в форму документа кнопку **Подбор преподавателей** (рис. 8.24), при нажатии которой будет открываться форма справочника с фамилиями преподавателей.

<u> К</u> онфигуратор (учебная версия) - Конфигурация	
🛛 🙅 айл Правка Форма Конфигурация Отладка Администрирование Сервис Окна Справка	
I 🗋 🔟 📓 🔏 📾 🖆 🖄 📩 🖉 🕅 🛤 📃 🔽 🕷 🖓 🛅	🥱 🐅 🙄
🚟 Документ Учет: ФормаДокумента	
Учет	
Действия - 👍 🔃 🍡 🏊 🥻 Перейти - 🥐 💭 Подбор преподавателей	
Номер: Последние тари	іфы
Дата: : : 🔲 Срез тарифов:	
Дата расчета:	
월 날 옷 철 묘 ↑ ↓ 밝 태 💭	
N Преподаватель Группа Вид занятия Число часов Тариф	Сумма
abi <none seoga=""> abi <none seoga=""></none></none></none></none></none></none>	аы <Поле в
	>
💭 ОК Записат	ть Закрыть
Для получения подсказки нажмите F1 САР NUM ги マ 110,0 а11590) x 25 //

Рис. 8.24. Изменения в форме документа Учет

Во-вторых, разработаем процедуру для реализации множественного подбора (листинг 8.7). Для этого следует использовать свойство формы МножественныйВыбор. Присвоив этому свойству значение истина, мы обеспечим возврат не одного выбранного элемента, а группы элементов, которых пользователь выделит в форме справочника.

Листинг 8.7. Процедура обработки нажатия кнопки Подбор преподавателей

```
Процедура ПодборПреподавателейНажатие(Элемент)
Форма = Справочники.Преподаватели.
ПолучитьФормуСписка(, ЭлементыФормы.Список);
Форма.РежимВыбора = Истина;
```

```
Форма.МножественныйВыбор = Истина;
Форма.Открыть();
КонецПроцедуры
```

В-третьих, создадим процедуру, добавляющую отмеченных преподавателей в табличную часть. Для этого в окне редактирования формы документа следует открыть окно свойств табличной части Список (рис. 8.25). Здесь нужно щелкнуть мышью по изображению лупы напротив события **Обработка выбора** и воспользоваться открывшимся окном для написания программного кода. Перед нами откроется заготовка процедуры для СписокОбработкаВыбора, где и следует разместить текст, который представлен в листинге 8.8.

📩 Конфигуратор (учебная версия) - Конфигурация				
📱 🛨 айл Правка Форма Конфигурация Отладка Администрирование Сервис Окна Справка				
I 🗋 🛋 🖬 🔏 📾 📾 🖬 🖕 🖬 🖬	n ▼ n *	î 🖻 🔭 🏅		
🚟 Документ Учет: ФормаДокумента	Свойства: Табличное поле	₽ X		
Учет	21 🗊 🍽 🗙 ∨			
Действия 🗸 🛃 🐼 🔩 🚹 Перейти 🗸	ПриВыводеСтроки	▼ Q △		
	ПриПолученииДаннь	▼ Q		
Номер:	ВыборЗначения	▼ Q		
Дата:	ПриИзмененииФлаж	▼ Q		
Дата расчета:	ПриОкончанииРедак	▼ Q		
	ОбработкаЗаписиНо	▼ Q		
от преподаватель Гриппа Вид	ОбработкаВыбора СписокОбработка	аВыбора 🔻 🔍		
аы «Поле ввода» аы «Поле ввода» аы «	НачалоПеретаскиван	- Q		
	ПроверкаПеретаски	- Q		
	ОкончаниеПеретаски	- Q		
	Перетаскивание	- Q		
имя, по которому оудет выполняться ооращени з лементу управления в алгоритме модуля		ращение к Ія		
📃 🗏 Диалог 🗏 Модуль 📑 Реквизиты	Имя, Name			
П Служебные сообщения				
Для получения подсказки нажмите F1	CAP NUM ru▼ 128,132 ₽	574 x 127		

Рис. 8.25. Окно свойств табличного поля

Листинг 8.8. Процедура для события ОбработкаВыбора

```
Процедура СписокОбработкаВыбора(Элемент, ВыбранноеЗначение,
СтандартнаяОбработка)
Для Счетчик = 0 По ВыбранноеЗначение.ВГраница() Цикл
Строка = Список.Добавить();
Строка.Преподаватель = ВыбранноеЗначение[Счетчик];
КонецЦикла;
КонецПроцедуры
```
Примечание

Таким образом, пользователь, работая с документом Учет, щелчком по кнопке **Подбор преподавателей** запускает процедуру, представленную в листинге 8.7. Данная процедура позволяет организовать выбор нескольких преподавателей за счет того, что у открываемой формы устанавливается значение Истина для свойств РежимВыбора и МножественныйВыбор. Далее (после нажатия в форме справочника клавиши <Enter>) автоматически запускается процедура, представленная в листинге 8.8, которая производит добавление отобранных строк с фамилиями преподавателей в табличную часть документа.

Теперь все, что необходимо сделать для множественного выбора, готово, и, перейдя в режим работы 1С:Предприятие, можно проверить созданную разработку в деле (рис. 8.26).

🏪 1С:Предприятие (учебная версия) - Конфигурация					
🛛 <u>Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авк	a				
D 🗳 🖬 🔏 🖻 🖷 🍜 🖄 🕁 🛃 🛤	🐥 📰 🧮 🦧 M M+ M- 🎇 🗸				
🗟 Учет: Учет (Новый)	🗐 Справочник Преподаватели 🔔 🗆 🗙				
Действия 🗸 💽 💽 🏠 🔂 Перейти 🗸	Действия 🗸 🔄 🔄 🐳 👔 🏹 🛅 🖷 - 🐥				
Номер: 000000004	Код Наименование 🔻 🛆				
Дата: 13.09.2008 0:00:00	 — 0000000 Акатее Д.ю. — 0000000 Бородин В.А. 				
Дата расчета:	О0000000 Синицын А.В.				
월 변동 혐 다 비 문 물					
N Преподаватель Группа Вид заня					
🎦 Документы Учет 🛛 🐱 Учет: Учет (Нов	ый) 🎒 Справочник Преподаватели				
Для получения подсказки нажмите F1	CAP NUM				

Рис. 8.26. Иллюстрация множественного подбора в справочнике

Примечание

Для группового выделения следует держать нажатой клавишу < Ctrl>.

Использование системы компоновки данных для создания сводной таблицы

Наверняка большинству читателей сводные таблицы хорошо знакомы благодаря приложению Microsoft Excel, которое обеспечивает удобный механизм создания отчетов такого вида. В системе 1С:Предприятие 8.1 заложены не менее эффективные технические возможности по созданию сводных таблиц.

Вообще, сводные таблицы предоставляют возможность существенно повысить информативность и наглядность отчетов. Наша задача здесь будет заключаться в разработке отчета, который при работе в режиме 1С:Предприятие выглядит по-

добно рис. 8.27. Здесь в столбцах таблицы перечислены преподаватели, в строках — учебные группы, а на пересечении столбца и строки указано количество часов занятий, которые провел конкретный преподаватель в определенной группе. В столбце **Итог** и строке с аналогичным названием подводятся итоги по группам и преподавателям. Таблицы подобного вида очень информативны и используются очень широко для представления количественной информации.

🏪 1С:Пре	дприятие (уче	бная версия)	- Конфигураци:	я		
<u>∎ Ф</u> айл <u>П</u> р	равка Таблица	Операции <u>С</u> ер	вис <u>О</u> кна Сп <u>р</u> а	вка		
I 🗋 🚢 I	- X 🖻 🖻	i 🛋 🗟 i 🖕	c 🐥	1 🗎 🧟	M M+	м- 🎇 🖵
🕸 Отчет	Сводная табл	лица				_ 🗆 ×
Действия	- 🕨 Сформи	ровать 🔤 🌺 Ко	онструктор настр	оек На	астройки	ĭn ⇒ ▼
Число ча	Преподаватель					^
Группа	Акатьев Д.Ю.	Синицын А.В.	Бородин В.А.	Итог		
07-дв-2	19	82	35	136		
07-ФМ-1		10		10		
07-ДВ-1			90	90		
Итог	19	92	125	236		
						~
<						>
政 Отчет С	водная таблица					
Для получе	ния подсказки н	ажмите F1			CAP	NUM //

Рис. 8.27. Сводная таблица по отработанной нагрузке

После такого введения перейдем в режиме конфигуратора к созданию нового отчета (нового объекта конфигурации) и, как и ранее, воспользуемся конструктором выходной формы. Информация, на основе которой будет строиться отчет, содержится в документах учет. Это связано с тем, что регистра для информации об отработанных часах мы не создавали.

В результате для отчета необходимо использовать имеющиеся в информационной базе заполненные документы учет. Нам потребуется информация из табличной части документов подобного вида. Начнем с привычных действий и в режиме конфигуратора создадим новый объект конфигурации — отчет Сводная Таблица. С помощью кнопки Открыть схему компоновки данных перейдем к формированию схемы компоновки данных и создадим новый набор данных (Добавить набор данных — запрос).

Далее, как и в примерах *главы* 7, с помощью соответствующей кнопки перейдем к работе с конструктором запроса и выберем табличную часть Список документа Учет.

Из данной таблицы в список полей включим три поля (рис. 8.28):

УчетСписок.Преподаватель;

- ♦ УчетСписок.Группа;
- УчетСписок.ЧислоЧасов.

<u>апроса</u>	n Dor		Объединениа/По	П	opasor	Kompohorkadah	Yapaktapuotu	~
Гарлицы и поля Труппировка Эсловия	ДОГ	Таблицы	оовединения/пс	. 10	Пола	КОМПОНОВКА ДАН	Ларактеристи	КИ
			Trucor		поля	ЧиетСписок Пропол		
						ПистСписок.Пренод	аватель	
штеподаватели		E=	ссылка		-	эчетсписок. Группа		
<u>⊢</u> н			НомерСтроки		····· -	9четСписок. Числоч	lacob	
⊞… 🚞 Дисциплины		±=	Преподаватель					
🛱 🛁 Документы		÷ =	Группа					
🚊 🖓 Учет		Ė… −	ВидЗанятия					
Ссылка	>		ЧислоЧасов	>				
ПометкаУдаления	>>		Тариф	\rightarrow				
— Номер		_	Симма					
Лата			-					
Проведен								-
	_							
				<				
	<<			<<				
представление								
МоментВремени								~
Запрос			< Назад Да	лее >		ОК Отмен	а Справк	a

Рис. 8.28. Закладка Таблицы и поля окна конструктора запроса

Перейдем на закладку **Группировка** и выберем все доступные поля для группировки (рис. 8.29). После этого можно щелкнуть по кнопке **ОК** и получить текст запроса (листинг 8.9).

🎬 Конструктор запроса		□ × □
Таблицы и поля Группировка Ус Поля ⊞⊶ ≣ Все поля	словия Дополнительно Объединения Порядок К рупповое поле <u>ЧчетСписок.Группа</u> учетСписок.Преподаватель	омпоновка д Характеристи
	Суммируемое поле УчетСписок.ЧислоЧасов	Функция Сумма
Запрос	< Назад Далее > ОК	Отмена Справка

Рис. 8.29. Закладка для установки группировок в запросе

Листинг 8.9. Текст запроса для формирования сводной таблицы

ВЫБРАТЬ

УчетСписок.Преподаватель, УчетСписок.Группа, СУММА (УчетСписок.ЧислоЧасов) КАК ЧислоЧасов

KЗ

Документ.Учет.Список КАК УчетСписок

```
СГРУППИРОВАТЬ ПО
УчетСписок.Группа,
УчетСписок.Преподаватель
```

Таким образом, в результате выполненных действий мы вернулись в окно схемы компоновки данных. Здесь следует перейти на закладку **Ресурсы** и выбрать доступный ресурс (рис. 8.30).

🛃 Конфигуратор (учебная в	ерсия) - Конфигурация			
🗄 <u>Ф</u> айл <u>П</u> равка Конфигурация	Отладка Администрирован	ние <u>С</u> ервис <u>О</u> кна Сп <u>р</u>	авка	
I 🗋 🛋 🖬 🕺 🏝 🛍 🧉	i 🖻 🖕 🖻 🛤 📃		• 🕯 🕯 🕒	🥱 🐅 糞
💐 Отчет СводнаяТаблица:	ОсновнаяСхемаКомпоно	вкиДанных		_ 🗆 ×
Наборы дан Связи набор	Вычисляем Ресурсы	Параметры Макеты	Вложенные	Настройки
Доступные поля	Поле	Выражение	Рассчитывать по) Â
🗕 Группа	- ЧислоЧасов	Сумма(ЧислоЧасов)		
🗕 Преподаватель				
🗕 ЧислоЧасов	>>			
	<			
				\checkmark
Для получения подсказки нажми	πe F1		CAP NUM	d ru <i>▼ ///</i>

Рис. 8.30. Установки на закладке Ресурсы

Далее процесс создания отчета будет заметно отличаться от предыдущих рассмотренных примеров. Сейчас нам необходимо создать форму отчета. Для этого перейдем на закладку **Формы** (рис. 8.31), где и создадим основную форму отчета.



Рис. 8.31. Закладка Форма окна редактирования объекта конфигурации

Для этого щелкнем по кнопке с изображением лупы напротив поля **Основная форма отчета**, после чего запустится конструктор форм (рис. 8.32).

В меню Таблица (раздел главного меню программы) выберем пункт Вид, где исключим режим Только просмотр. После чего установим курсор в верхнюю левую ячейку поля табличного документа (рис. 8.32) и выберем в меню Таблица раздел Встроенные таблицы.



Рис. 8.32. Окно формы отчета в режиме конфигуратора

Теперь осталось выбрать пункт Вставить сводную таблицу. При этом в табличное поле будет добавлена сводная таблица и откроется окно полей сводной таблицы (рис. 8.33).

Далее необходимо написать фрагмент программного кода. Для начала откроем модуль формы отчета и добавим переменную (рис. 8.34), которой воспользуемся в последующей программной процедуре.

Теперь необходимо создать обработчик события формы отчета приоткрытии. Для этого в окне свойств *формы* следует щелкнуть по изображению лупы напротив события приоткрытии и сформировать текст программной процедуры (листинг 8.10).

Листинг 8.10. Обработка события ПриОткрытии для формы

```
Процедура ПриОткрытии()
```

```
ИнформацияПоНагрузке = Новый ИсточникДанныхСводнойТаблицыКомпоновкиДанных;
```

ИнформацияПоНагрузке.УстановитьСхему (СхемаКомпоновкиДанных);

```
ЭлементыФормы.Результат.ВстроенныеТаблицы.СводнаяТаблица1.
```

```
ИсточникДанных = ИнформацияПоНагрузке;
```

ИнформацияПоНагрузке.УстановитьНастройки (КомпоновщикНастроек.Настройки); КонецПроцедуры

🛃 Конфигуратор (у	учебная версия) - Конфигурация	_ 🗆 🗙
<u>Ф</u> айл <u>П</u> равка Фо	рма Таблица Конфигурация Отладка Администр	рирование
<u>Сервис О</u> кна Сп <u>р</u> а	авка	
🗋 🚢 🖬 🐰 🖻	Ъ 💼 🥌 🔯 🖕 🗢 🖒 🚵 м	▼
📕 🚟 Отчет Сводна:	яТаблица: ФормаОтчета1	_ 🗆 ×
Отчет Сводная	таблица	
Действия - 🕨 🕻	Сформировать 🛛 🔨 Конструктор настроек 🛛 Нас	стройки 🕨
	Область колоноі Поля сво	дной таб 🗙
Область строк	Область данны»	
🗏 Диалог 🗮 М	Модуль 📋 Реквизиты	
Для получения подска	азки наж САР NUM ги т 108, 33 т	385 x 185

Рис. 8.33. Форма отчета в виде сводной таблицы

🍌 Конфигуратор (учебная версия) - Конфигурация 📃 🔲 🗙
📱 🛨 айл Правка Текст Конфигурация Отладка Администрирование
Сервис Окна Справка
II 🛍 现 🕢 🎉 🛤 💽 🗸 🗸
📕 🚟 Отчет СводнаяТаблица: ФормаОтчета1 🛛 💶 🗙
Перем ИнформацияПоНагрузке;
🗐 Диалог 🗏 Модуль 📑 Реквизиты
Для получен [CAP NUM OVB и ч Стр. 1 Кол. 28

Рис. 8.34. Определение переменной в модуле формы отчета

Здесь мы присвоили сводной таблице в качестве источника данных схему компоновки данных и установили для нее настройки, которые определены в объекте компоновщикНастроек. На этом работа в режиме конфигуратора завершена, и можно перейти в режим 1С:Предприятие, где мы и откроем разработанный отчет (рис. 8.35). Здесь перед нами открываются форма отчета и окно выбора полей сводной таблицы.

Далее от нас требуется с помощью мыши перенести определенные поля в колонки таблицы, строки таблицы и в область данных. В результате мы получим необходимый отчет (рис. 8.36).

🏪 1С:Пре	цприятие (учебная версия) - Конфигурация
<u>∎ Ф</u> айл <u>П</u> р	авка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка
I 🗋 🛋 I	1 X 🖻 🛍 🍜 🖸 5 C 8 👘 🐥 📲 📰 🖉 M M+ M- 🎘 🗸
🕸 Отчет	Сводная таблица 📃 🗖 🗙
Действия	- 🕨 Сформировать 🛛 🔨 Конструктор настроек 🛛 Настройки 🕅 🖳 🤶
	Область колонок
Область строк	Область данных Поля сводной таблицы ×
 Отчет С Для получе 	зодная таблица ния подсказки нажмите F1

Рис. 8.35. Окно отчета со сводной таблицей

🏪 1С:Предприятие (учебная верс	:ия) - Конфигураци	я			_ 🗆 🗵
🖩 🕂 айл Правка Таблица Операции	<u>С</u> ервис <u>О</u> кна Сп <u>р</u> а	авка			
D 🗳 🖬 🐰 🏪 🛍 🛋 🙆	5 C M		• # #	🕒 🍟 🖩 🗐 🧣	M M+ M- 🎇 🗸
🖤 Отчет Сводная таблица					_ 🗆 ×
Действия - 🕨 Сформировать	📩 Конструктор настр	оек Настр	ойки 🔄 📑	?	
Число ча Преподаватель					
Группа Акатьев Д.Ю. Бородин В	.А. Синицын А.В.	Орлов Д.О.	Орлов Д.О. Петр	овская В.Л. Князева	E.П. Итог
07-дв-2 19	47 82				10 158
07-дв-1	90				90
07-ФМ-1	44	12	14	17	87
Итог 19	137 126	12	14	17	10 335
📇 Документы Учет 🛛 🗐 О-	тчет Сводная таблица	a			
Для получения подсказки нажмите F1					CAP NUM

Рис. 8.36. Отчет в виде сводной таблицы

Формирование сводной таблицы с использованием конструктора

Кроме системы компоновки данных, для создания сводных таблиц можно также воспользоваться и более простыми ресурсами, к которым относится конструктор выходной формы. В этом разделе мы продемонстрируем создание такой же сводной таблицы, но другим методом.

Итак, приступим в режиме конфигуратора к созданию нового отчета. На закладке Основные окна редактирования объекта конфигурации укажем имя Сводная Таблица2. Теперь перейдем на закладку Макеты, где воспользуемся кнопкой Конструкторы, и выберем Конструктор выходной формы.

Отчет также будет использовать имеющиеся в информационной базе заполненные документы учет. Нам потребуется информация из табличной части этих документов.

На рис. 8.37 показана закладка Таблицы и поля окна конструктора выходной формы. В разделе База данных следует отметить таблицу Список документа учет и



Рис. 8.37. Закладка Таблицы и поля окна конструктора выходной формы

📩 Конфигуратор (учебная версия) - К	онфигурация				
<u>п Ф</u> аил <u>П</u> равка Конфигурация Отладка П	Администрирование <u>С</u> ервис <u>О</u> кн	а справка			
<u></u> ∑ СводнаяТаблица2 *					_ 🗆 ×
Таблицы и поля Группировка Условия	Дополнительно Объединения/П	се Порядок Итог	и Построитель	Отчет В	Выходная форма
Поля	Группировочное поле	Тип итогов	Псевдоним		
f(x) ПреподавательПредставлен	Преподаватель	Элементы	Преподаватель		
f(x) ГруппаПредставление >>	Группа	Элементы	Группа		+
🗄 🖷 🚍 Все поля					
<					
11					
	✓ Общие итоги				
	Итоговое поле	Выражение			
	ЧислоЧасов	СУММА(ЧислоЧасов)			
<					
< > <<					~
Запрос	< Ha	азад Далее >	OK	Отмена	Справка
Для получения подсказки нажмите F1				CAF	NUM ru•

Рис. 8.38. Закладка Итоги окна конструктора выходной формы

перенести ее в раздел **Таблицы**. Так же, как и в предыдущем примере, в раздел **Поля** перенесем три поля. Конструктор самостоятельно добавит еще два поля: представления полей Преподаватель и Группа.

Затем следует перейти на закладку **Итоги** и определить группировочные поля, а также итоговое поле в соответствии с рис. 8.38. Осталось установить флажок на закладке **Отчет** (рис. 8.39), показывающий, что нам понадобится сводная таблица, и после этого щелчком по кнопке **ОК** завершить работу конструктора.



Рис. 8.39. Закладка Отчет окна конструктора выходной формы

📩 Конфигуратор (учебная версия) - Конфигурация	
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка Администрирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
🚟 Отчет СводнаяТаблица2: СводнаяТаблица2	_ 🗆 🗙
Сводная таблица2	
Сформировать 💭	
Диалог Кодуль Реквизиты	
Для получения подсказки нажмите F1 САР NUM ru + +0,0 +600 × 4	100 //

Рис. 8.40. Незаполненная форма сводного отчета

Теперь можно перейти в режим работы 1С:Предприятие и начать работу со сводной таблицей. В меню **Отчеты** следует выбрать название разработанного отчета, и в результате мы увидим окно, показанное на рис. 8.40.

Здесь от пользователя требуется сначала щелкнуть по кнопке Сформировать, а затем с помощью контекстного меню перенести поля сводной таблицы на форму (рис. 8.41).

🏪 1С:Предприятие (у	чебная в	ерсия) -	Конфиг	урация		×	
🛛 <u>Ф</u> айл <u>П</u> равка Таблица Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка							
🗋 📫 🖬 🐰 🖻	6 6	0 5	c m	» ▼	🎹 🦳 🧟 M M+ M- 🎇	-	
🖤 Сводная таблица	2				_ 🗆 >	<	
Сформировать					Поля сводной таблицы 🛛 🗙	U	
Числочасов	Γργππα	1			– Преподаватель т Гриппа		
Преподаватель	07-дв-2	07-ФМ-1	07-ДВ-1	Итог	ЧислоЧасов		
Акатьев Д.Ю.] 19			19		ш	
Синицын А.В.	82	44		126		ш	
Бородин В.А.	47		90	137		ш	
Князева Е.П.	10			10		ш	
Орлов Д.О.		12		12		ш	
Орлов Д.О.		14		14			
Петровская В.Л.		17		17			
Итог	158	87	90	335	Чбрать 🔻 ОК		
🎒 Сводная таблица2							
Для получения подсказк	и нажмите	e F1			CAP NUM	///	

Рис. 8.41. Настройка полей сводного отчета

Передача данных между приложениями

В практической работе интеграция различных приложений используется очень часто. Так, разные сотрудники организаций, как правило, работают с различным набором программных продуктов, однако им нужны одни и те же данные. Набор используемых программ зависит и от занимаемых ими должностей, и связан с различными профессиональными обязанностями, а иногда и с хорошим знанием определенных программ. Поэтому даже при хорошей организации информатизации предприятия все равно часто требуется передавать данные, созданные в одних приложениях, в другие программы.

Наверное, все согласятся, что для большинства фирм характерно использование в офисной работе приложений Microsoft Word и Microsoft Excel. С другой стороны, статистика говорит о том, что и программный продукт 1С:Предприятие 8.1 также охватывает в той или иной степени участки деятельности практически каждой организации в России. Для совместной работы сотрудникам приходится организовывать передачу текущей или отчетной документации из одних программ в другие. Организация подобного обмена может возлагаться либо на системных программистов, либо на обыкновенных пользователей, но в обоих случаях этот этап требует заметных затрат и усилий. В данном разделе на примерах рассматривается организация электронного обмена данными, затрагивающая три приложения — 1С:Предприятие 8.1, Microsoft Word и Microsoft Excel. Рассмотренные далее приемы можно использовать и при решении других офисных задач, требующих подобной отчетной информации.

Приложения, входящие в пакет Microsoft Office, и система 1С:Предприятие являются открытыми для взаимного обмена информацией. В программный продукт 1С:Предприятие 8.1 заложены ресурсы [8, 9], которые позволяют:

- осуществлять обработку и обмен данными различных форматов;
- осуществлять доступ ко всем объектам системы 1С:Предприятие, реализующим ее функциональные возможности;
- поддерживать различные протоколы обмена;
- поддерживать стандарты взаимодействия с другими системами.

Разработчиками в систему 1С:Предприятие 8.1 включен набор технологий, ориентированных на решение разнообразных задач интеграции. В литературе [6, 8, 9] дается обзор всех ресурсов, здесь же мы уделим внимание только практическому использованию некоторых (наиболее интересных на наш взгляд) способов обмена данными.

Взгляд на технологии обмена данными

Начнем с рассмотрения COM-технологии, что расшифровывается следующим образом — Component Object Model (компонентная объектная модель). Эта информационная модель представляет собой технологию создания программ, использующих набор определенным образом взаимодействующих компонентов, которые в свою очередь состоят из объектов. Данные компоненты представляют собой выполняемый двоичный код, и для использования того или иного компонента достаточно просто зарегистрировать его в операционной системе.

Примечание

Фактически при использовании СОМ-технологии не требуется осуществлять компоновку компонентов с основным модулем.

Другая не менее известная технология — Automation Client/Server. Еще одно ее название — OLE Automation, что является сокращением от Object Linking and Embedding (связывание и внедрение объектов). В этом случае пользователям предоставляется возможность работы с составными объектами, созданными в других приложениях (документы, таблицы, рисунки и др.). Основные понятия, с которыми оперирует данная технология, — это OLE-объект, сервер приложения и контейнер приложения.

Объектом OLE называют объект, созданный в другом приложении, но при этом сохранивший связь с текущим приложением. Для этого он должен быть вставлен в интересующий нас документ соответствующим способом.

Контейнером OLE является приложение, в котором создается составной документ, содержащий OLE-объект. В этом случае при необходимости работы с OLEобъектом открывается исходное приложение (то, в котором рассматриваемый объект был создан). Это исходное приложение называется *сервером OLE*. Офисные приложения Microsoft Word, Microsoft Excel и система 1C:Предприятие 8.1 могут выступать и как OLE-сервер, и как OLE-контейнер.

В целом работа системы 1С:Предприятие через СОМ-соединение похожа на работу в режиме Automation Client/Server. Существенное отличие заключается в том, что при СОМ-соединении запускается вариант информационной системы 1С:Предприятие, не содержащий интерфейсной части. Работа в этом случае производится только программными средствами. Интерактивные возможности работы пользователя исключаются. Основное назначение Automation-сервера 1С:Предприятия — управление работой системы 1С:Предприятие 8.1 из других приложений и выполнение действий, аналогичных интерактивной работе.

Использование СОМ-технологии

Рассмотрим практический пример, который будет иллюстрировать передачу информации из Microsoft Excel в справочник системы 1С:Предприятие 8.1. Вы увидите, что эти действия не требуют написания длинного кода, что является существенным плюсом для разработчика.

Для обмена мы воспользуемся СОМ-соединением, которое реализуем в процедуре для системы 1С:Предприятие. В данной процедуре из рабочей книги Microsoft Excel будет передаваться список преподавателей в соответствующий справочник системы 1С:Предприятие. Для определенности будем считать, что информация со списком преподавателей располагается в отдельном файле — книге Microsoft Excel. Лист этой книги с информацией показан на рис. 8.42.



Рис. 8.42. Информация о преподавателях в книге Microsoft Excel

Основные действия будут происходить в конфигураторе. Воспользуемся ранее созданным объектом конфигурации — справочником преподаватели. На форме списка справочника расположим кнопку (рис. 8.43) для передачи информации из книги Microsoft Excel в базу данных на платформе 1С:Предприятие 8.1. Для обработки нажатия кнопки следует написать процедуру, представленную в листинге 8.11.

Здесь в первой строке процедуры мы устанавливаем связь с файлом приложения Microsoft Excel, который должен предварительно быть создан и размещен на диске C: под именем Преподаватели.xlsx.

🛃 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🗙				
<u>Ф</u> айл Правка Форма Конфигурация Отладка Администрирование <u>С</u> ервис				
Окна Справка				
📓 Справочник Преподаватели: ФормаСписка 💶 🗌	×			
Справочник Преподаватели				
Действия - 😫 😫 🔩 🐐 🏹 🎦 🕅 - 🌾 🔯 🥐 💭				
Код Наименование				
аы «Поле ввода» аы «Поле ввода»				
Передать в IC				
	_			
Диалог Модуль 🗧 Реквизиты				
Для получения подсказки САР NUM ги т 📩 0, 0 📫 387 x 172	///			

Рис. 8.43. Добавление кнопки в форму списка справочника Преподаватели

Листинг 8.11. Процедура переноса информации по преподавателям

```
Процедура ПеренестиИнформациюОПреподавателяхНажатие (Элемент)

ConExcel = ПолучитьСОМОбъект ("C:\Преподаватели.xlsx");

Cпp = Справочники.Преподаватели;

i = 1;

Пока СтрДлина (ConExcel.Sheets(1).Cells(i, 1).Value) > 0 Цикл

НовыйЭлемент = Спр.СоздатьЭлемент();

НовыйЭлемент.Наименование = ConExcel.Sheets(1).Cells(i, 1).Value;

НовыйЭлемент.Записать();

i = i + 1;

КонецЦикла;

КонецПроцедуры
```

В цикле последовательно просматриваются все ячейки первого столбца на первом листе книги Microsoft Excel. Цикл выполняется до тех пор, пока очередная ячейка листа Excel не окажется пустой.

Примечание

Функция СтрДлина () возвращает количество символов в строке. В качестве единственного параметра функции необходимо указать строку.

Таким образом, мы обеспечили передачу информации с рабочего листа Microsoft Excel в справочник системы 1С:Предприятие. На рис. 8.44 показан результат — в столбце наименование мы видим фамилии новых преподавателей.

🏪 1С:Предприятие (учебная версия) - Конфигурация 📃 🗖 🗙				
🛯 <u>Ф</u> айл <u>П</u> равка Операции ј	<u>Сервис О</u> кна Сп <u>р</u> авка			
I 🗋 🛋 🖬 🐰 堶 👘	a 🗟 🙄 i 🎹 🛅 🧝 M M+ M- 🎘 🤳			
🗐 Справочник Преподав	атели _ 🗆 🗙			
Действия 🗸 🔄 🔄 ≼ 🍝	i 🕅 🔁 🖷 - 🌾 🖸 ?			
Код	Наименование 🔹 🔿			
- 00000001	Акатьев Д.Ю.			
- 00000002	Бородин В.А.			
- 00000006	Князева Е.П.			
- 00000004	Орлов Д.О.			
- 00000005	Петровская В.Л.			
- 00000003	Синицын А.В.			
Передать в 1С				
л. Правочник Преподаватели				
Для получения подсказки нажмите F1 САР NUM				

Рис. 8.44. Результат переноса информации в системе 1С:Предприятие

В процедуре, приведенной в листинге 8.11, для реализации данного эффекта используются объекты VBA (Visual Basic for Applications — язык программирования, применяемый в Microsoft Office) — коллекции Sheets и Cells. Коллекция Sheets представляет имеющиеся в книге рабочие листы. Указав в качестве параметра единицу, мы получаем возможность работать с первым листом. Объект Cells позволяет работать с содержимым ячеек рабочих листов. При обращении к нему необходимо указать два параметра. Первый из них определяет номер строки ячейки, а второй — номер столбца.

Передача информации в Microsoft Excel с использованием технологии OLE Automation

СОМ-технология не обеспечивает использования визуальных ресурсов приложения-сервера, и в противоположность ей OLE Automation позволяет это делать. Рассмотрим ситуацию, когда требуется организовать передачу информации из системы 1С:Предприятие в рабочую книгу Microsoft Excel.

Нам понадобится создать еще один объект конфигурации — справочник дисциплины. Кроме обязательных реквизитов добавим еще два:

- лекции (тип данных число) для отражения количества лекционных занятий по данной дисциплине;
- практика (тип данных число) для отражения количества практических занятий.
 Табличных частей у элементов справочника не будет.

На закладке **Формы** создадим форму списка справочника. На форме разместим кнопку **Передать данные в Excel** (рис. 8.45), при нажатии которой в файл Дисциплины.xlsx будет передаваться полная информация по дисциплинам (с включением данных о количестве занятий).

📩 Конфигуратор (учебная версия) - Конфигурация 📃 🗖 🔀					
<u>Ф</u> айл <u>П</u> равка Форма) Конфигурация Отладка /	Администрирс	вание <u>С</u> ервис	<u>О</u> кна	
Справка					
🧮 Справочник Дисци	иплины: ФормаСписка		_ [⊐ ×	
Справочник Дисцип	лины			\neg	
Действия - 😫 😫 -	s 😼 🏹 To 🗃 - Tr	0 ?)		
Код	Наименование	Лекции	Практика	7	
аы «Поле ввода»	аы <Поле ввода>	аы «Поле	аы «Поле в		
<			>		
П Передать данные в Excel D+					
	·····				
🗐 Диалог 📃 Модул	њ 🗧 Реквизиты				
Для получения подсказки	ина CAP NUM ru▼ *	123, 139	‡ 177 x 29		

Рис. 8.45. Форма списка справочника Дисциплины

Отличием выполняемого обмена является то, что в процессе передачи данных окно рабочей книги Microsoft Excel показывается на экране. Текст программной процедуры, обеспечивающей рассматриваемый эффект, приведен в листинге 8.12, а сам результат ее выполнения показан на рис. 8.46.



Рис. 8.46. Заполнение листа Microsoft Excel с помощью процедуры в системе 1С:Предприятие

Листинг 8.12. Передача информации с использованием технологии OLE Automation

Процедура ПередатьВЕхсеlНажатие(Элемент) ApplicationExcel = Новый СОМОбъект("Excel.Application"); ApplicationExcel.Visible = Истина;

```
ApplicationExcel.Workbooks.Open("C:\Дисциплины.xlsx");

Спр = Справочники.Дисциплины;

i = 2;

Выборка = Спр.Выбрать();

Пока Выборка.Следующий() > 0 Цикл

ApplicationExcel.Sheets(1).Cells(i, 1).Value = Выборка.Наименование;

ApplicationExcel.Sheets(1).Cells(i, 2).Value = Выборка.Лекции;

ApplicationExcel.Sheets(1).Cells(i, 3).Value = Выборка.Практика;

i = i + 1;

КонецЦикла;

КонецПроцедуры
```

Примечание

Процедура, приведенная в листинге 8.12, предполагает, что файл С:\Дисциплины.xlsx соответствующим образом подготовлен: он должен содержать заголовки и не должен содержать большее количество дисциплин, чем передается из программы 1С:Предприятие. В реальных разработках не стоит делать таких допущений, а лучше явно очищать лист и заполнять шапку в программной процедуре. Тут это не сделано, чтобы не усложнять пример.

Выполнение макросов при передаче информации в Microsoft Excel

Немного усложним ситуацию, рассмотренную в предыдущем разделе. Допустим, что в файле Дисциплины2.xlsm содержится информация о дисциплинах с указанием количества учебных часов. Для определенности эта информация будет располагаться на втором листе, который называется **План** (рис. 8.47). Можно сказать, что данный лист хранит базовую информацию, а в системе 1С:Предприятие ведется

	💽 🗐 🛪 📷 🗊 📮 Фисциплины2.xlsm - Microsoft Ex 📁 🛪					
C	Лавна Вставк П	Разме Форм	иу Даннь Рец	ен: Вид Раз	spał 🕜 –	- x
Te	Ада Ат мы От Темы	ры Вписать	[] Параметры листа ▼	порядочить •		
	E12	+ ()	f_{x}			≽
	А	В	С	D	E	
1	Дисциплина	Лекции	Практика			
2	Информатика	30	10	D		
3	3 История 25 17					
4	Экономика	30	10	D		
Гот	ово		田口口 10	0%	U	(+)

Рис. 8.47. Лист План рабочей книги Дисциплины2.xlsm

оперативная работа. Очевидно, что время от времени требуется устанавливать соответствие.

В данной ситуации нам необходимо, чтобы после переноса информации из справочника дисциплины на первый лист книги Дисциплины2.xlsm имеющиеся данные сравнивались бы с полученными из системы 1С:Предприятие. Необходимо отмечать наличие противоречия в указании часов занятий для определенной дисциплины. Это противоречие должно отражаться в виде автоматически формируемого сообщения в четвертом столбце (рис. 8.48).

Задача теперь усложняется, т. к. требуется и выполнение программы в среде 1С:Предприятие 8.1, и выполнение другой программы на языке VBA для Microsoft Excel. Причем вторая программа должна быть запущена не вручную, а автоматически из первой (процедура, выполняемая в системе 1С:Предприятие 8.1, должна инициировать выполнение процедуры в Microsoft Excel). Программирование для Microsoft Excel мы затрагиваем только в объеме, необходимом для понимания приведенных здесь листингов. Желающие ознакомиться подробнее с этим предметом могут обратиться к книгам [3, 5, 10].



Рис. 8.48. Диагностика при расхождении информации в двух системах учета

Для достижения подобного эффекта нам потребуется разработать в книге Дисциплины2.xlsm макрос, текст которого представлен в листинге 8.13. Технически для размещения макроса в книге Microsoft Excel необходимо на вкладке **Разработ**чик выбрать пиктограмму **Макросы**, в открывшемся диалоговом окне (рис. 8.49) ввести имя нового макроса и, нажав кнопку **Создать**, перейти в окно для написания программного кода. Сам текст макроса достаточно несложен и снабжен комментариями, которые начинаются с апострофа ('). В нем происходит последовательное сравнение данных с первого листа (на который информация поступает из базы данных на платформе 1С:Предприятие 8.1) с данными, расположенными на втором листе. В случае нахождения различия в числе часов на первый лист выводится соответствующая диагностика.

Листинг 8.13. Текст макроса, расположенного в книге Дисциплины2.xlsm

```
Sub Обработка ()
N1 = 0

    Ищем первую пустую ячейку в первой колонке на первом листе,

' начиная со второй строки.
While Worksheets(1).Cells(N1 + 2, 1).Value <> ""
                                                      ' Шикл
  N1 = N1 + 1
Wend ' Конец цикла
N2 = 0
' Ищем первую пустую ячейку в первой колонке на втором листе,
' начиная со второй строки.
While Worksheets(2).Cells(N2 + 2, 1).Value <> ""
                                                      ' Цикл
  N2 = N2 + 1
Wend ' Конец цикла
For i = 1 To N1 ' Цикл с присвоением і значений от 1 до N1
  ' Запишем в переменную а название і-го курса на первом листе.
  a = Worksheets(1).Cells(i + 1, 1).Value
  For j = 1 To N2 ' Цикл с присвоением ј значений от 1 до N2
    ' Запишем в переменную b название j-го курса на втором листе.
    b = Worksheets(2).Cells(j + 1, 1).Value
    If a = b Then
      ' Если названия курсов совпадают, проверяем совпадение часов.
      ' Подчеркивание () служит для переноса строки.
      If Worksheets(1).Cells(i + 1, 2).Value <>
              Worksheets(2).Cells(j + 1, 2).Value Or
              Worksheets(1).Cells(i + 1, 3).Value <>
              Worksheets(2).Cells(j + 1, 3).Value Then

    делаем отметки о различиях

        Worksheets(1).Cells(i + 1, 4).Value =
          "Противоречие с " + CStr(j + 1) + " -й строкой 2-го листа"
        Worksheets(2).Cells(j + 1, 4).Value =
           "Противоречие с " + CStr(i + 1) + " -й строкой 1-го листа"
      End If
      Exit For ' выходим из внутреннего цикла (по j)
    End If
  Next ' Конец цикла
Next
End Sub
```

Таким образом, мы рассмотрели процесс программирования для приложения Microsoft Excel, и теперь наступает этап программирования для системы 1С:Предприятие 8.1. В листинге 8.14 приведен текст процедуры обработки нажатия кнопки в форме списка справочника дисциплины. В тексте процедуры для выполнения макроса Excel используется метод Run(), параметром которого является имя ранее созданного макроса. Результат выполнения мы уже видели на рис. 8.48 — для удобства пользователя помечаются строки на листе Microsoft Excel, на которые ему следует обратить внимание.

Макрос	? ×
Им <u>я</u> макроса:	
Обработка 🗾	Выполнить
Обработка	Войти
	Изменить
	Соз <u>да</u> ть
	Удалить
×	Параметры,
Находится в: Дисциплины2.xlsm	
Описание	
	Отмена

Рис. 8.49. Окно для работы с макросами

Листинг 8.14. Текст процедуры передачи данных для 1С:Предприятие 8.1

```
Процедура ПередатьВЕхсеlНажатие (Элемент)

ApplicationExcel = Новый СОМОбъект ("Excel.Application");

ApplicationExcel.Visible = Истина;

ApplicationExcel.Workbooks.Open ("C:\Дисциплины2.xlsm");

Cnp = Справочники.Дисциплины;

i = 2;

Выборка = Спр.Выбрать();

Пока Выборка.Следующий() > 0 Цикл

ApplicationExcel.Sheets(1).Cells(i, 1).Value=Выборка.Наименование;

ApplicationExcel.Sheets(1).Cells(i, 2).Value=Выборка.Лекции;

ApplicationExcel.Sheets(1).Cells(i, 3).Value=Выборка.Практика;

i = i + 1;

КонецЦикла;

ApplicationExcel.Run("Обработка");

КонецПроцедуры
```

Интеграция трех офисных приложений

В данном разделе рассмотрим еще одну ситуацию, связанную с обменом информацией между Microsoft Word, Microsoft Excel и 1С:Предприятие. В силу примерно одинаковой популярности этих приложений она вполне возможна при практической работе. Начнем наши действия с создания новой обработки. На рис. 8.50 показана форма, которая нам понадобится. На форме располагаются два поля ввода для внесения интервала дат (тип данных, разумеется — дата), а также кнопка, с помощью которой информация будет передаваться в рабочую книгу Microsoft Excel. Передаваемая информация представляет собой суммы, начисленные преподавателям в указанном временном интервале.

🛃 Конфигуратор (учебная версия) - Конфигурация 📃 🗖	×
<u>Ф</u> айл <u>П</u> равка Форма Конфигурация Отладка	
Администрирование <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка	
🚟 Обработка ОбработкаДанных: Форма 📃 🗖	x
Форма	<u> </u>
Дата начала:	
Дата окончания: 🕢 🔳	
р Передать данные об оплате р-	
П. Слижебные сообщения	
□ ag pogue CAP NUM rux 1:-72 109 1:195 x 25	_

Рис. 8.50. Форма обработки для передачи данных

0	😱 🖉 – 📓 – 😨 🗋 – ИнформацияИз1C.xlsm - Microsoft Excel – 🗖 🗙					x	
Ľ	Лавная Вставка Ра	азметка Формулы	Данные Ре	ецензир Ви	ід Разработ 🧕) - 📼 :	x
Вст	Саlibri Ж К Ц тавить • • • • • • • Шриф	× 11 × ≡ = × A* A* ≡ = ▲ × □ T □ Bupa	≡ <mark>=</mark> = ≣ = ■ २ внивание	% Число т	А Стили Эчейки	Σ - ∯7 - 	
	E10 • (• fx					¥
	А	В	С	D	E	F	
1	Фамилия	Сумма		Начало	Окончание		
2	Акатьев Д.Ю.	2280)	01/09	22/09		
3	Бородин В.А.	5200)	_			
4	Синицын А.В.	2400)	Перед	ать в Word		
5							
6			1				-
H 4	И ◆ ► Ы ЛИСТ1 / ЛИСТ2 / ЛИСТ3 / 💭 / 🛛 🛛 🛛 🗼						
Гот	Готово 🛅 🔲 🛄 100% 😑 — 🖓 🕂						

Рис. 8.51. Добавление элемента управления на лист Microsoft Excel

Мы хотим записать на лист Microsoft Excel информацию, подобную показанной на рис. 8.51. А именно, после нажатия кнопки **Передать данные об оплате** должна открываться рабочая книга Microsoft Excel и данные о начисленных суммах долж-

ны переноситься в ячейках листа этой рабочей книги (рис. 8.51). Для полноты информации, кроме начисленных сумм на лист Microsoft Excel, должен быть записан еще и интервал дат. Это первая часть разработки, которая связана с программированием в системе 1С:Предприятие.

Другая ее часть будет связана с передачей информации в Microsoft Word (но уже из приложения Microsoft Excel). В связи с этим на созданный лист необходимо добавить кнопку **Передать в Word**. Нам потребуется запрограммировать событие — нажатие этой кнопки. В этом случае будет выполнена процедура на языке VBA, которая запустит (с отображением на экране) приложение Microsoft Word и создаст в нем новый документ (рис. 8.52) — отчет о начислениях.

Таким образом, мы поставили себе цель и теперь опишем технические шаги для ее реализации. Конвейер передачи данных включает в себя уже три приложения: 1С:Предприятие, Microsoft Word и Microsoft Excel.



Рис. 8.52. Отчет в приложении Microsoft Word

Первая часть разработки, связанная с программированием в системе 1С:Предприятие, требует создания нового объекта конфигурации — обработки. В этой обработке мы воспользуемся формой произвольного вида, на которой разместим следующие элементы управления:

- поле ввода Датаначала (тип данных Дата);
- поле ввода ДатаОкончания (тип данных Дата);
- кнопка ПередатьданныеОбОплате.

Теперь для события нажатие, относящегося к кнопке **Передать данные об** оплате, на закладке **Модуль** расположим процедуру (листинг 8.15). Ее программные конструкции нам уже знакомы, а для обмена информацией мы воспользовались технологией OLE Automation. В процедуре открывается файл C:\ИнформацияИз1C.xlsm, куда и записывается информация, полученная в результате выполнения запроса.

Листинг 8.15. Процедура обработки нажатия кнопки Передать данные об оплате

```
Процедура Передать Данные ОбОплате Нажатие (Элемент)
ApplicationExcel = HOBENT COMOGDERT("Excel.Application");
ApplicationExcel.Visible = Истина;
ApplicationExcel.Workbooks.Open("C:\ИнформацияИз1C.xlsm");
Запрос = Новый Запрос;
Запрос. Текст = "ВЫБРАТЬ
     НачислениеОбороты. Преподаватель,
     НачислениеОбороты.Преподаватель.Представление КАК Преп,
     НачислениеОбороты.СуммаОборот КАК Оборот
IN3
| РегистрНакопления.Начисление.Обороты(«ДатаНачала, «ДатаОкончания, Период,)
| КАК НачислениеОбороты";
Запрос.УстановитьПараметр("ДатаНачала", ДатаНачала);
Запрос.УстановитьПараметр ("ДатаОкончания", ДатаОкончания);
Результат = Запрос.Выполнить();
Выборка = Результат.Выбрать ();
i = 2;
Пока Выборка.Следующий() Цикл
    ApplicationExcel.Sheets(1).Cells(i, 1).Value = Выборка.Преп;
    ApplicationExcel.Sheets(1).Cells(i, 2).Value = Bufopra.Ofopor;
    i = i + 1;
КонецЦикла;
Дата1 = Формат (ЭлементыФормы.ДатаНачала.Значение, "ДФ=""дд/ММ""");
ApplicationExcel.Sheets(1).Cells(2, 4).Value = Jara1;
Дата2 = Формат (ЭлементыФормы.ДатаОкончания.Значение,"ДФ=""дд/ММ""");
ApplicationExcel.Sheets(1).Cells(2, 5).Value = Дата2;
КонецПроцедуры
```

В данной процедуре используется функция форматирования для отображения даты в необходимом виде. Сама функция имеет следующий формат:

```
Формат (Значение, Форматная строка)
```

Здесь значение и есть те данные, которые требуется определенным способом отформатировать. Форматная строка представляет собой строковое выражение, включающее в себя параметры форматирования. Каждый параметр задается своим именем, за которым следует символ "=" и значение. Имя дф определяет формат даты. В тексте листинга 8.15 используется вариант представления даты в виде дд/мм. Теперь перейдем к программированию в приложении Microsoft Excel. В листинre 8.16 представлен текст процедуры, которая должна выполняться при щелчке по кнопке **Передать в Word**.

Для запуска приложения Microsoft Word и его отображения на экране в процедуре Microsoft Excel используются следующие программные конструкции:

```
Dim appWD As Object
Set appWD = CreateObject("Word.Application")
appWD.Visible = True
```

Здесь функция createObject() создает объект Microsoft Word и возвращает ссылку на это приложение. С помощью этой ссылки можно программно работать с объектами редактора Microsoft Word — документами и их составляющими (абзацами, строками, символами и т. д.). В третьей строке приведенного фрагмента используется свойство visible (видимость), и с помощью присвоения ему значения True (истина) окно текстового редактора появляется на экране. Если эти три строки оформить в виде текста процедуры, выполняемой по щелчку по кнопке, то, щелкнув по кнопке, мы из приложения Microsoft Excel попадем в редактор Microsoft Word.

Располагая ссылкой на объект-приложение Microsoft Word, можно работать с подчиненными объектами и коллекциями, входящими в данный объект. Так, для добавления нового документа можно воспользоваться коллекцией Documents. Эта коллекция включает в себя объекты Document, каждый из которых представляет документ Microsoft Word. Например, если строку

appWD.Documents.Add

добавить к трем предыдущим, то это приведет к появлению нового документа в окне текстового редактора.

В рассматриваемой процедуре часто используется объект ActiveDocument, который представляет собой активный документ, открытый в приложении Microsoft Word. С его помощью можно получить доступ к содержанию документа через так называемые внутренние объекты — абзацы, слова и т. д.

В предыдущем программном фрагменте после выполнения строки процедуры аppWD.Documents.Add автоматически произошло создание объекта ActiveDocument. Перейдем на более низкий уровень и посмотрим, как работать с объектами, составляющими документ. Так, для добавления нового абзаца в активный документ следует воспользоваться следующей конструкцией:

```
appWD.ActiveDocument.Paragraphs.Add
```

Здесь мы учли сделанную ранее ссылку appwD на объект Microsoft Word, а также использовали коллекцию Paragraphs, которая представляет собой все абзацы документа. А уже для этой коллекции использовали метод Add() для добавления нового абзаца в документ.

Программная вставка текста в документ (как в новый, так и в уже существующий) реализуется с помощью объекта Range, представляющего собой часть текста документа (например, созданный только что абзац). В приложении Microsoft Word объект Range представляет собой диапазон текста. Он может быть создан разными способами. Например, можно указать номер абзаца или начальный и конечный символы диапазона.

Примечание

Для работы с ячейками с использованием VBA в Microsoft Excel также имеется объект Range (что в переводе значит диапазон). Использование этого объекта требует задания в качестве параметра фрагмента ячеек, который нам нужен, например, Range ("a1") или Range ("a5:d8").

Вставка в документ текста с помощью объекта Range возможна с использованием двух методов — InsertBefore (что можно перевести как "вставить перед диапазоном") и InsertAfter ("вставить после диапазона"). Например, для вставки фрагмента текста можно использовать следующую конструкцию:

appWD.ActiveDocument.Paragraphs(1).Range().InsertAfter("текст 1")

У коллекции Paragraphs есть свойство Count, которое позволяет получить количество абзацев в документе. В следующей конструкции в переменной а подсчитывается количество имеющихся абзацев в активном документе:

a = appWD.ActiveDocument.Paragraphs.Count

Также с помощью объекта Range можно изменить шрифт текста в определенном абзаце. Например, для первого абзаца установим 14-й размер шрифта:

appWD.ActiveDocument.Paragraphs(1).Range().Font.Size = 14

Можно также задать стилевое оформление текста. Так, если первый абзац мы хотим оформить как заголовок второго уровня, то следует написать:

appWD.ActiveDocument.Paragraphs(1).Range().Style="Заголовок 2"

Листинг 8.16. Процедура обработки нажатия кнопки Передать в Word

```
Private Sub CommandButton1 Click()
Dim appWD As Object
' Открываем приложение Microsoft Word
Set appWD = CreateObject("Word.Application")
appWD.Visible = True
' Добавление нового документа
appWD.Documents.Add
appWD.ActiveDocument.Paragraphs(1).Range().InsertAfter("OTyet")
' Установка размера шрифта первого абзаца
appWD.ActiveDocument.Paragraphs(1).Range().Font.Size = 20

    Добавление нового абзаца

appWD.ActiveDocument.Paragraphs.Add
' Вставка в документ информации об интервале дат
Z = Range("d1").Value + " - " + Range("d2").Value
appWD.ActiveDocument.Paragraphs(2).Range().InsertAfter (Z)
appWD.ActiveDocument.Paragraphs(2).Range().Font.Size = 14
appWD.ActiveDocument.Paragraphs.Add
Z = Range("e1").Value + " - " + Range("e2").Value
```

Документ Microsoft Word, получаемый в результате выполнения процедур в среде 1С:Предприятие 8.1 и в приложении Microsoft Excel, мы уже видели (см. рис. 8.52).

Использование цветов в электронных формах

При рассмотрении примеров на протяжении всех глав книги мы не использовали возможностей цветовой гаммы при отображении данных в электронных формах. Однако информация в справочниках или регистрах может быть существенно информативнее для пользователя, если использовать различные цвета при оформлении. При этом цвета можно применять как для основного текста, так и для фона ячеек. При таком оформлении унылые численные данные могут смотреться более привлекательно. В этом разделе мы рассмотрим несколько примеров, которые помогут реализовать подобный эффект при решении прикладных задач.

Для определенности будем считать, что в конфигурации мы располагаем справочником товары, в котором укажем единственный дополнительный реквизит цена (выберем числовой тип данных). На рис. 8.53 показано окно редактирования объекта конфигурации, где следует выполнить необходимые установки.

Далее перейдем на закладку **Формы**, где создадим пользовательскую форму списка (рис. 8.54). Эта форма нам потребуется в связи с последующим написанием программных процедур, которые добавят оформительские эффекты в форму списка справочника.

Чередование цветов в формах справочников и документов

В качестве первой задачи по оформлению данных будем считать, что для формы списка справочника товары в режиме 1С:Предприятие необходимо обеспечить чередование цветов (строка в одном стиле сменяется строкой в другом стиле). Для

Основные Иерархия Владельцы ▶ Данные Нумерация Формы Макеты Подсистемы Ввсд на основании Права Интерфейсы Обмен данными Прочее	Длина кода Длина наименования Тип кода Число Строка Стро
--	---

Рис. 8.53. Окно редактирования объекта конфигурации

🚟 Справочник Товары: ФормаСписка 🛛 💶 🗙				
Справочник Товары				
Действия • 🔚 🛬 🔩 🍇 🏠 👔 🖆 🔞 • 🌾 🔃 🕐 💭				
Код Наименование Цена				
аы <Пол аы <Поле ввода> аы <Поле ввода>				
I Пиалог II Модиль II Реквизиты				
🗏 Диалог 🗏 Модуль 📑 Реквизиты				

Рис. 8.54. Закладка Формы окна редактирования объекта конфигурации

этого нам потребуется создать обработчик (программу) события приоткрытии для формы. Технические действия в этом случае заключаются в щелчке правой кнопкой по форме (именно по форме, а не по элементу, расположенному на ней) и выборе в контекстном меню пункта Свойства (рис. 8.55). Среди многообразия событий формы сейчас нас интересует событие приоткрытии. Поэтому щелкнем по изображению лупы напротив названия этого события, что позволит нам перейти в окно редактора программного кода. Здесь требуется написать программный код в соответствии с листингом 8.17.



Рис. 8.55. Закладка Свойства формы списка справочника Товары

Листинг 8.17. Процедура, выполняемая при открытии формы

```
Процедура ПриОткрытии()
```

```
СписокСтрок = ЭлементыФормы.СправочникСписок;
СписокСтрок.ЦветФонаЧередованияСтрок = WEBЦвета.Бирюзовый;
СписокСтрок.ЧередованиеЦветовСтрок = Истина;
КонецПроцедуры
```

Здесь мы обращаемся к табличному полю СправочникСписок на форме. Это обращение производится через коллекцию Элементыформы. У табличного поля имеется свойство цветфоначередованияСтрок, которое предназначено для задания фона чередующихся строк. Данное свойство доступно как для чтения, так и для записи. Кроме такой установки цвета для обеспечения чередования цветов требуется еще задать значение истина для свойства чередованиецветовСтрок.

После выполненных действий результат в режиме 1С:Предприятие выглядит так, как показано на рис. 8.56.

Рассмотрим теперь аналогичное оформление для табличной части документа. Для этого сначала создадим документ поступление (рис. 8.57). Здесь информация сосредоточена в табличной части, которая представляет собой перечень товаров. И наша задача заключается в обеспечении ее оформления с использованием чередования цветов.

Начало действий связано с созданием формы документа (выполняется с использованием конструктора на закладке **Формы**). После этого аналогично приведенной выше разработке формы справочника откроем окно свойств формы документа. Нам необходимо запрограммировать событие приоткрытии. Для этого с помощью щелчка по изображению лупы напротив названия этого события перейдем в окно редактора программного кода, в котором требуется написать программный код в соответствии с листингом 8.18.

🏪 1С:Предпри	🏪 1С:Предприятие (учебная версия) - Конфигурация 💦 💶 🗙				
🗄 <u>Ф</u> айл <u>П</u> равка	Операции <u>С</u> ервис <u>О</u> кна С	Справка			
I 🗋 📫 🔛	X 🖻 🖻 🧉 📰 🛛	🔜 🧟 м м+ м- 🎘 🖕			
🗐 Справочни	к Товары	_ 🗆 ×			
Действия 🕶 ု	i 🔁 🗟 🧏 🕅 To 🖷	- Tx 🖸 ?			
Код	Наименование 🔹	Цена 🔨			
- 00000003	Вешалка	2 500			
- 00000005	Кухня Уют	17 000			
00000001	Мягкая мебель	50 000			
00000004	Стенка Молодежная	45 050			
- 00000002	Стол	1 450			
🗐 Справочник Товары					
Для получения подсказки нажмите F1 САР NUM					

Рис. 8.56. Справочник Товары с использованием чередования цветов



Рис. 8.57. Форма документа Поступление

Листинг 8.18. Процедура, выполняемая при открытии формы документа

```
Процедура ПриОткрытии()
СписокСтрок = ЭлементыФормы.Перечень;
СписокСтрок.ЦветФонаЧередованияСтрок = WEBЦвета.БледноСиреневый;
СписокСтрок.ЧередованиеЦветовСтрок = Истина;
КонецПроцедуры
```

🏪 1С:Пре	📙 1С:Предприятие (учебная версия) - Конфигурация 📃 🗖 🗙				
<u>⊕</u> айл <u>П</u>	равка Опеј	рации <u>С</u> ервис <u>О</u>	кна Справка		
I 🗋 🚢 I	. X 🖻	s 🖻 🧉 🗋	* 📰 🖹 🐇	М М+	м- 🞇 🖕
🐱 Постуг	іление: По	оступление (Но	вый) *		_ 🗆 ×
Действия •	• 🛃 🛛	😼 🔁 🛃 (?		
Номер: 0	00000001				
Дата: О	7.12.2010 0	:00:00			
달달.	3, *2 🖬	↑↓輕鬆	2		
N Това	p	Цена	Количество	Сумма	
1 Мягк 2 Клун	ая мебель • Чют	50 000	2	100	000
3 Стол	1 3101	1 450	13		450
	ОК Записать Закрыть				
🚬 Документы Поступление 🛛 🐱 По: Поступление (Новый) *					
Для получе	ния подска	зки нажмите F1		CAP	NUM //

Рис. 8.58. Оформление документа Поступление

Результат выполнения данной процедуры после открытия формы документа в режиме 1С:Предприятие приведен на рис. 8.58. Видно, что подобное оформление табличной части документа делает его весьма наглядным (рис. 8.58).

Условное форматирование в формах

Рассмотрим другую ситуацию в плане использования цветов. Будем считать, что наша задача заключается в том, чтобы обеспечить выделение красным цветом тех цен товаров в справочнике товары, которые больше 3000. Для этого в режиме конфигуратора нам необходимо оформить (разработать программный код) другую стандартную программную процедуру — СправочникСписокПриВыводеСтроки. Данная процедура автоматически выполняется при последовательном выводе строк в форме списка справочника.

Начнем рассмотрение технических действий, и для начала нам необходимо открыть окно свойств элемента табличное поле (рис. 8.59), расположенного на форме. После этого в перечне событий присутствующих в окне свойств следует найти СправочникСписокПриВыводеСтроки и щелкнуть по изображению лупы напротив названия этого события. В результате мы попадем в окно редактора программного кода, где требуется написать текст программы в соответствии с листингом 8.19.

Листинг 8.19. Процедура, выполняемая при выводе строки списка

Процедура СправочникСписокПриВыводеСтроки(Элемент, ОформлениеСтроки, ДанныеСтроки) Если ДанныеСтроки.Цена > 3000 Тогда ОформлениеСтроки.Ячейки.Цена.ЦветТекста=WEBЦвета.Малиновый; Иначе

ОформлениеСтроки.Ячейки.Цена.ЦветТекста=WEBЦвета.ЗеленыйЛес; КонецЕсли

КонецПроцедуры

🛃 Конфигуратор (учебная версия) - Конфигурация 📃 🗆 🗙				
🛛 🙅 айл Правка Форма Конфигурация Отладка Администрирование Сервис Окна Справка				
🚟 Справочник Товары: ФормаСписка	Свойства: Табличное поле 🛛 🕂 🗙			
Справочник Товары	91 BI 1 × ∨			
Действия - 🔤 🔄 式 🍓 🏹 🎦 🕅 - 🌾	Имя СправочникСписок			
Код Наименование Цена аы <Пол аы <Поле ввода> аы <Поле в	Данные СправочникСписок ТипЗначения СправочникСписок.Тов. ▼ ИзменяетДан ПропускатьПр Первый8Групг			
	Видимость 🖌			
	Имя, по которому будет выполняться обрашение к элементу управления в			
Для получения подсказки нажмите F1 CAP	алгоритме модуля			

Рис. 8.59. Окно свойств табличного поля СправочникСписок

Результат выполненных действий после перехода в режим 1С:Предприятие приведен на рис. 8.60.

Рассмотрим теперь другую ситуацию. Допустим, нам нужно оформить ряд строк в несколько ином стиле. Скажем, требуется элементы справочника товары, которые

🏪 1С:Предприятие (учебная версия) - Конфигурация 🛛 💶 🗙						
<u>■ Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка						
I 🗋 🔟 📓 🔏 📾 💭 🎆 🛅 🧝 🕺 📰 🛅 🧟 M. M+ M- 🎘 🖵						
🗐 Справочник 1	овары	_				
Действия • 😫 🚔 🐳 🎢 Та Та 🖓 - Ту 🔃 ?						
Код	Наименование 🔻	Цена				
- 000000003	Вешалка	2	500			
- 000000005	Кухня Уют	17	000			
- 000000001	Мягкая мебель	50	000			
- 000000004	Стенка Молодежная	45	050			
- 000000002	Стол	1	450			
			~			
🗐 Справочник Товары						
Для получения подсказки нажмите F1 САР NUM						

. Рис. 8.60. Вид документа с условным оформлением при выводе цен

содержат цены более определенного уровня (более 3000 рублей), выделять по высоте ячейки. Для этого изменим текст процедуры СправочникСписокПриВыводеСтроки в соответствии с листингом 8.20. В этой разработке мы воспользовались еще одним свойством — высотаячейки. Результат выполненных действий после перехода в режим 1С:Предприятие приведен на рис. 8.61.

Листинг 8.20. Процедура, выполняемая при выводе строки списка

Процедура СправочникСписокПриВыводеСтроки(Элемент, ОформлениеСтроки, ДанныеСтроки) Если ДанныеСтроки.Цена > 3000 Тогда ОформлениеСтроки.Ячейки.Цена.ВысотаЯчейки=2; КонецЕсли КонецПроцедуры

Таким образом, рассмотренные примеры показали наличие ресурсов в плане использования цветов по оформлению табличных частей справочников и документов. Используя эти возможности, можно придать существенно большую информативность выводимым числовым данным. В этом случае пользователю достаточно одного взгляда на табличную часть, чтобы получить необходимую информацию. В качестве практики можно самостоятельно применить рассмотренные ресурсы для справочников и документов, разработанных в предыдущих главах.

	🏪 1С:Предприятие (учебная версия) - Конфигурация 🛛 💶 🗙					
≣.,	🛛 🛨 айл Правка Операции Сервис Окна Справка					
Ē	I 🗋 🔔 Х 🖻 🛍 🥌 🍟 I 🎹 🛅 🧝 м м+ м- 🎘 🖕					
2	🗐 Справочник Товары 📃 🗆 🗙					
	Дей	ствия 🗸 😫 🕇	5 🗟 🏂 🕅 To 🖷	• 🕅 🛛 🖉		
		Код	Наименование 🔹	Цена	^	
	-	000000003	Вешалка		2 500	
	-	000000005	Кухня Уют		17 000	
	-	000000001	Мягкая мебель		50 000	
	-	000000004	Стенка Молодежная		45 050	
	-	000000002	Стол		1 450	
				<u> </u>	~	
ду Справочник Говары						
Для получения подсказки нажмите F1 САР NUM						

Рис. 8.61. Пример условного оформления ячеек по высоте в списке справочника

глава 9



Анализ данных

Ряд уже рассмотренных в книге примеров касался в той или иной степени анализа данных, имеющихся в информационной базе. При этом мы самостоятельно создавали запросы на внутреннем языке запросов и формировали вывод информации, используя объекты системы 1С:Предприятие 8. Однако в эту систему заложены специальные возможности для проведения анализа имеющейся информации в базе данных.

Анализ данных позволяет увидеть явно не бросающиеся в глаза закономерности в имеющихся данных и представить их в виде легко воспринимаемого результата [11]. Понятно, что при небольшом объеме данных закономерности найти не очень сложно. Если количество данных значительное, но при этом закономерности очевидны, то особых средств автоматизации также не требуется. Наиболее сложный случай связан с нахождением неочевидных закономерностей при большом объеме информации. Эта ситуация наиболее сложная, и здесь средства автоматизации позволяют получить существенную помощь. После этого результат анализа можно эффективно использовать при принятии решений управленческим составом организации.

Для подчеркивания значимости анализа данных приведем следующий пример. Так, будем считать, что в нашей организации имеется несколько филиалов, в которых менеджеры принимают заказы от клиентов. Так или иначе профессионализм менеджеров влияет на товары, которые они рекомендуют клиентам. И результат анализа может показать, что наиболее технологичные модели эффективно продаются менеджерами только в некоторых филиалах. Подобный результат, разумеется, важен для руководства компании. Это позволяет внести необходимые коррективы в работу фирмы.

В анализе данных можно выделить несколько составляющих. Источник данных представляет собой исходную информацию для анализа. В качестве источника может выступать результат запроса, область ячеек табличного документа или таблица значений.

Тип анализа определяет вид результата, к которому должны быть сведены исходные данные. В системе 1С:Предприятие поддерживаются пять различных типов анализа:

- общая статистика, которая позволяет сформировать общую статистическую информацию об источнике данных (количество значений и количество уникальных значений; минимальное, максимальное и среднее значения; размах, среднее отклонение, медиана, мода, частота появления каждого значения в источнике данных);
- поиск ассоциаций, который предназначен для поиска часто встречающихся групп характеристик объектов и формирования правил ассоциации (например, поиск ассоциаций позволяет сформировать список менеджеров, которые чаще других продают товары определенной категории);
- поиск последовательностей, применяемый для выявления цепочки событий, часто наблюдаемых в источнике данных (наиболее показательный пример связан с формированием набора товаров, которые последовательно приобретают наши покупатели);
- дерево решений, которое предназначено для выявления закономерностей того, что объект относится к тому или иному классу;
- кластерный анализ, позволяющий объединить объекты в группы (кластеры), в которых могут находиться объекты, наиболее схожие по характеристикам.

Объект анализданных является объектом встроенного языка, который непосредственно и выполняет анализ данных. Для него устанавливается источник данных, задаются параметры и производится настройка колонок. Итогом работы этого объекта является *результат анализа данных*, тип которого зависит от типа анализа. *Результат анализа данных* — объект встроенного языка, содержащий информацию о результатах анализа. Заметим, что для каждого типа анализа существует свой объект для работы с результатом анализа:

- РезультатАнализаДанныхОбщаяСтатистика;
- РезультатАнализаДанныхПоискАссоциаций;
- РезультатАнализаДанныхПоискПоследовательностей;
- РезультатАнализаДанныхДеревоРешений;
- РезультатАнализаДанныхКластеризация.

Полученный результат анализа может быть выведен в табличный документ при помощи построителя отчета анализа данных.

Далее в этой главе мы на примере рассмотрим использование ряда из объектов. Для этого предварительно создадим новую конфигурацию.

Создание новой конфигурации

Тематика разрабатываемой информационной базы будет связана с оказанием нашей фирмой ряда услуг — остекление окон, замена сантехники, установка дверей, строительство дачных домиков и т. д. Будем считать, что в нашей фирме несколько филиалов, где оказываются данные услуги.

Для начала в новой конфигурации мы создадим справочники услуги (рис. 9.1), Филиалы (рис. 9.2) и Клиенты (рис. 9.3).

10 V8	1C:	Предприятие (у	чебная версия) - Конфигурация			Ľ	
1	<u>∎ Ф</u> айл <u>П</u> равка Операции <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка						
1		📫 🖬 🐰 🖻	🛍 🧉 🙋 🐥 📰 📰 🧝 🛛	I M+	м- 🖇	- %	
🗐 Справочник Услуги 💶 🗖 🗙							
ŀ	Дейс	ствия 🕶 😫 🛃	🛃 😽 🕅 To 🗃 🐨 🔯 🕐				
		Код	Наименование		•		
	-	000000005	Замена сантехники				
	-	000000002	Подвесные потолки			-	
	-	000000003	Установка дачных домиков				
	-	000000004	Установка дверей				
	-	000000001	Установка окон				
						-	
						_	
Ð	7 Oτ·	ют Отч 🚠 Табл	ица 📇 Документ 🙀 Таблица	đ	Справо	чн	
Д	Для получения подсказки нажмите F1 САР NUM				1.		

Рис. 9.1. Заполнение справочника Услуги

🏪 1С:Предприятие (у	учебная версия) - Конфигурация 📃 🔲 🗙				
<u>Файл Правка Операции Сервис Окна Справка</u>					
🗋 🛋 🖬 🗶 🖻	💼 🥌 💐 📲 🛅 🧝 M M+ M- 🎇 🗸				
🗐 Справочник Филиалы 💶 🗆 🗙					
Действия 🕶 🔄 불	🛋 🚰 🕅 Te 🖷 - Tr 🛛 🔃 ?				
Код	Наименование 🔹 🔿				
- 000000003	Новый				
- 000000001	Окский				
- 000000002	Центральный				
💷 Отчет 🚠 Таблица 🚢 Докум 🚠 Таблица 🍠 Справо 🗐 Справо					
Для получения подсказ	ки нажмите F1 CAP NUM				

Рис. 9.2. Заполнение справочника Филиалы

Для оформления заказов на услуги нам понадобится документ продажатоваров (рис. 9.4). В основной части документа разместим два поля:

- Заказчик ТИП ДАННЫХ СправочникСсылка. Клиенты;
- 🔶 Филиал ТИП ДАННЫХ СправочникСсылка.Филиалы.

Табличная часть также будет содержать два реквизита:

- Услуга ТИП Данных СправочникСсылка.Услуги;
- Стоимость числовой тип данных.

Фиксирование услуг в базе данных реализуем с помощью регистра накопления РегистрацияПродаж (сделаем его регистром оборотов). На рис. 9.5 показана структура данного регистра накопления. В качестве измерений регистра укажем — Услуга, Филиал и Заказчик. Они будут соответствовать ранее созданным справочникам.



Рис. 9.3. Заполнение справочника Клиенты



Рис. 9.4. Заполнение документа ПродажаТоваров

Движения по регистру реализуем с помощью проведения документа продажатоваров. Для этого вернемся к работе с данным документом и на закладке Движения создадим необходимую программную процедуру. При этом мы воспользуемся уже знакомым конструктором движения регистра (рис. 9.6). Результат работы конструктора представлен в листинге 9.1.



Рис. 9.5. Структура регистра Регистрация Продаж

Конструктор двих	кения ре	гистров			□ × □	
열혈혈수부						
Регистры			Реквизиты документа			
💽 РегистрНакопл	ения.Реги	страцияПродаж	🕳 Дата			
			🗕 Номер			
			🗕 Заказчик			
			😑 Филиал 🔄 ТекСтрокаПеречень.Услуга			
			🚛 ТекСтрокаПеречень.(Стоимость		
Табличная часть:	Перечень	· ·		< <u>Н</u> азад	<u>Д</u> алее>	
Поле		Выражение		<u>З</u> аполнить	выражения	
🛴 Услуга		ТекСтрокаПеречень.Услуга		0		
ҍ Филиал		Филиал		Очистить выражения		
🛴 Заказчик 🛛 Заказчик		Заказчик				
🥤 Сумма ТекСтрокаПеречень.Стои		мость	ОК			
				Отм	иена	
				Спр	авка	

Рис. 9.6. Окно конструктора движения регистра
Листинг 9.1. Процедура обработки проведения документа Продажа Товаров

Процедура ОбработкаПроведения (Отказ, Режим) Для Каждого ТекСтрокаПеречень Из Перечень Цикл Движение = Движения.РегистрацияПродаж.Добавить(); Движение.Период = Дата; Движение.Услуга = ТекСтрокаПеречень.Услуга; Движение.Филиал = Филиал; Движение.Заказчик = Заказчик; Движение.Сумма = ТекСтрокаПеречень.Стоимость; КонецЦикла; КонецПроцедуры

На этом этап, связанный с начальной разработкой конфигурации, завершен. Теперь осталось в режиме 1С:Предприятие внести (с проведением) несколько документов. Информация, содержащаяся в документах, автоматически будет фиксироваться в созданном регистре накопления (рис. 9.7).

1 0 V8	1C:	Предприятие (учебная вер	сия) - К	онфигу	рация			<u> </u>					
	<u>Ф</u> ай	ил <u>П</u> равка Операции <u>С</u> ервис	<u>О</u> кна	Справка									
		🛋 🖬 X 🖻 🛍 🍝 🖻	15	et 24		▼	5 🙄 📰 🗎	🤶 М М+ М- 🎘 🖵					
1	📭 Список Регистрация продаж 📃 🗆 🗙												
II.	Действия • [++] 🏦 🛅 🖷 • 🦒 🙋 🕐												
		Регистратор	Номе	Актив	Период 🔻	Услуга	Филиал	Заказчик 🔼					
	•	Продажа товаров 000000002	1	~	01.06.2009 0:00	Установка дачных	Новый	Сафронов Д.В.					
	٠	Продажа товаров 000000001	1	~	09.06.2009 0:00	Установка дачных	Окский	Колесов Д.М.					
	•	Продажа товаров 000000004	1	~	21.07.2009 10:2	Установка окон	Новый	Сафронов Д.В.					
	•	Продажа товаров 000000003	1	~	21.07.2009 10:3	Установка окон	Окский	Сафронов Д.В.					
	•	Продажа товаров 000000005	1	~	21.07.2009 10:3	Установка окон	Окский	Колесов Д.М.					
	יד0 7	чет Отче ≩ Таблица 🛛 🔮	<mark>\</mark> Докум	енты 🛓	Таблица 🛃	🗊 Справочни 🗐 Сп	равочни 🧐 Справо	очни 👔 Список Рег					
Д	ля п	олучения подсказки нажмите F	1					CAP NUM					

Рис. 9.7. Форма списка регистра Регистрация Продаж

Общая статистика

Таким образом, мы подготовили необходимую базу для проведения анализа. В этом разделе главы созданная информационная база будет использоваться для иллюстрации типа анализа, который называется общей статистикой.

Общая статистика представляет собой механизм сбора общей информации о данных, находящихся в полученном источнике данных. Этот тип анализа предназначен для предварительного исследования анализируемой информации. А для получения информации по статистике следует использовать уже упоминаемый объект АнализДанных. Исходные данные для статистического анализа могут содержать информацию двух видов:

- непрерывные данные;
- дискретные данные.

Непрерывный вид данных используется в том случае, если анализируемые данные являются числовыми или имеют тип дата. При анализе данных другого типа используется вид — дискретный.

Рассмотрим получение общей статистики на примере. Для демонстрации статистического анализа нам понадобится отчет, и в режиме конфигуратора начнем создание нового отчета, который назовем отчет9_1. Система компоновки данных, рассмотренная в *главе* 7, здесь нам не потребуется. Фактически все основные действия развернутся на закладке **Формы**, на которой мы и создадим простую форму отчета (рис. 9.8).



Рис. 9.8. Форма для получения отчета

Далее от нас потребуется на закладке Модуль разместить текст процедуры, которая приведена в листинге 9.2.

Листинг 9.2. Процедура формирования отчета с общей статистикой



Анализ данных

Запрос = Новый Запрос; Запрос.Текст = "ВЫБРАТЬ Услуга, Филиал, Заказчик, Сумма | ИЗ РегистрНакопления.РегистрацияПродаж КАК Продажи"; Анализ.ИсточникДанных= Запрос.Выполнить(); РезультатАнализа = Анализ.Выполнить(); Построитель=Новый ПостроительОтчетаАнализаДанных(); Построитель.Макет=Неопределено; Построитель.ТипАнализа=Тип("АнализДанныхОбщаяСтатистика"); ТабДок = Новый ТабличныйДокумент; Построитель.Вывести(РезультатАнализа, ТабДок); ТабДок.Показать(); КонецПроцедуры

После размещения процедуры перейдем в режим 1С:Предприятие и сформируем необходимый отчет, фрагмент которого представлен на рис. 9.9.

<mark>12</mark> V8	С:Пре,	дприя	тие	(уч	ебная версия) - Конфигура	ция										_	
<u>Φ</u>	айл ∏р	равка	Tat	блица	а Операции <u>С</u> ервис <u>О</u> кна (Справка											
	1		6 8	6	a 🤹 🖸 to i et i m 🗌			-] n 'n	🖻 🖻		🗎 🧟 🛛	а м	+ M-	- %		
A	Табли	ица															
	230		1	2	3			1	4	5	1	6		7	1	8	^
		1															_
		2		0	бщая статистика	3											
		3															
曱	4 Информация о данных																
	5 Количество объектов:								5								
1 ^L		7	1														
曱		8		He	прерывные поля												
		10	j		Поле	Значен	ий	Минимум Максиму		імум	мум Средне			Разк	łax	Ст	
L.		11		Сум	Ma		5		30 000		200 000) 9	6 400	,0000	1	70 000	70
		12		-												_	
ΠĘ		13		ди	скретные поля												
6	Ę	15		Усл	пуга												
		17			Количество значений:				5								
		18	1		Количество уникальных значе	эний:			2								
		19			Мода:			Установ	ка окон								
		20		Таб	блица частот												
	IΎ	21		Tak	лица частот	I											
		23			Значение		Час	тота	Относи час	тельная тота	Наког ча	тленная стота	0	Нак тносит	оплен ельная	ная і частої	ra
		24		Уста	ановка окон			3		60,00			3			60	,00
	L	25		Уста	ановка дачных домиков			2		40,00			5			100	,00}
		26	1			1											~
E	_	_	-	-			_						_				
	Рорма				🚛 Габлица												
Для	получе	ения по	дск	азки	нажмите F1										C4	NU	M //

Рис. 9.9. Фрагмент отчета с общей статистикой

Проанализируем сначала содержание отчета, а затем разберем текст программного кода. Для непрерывных и дискретных данных рассчитываются различные статистические показатели. Так, в случае непрерывных данных производится следующий расчет:

 количество значений — количество значений, присутствующих в исходной выборке;

- минимальное значение минимальное значение, присутствующее в исходной выборке;
- максимальное значение максимальное значение, присутствующее в исходной выборке;
- ◆ *среднее значение* среднее арифметическое значение выборки;
- ◆ *размах* разность между максимальным и минимальным значением выборки;
- стандартное отклонение среднее квадратичное отклонение, равное корню квадратному из дисперсии выборки;
- *медиана* значение, лежащее в середине выборки, упорядоченной по возрастанию или убыванию.

На рис. 9.9 эти показатели рассчитаны для поля сумма по имеющимся записям в регистре накопления. Данные записи соответствуют строкам табличной части введенных документов.

Для дискретных данных рассчитываются:

- количество значений общее количество значений, присутствующих в исходной выборке;
- количество уникальных значений количество уникальных значений, присутствующих в исходной выборке;
- ◆ *мода* значение, наиболее часто встречающееся в исходной выборке.

Кроме того, для дискретных значений формируется таблица частот, содержащая следующие показатели:

- ♦ частота количество вхождений уникального значения в выборку;
- относительная частота частота, выраженная в процентах от общего количества значений выборки;
- накопленная относительная частота сумма относительной частоты и относительных частот всех предыдущих значений выборки.

В отчете на рис. 9.9 дискретными являются поля — Услуга, Филиал и Заказчик. В представленном фрагменте информация по филиалам и заказчикам не видна, однако если в режиме 1С:Предприятие прокрутить окно со сформированным отчетом, то мы все увидим.

Теперь перейдем к рассмотрению программного кода, приведенного в листинге 9.2. Ключевую роль в процедуре играет объект Анализданных, который предназначен для выполнения анализа входной информации. В первой строке

Анализ = Новый АнализДанных;

создается новый объект, а далее используется свойство данного объекта типАнализа, значение которого определяет тип проводимого анализа. В рассматриваемом случае нас интересует общая статистика:

Анализ.Типанализа=Тип("АнализДанныхОбщаяСтатистика")

После этого формируется запрос к регистру продажи:

Запрос.Текст = "ВЫБРАТЬ Услуга, Филиал, Заказчик, Сумма | ИЗ РегистрНакопления.РегистрацияПродаж КАК Продажи"; Результат запроса устанавливается в качестве источника данных анализа. Сам анализ выполняется в результате работы метода Выполнить (). Анализ не имеет средств по визуализации результата полученного анализа. Для этой цели используется объект построительОтчетаАнализаДанных. При создании данного объекта повторно указывается тип проводимого анализа. В рассматриваемой процедуре мы используем метод вывести (), в качестве первого параметра которого передается результат полученного анализа.

Построитель отчета анализа данных автоматически создает круговую диаграмму (рис. 9.10) по относительной частоте в выборке.



Рис. 9.10. Диаграмма с показателями статистики

Поиск ассоциаций

Если указать в качестве типа анализа значение АнализданныхПоискАссоциаций, то мы обеспечим поиск часто встречающихся вместе объектов или значений характеристик. Этот тип анализа можно использовать для определения часто приобретаемых вместе товаров. Также в этом случае можно провести связь между образованием менеджера и предлагаемыми им товарами. В качестве источника данных может использоваться результат запроса, таблица значений или область ячеек табличного документа.

В плане данного типа анализа колонки источника данных можно разделить на:

- НеИспользуемая (игнорируется анализом);
- Объект (данные из этой колонки используются как объекты для приводимого анализа; исходя из значения данной колонки, значения колонки Элемент относятся к одной ассоциативной группе);
- Элемент (данные из этой колонки используются для получения устойчивых групп значений и построения ассоциативных правил).

Кроме настройки типов колонок, на результат проводимого анализа влияют следующие параметры:

- ♦ МинимальныйПроцентСлучаев (число) минимальный процент случаев, в которых наблюдается группа элементов (найденные группы, в которых процент случаев меньше, в отчет не включаются);
- МинимальнаяДостоверность минимальная достоверность правила (найденные правила, у которых процент случаев меньше, в отчет включены не будут);
- Минимальная Значимость минимальная значимость правила (найденные правила, значимость которых меньше данного значения, в отчет включены не будут; значимость правила величина, характеризующая, насколько правило важно).

Рассмотрим пример, который позволяет выявить оказываемые вместе услуги. Для этого мы создадим еще один отчет. В нем нам просто потребуется скорректировать процедуру из листинга 9.2. Ее новый вариант представлен в листинге 9.3. Признаком, по которому услуги относятся к одной группе, будем считать значение регистратора (услуги, указанные в одном документе, считаются оказанными одновременно). Здесь Регистратор будет объектом анализа, а услуга — элементом.

Листинг 9.3. Процедура формирования отчета с поиском ассоциаций

```
Процедура КнопкаСформироватьНажатие (Кнопка)
Анализ = Новый АнализДанных;
Анализ.ТипАнализа=Тип ("АнализДанныхПоискАссоциаций");
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Продажи.Регистратор, Продажи.Услуга
| ИЗ РегистрНакопления.РегистрацияПродаж КАК Продажи";
Анализ.ИсточникДанных= Запрос.Выполнить();
РезультатАнализа = Анализ.Выполнить();
Построитель=Новый ПостроительОтчетаАнализаДанных();
Построитель.Макет=Неопределено;
Построитель.ТипАнализа=Тип ("АнализДанныхПоискАссоциаций");
ТабДок = Новый ТабличныйДокумент;
Построитель.Вывести (РезультатАнализа, ТабДок);
ТабДок.Показать();
КонецПроцедуры
```

В результате выполненных действий, когда мы перейдем в режим 1С:Предприятие и сформируем отчет, получим услуги, оказываемые вместе (рис. 9.11).

	1С:П	редпр	грият	гие	(учебная версия) - Конфигура	ция									1×
1	<u>₽</u> айл	Правн	<a< td=""><td>Таб.</td><td>пица Операции <u>С</u>ервис <u>О</u>кна С</td><td>Справка</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></a<>	Таб.	пица Операции <u>С</u> ервис <u>О</u> кна С	Справка									
Ī] 4		Ж	B	a 🖻 🔿 Q 🖕 C 🛛 M		▼ 38	` @0	î 🖻 📴 🖕 🛙		і 🌠 м	M+ M-	℀.		
A	Tat	і́лица												_ 🗆	×
1	23		1	2	3		4		5		6	7		8	_ ^
lĘ)	28 29		Чa	сто встречаемые группі	<u>61</u>									
		30		No	Количество станзев По										-
		32		Cor	тав	odeni eviyae									- 1
	Θ	33			1 3	60,0	00								- 1
		34	j)	слуга = Установка окон										
	L	35		3	слуга = Подвесные потолки										
	曱	36			2 2	40,0	00			_					_
		37)	(слуга = Установка дачных домико)B	-			_					_
L r	. L	38		ر	слуга = установка двереи										
Ę	9	40		Ac	социативные правила			+							
		41		Hev	ดาษมพ์ ผลกิดก	Спелствие			Процент с п	aaen	Постове		Jauminor	ть	7
	Ð	43		1	одный насор	следствие			inpodent est	60.00	достове	100.00	ia maoa	1.67	+
	T	44	1)	слуга = Установка окон	Услуга = П	одвесные потол	ки							†
	Ō	45	Í	2						60,00		100,00		1,67	
		46)	(слуга = Подвесные потолки	Услуга = У	становка окон								
	曱	47		3						40,00		100,00		2,50	4
		40		1	(слуга = Установка дачных	Услуга = У	становка дверей	4							
	L	48		 Д	UNINFOR					40.00		100.00		2 50	÷ I
	Ϋ́	49		- 1	слуга = Установка дверей	Услуга = У	становка дачны	×	-	40,00		100,00		2,00	+
		50				домиков									
		51	—		[~
<														[>
	7 Отче	т Отче	т2		📇 Документы Продаж	а товар 🗼 Т	аблица								
Д	ля пол	учения	под	іска	зки нажмите F1								CAP	NUM	///

Рис. 9.11. Отчет с отражением ассоциаций товаров

Поиск последовательностей

Этот вид анализа позволяет выявить цепочки возникающих событий. Он может использоваться тогда, когда одним из важных анализируемых показателей является последовательность наступления событий во времени. Например, можно выявить последовательность услуг, которые оказываются друг за другом в течение какоголибо определенного промежутка времени.

В качестве источника данных может использоваться результат запроса, таблица значений, область ячеек табличного документа. С точки зрения данного типа анализа колонки источника можно разделить:

- НеИспользуемая (игнорируется анализом);
- Последовательность (данные из этой колонки используются для анализа как объект события последовательности; по значению данной колонки анализ и ассоциирует данные с одной цепочкой событий);
- Элемент (данные из этой колонки используются как элементы последовательности);
- *Время* (по данной колонке определяется время наступления событий).

Кроме настройки типов колонок, на результат проводимого анализа влияют следующие параметры анализа:

- МинимальныйПроцентСлучаев (данный параметр представляет собой минимальный процент последовательностей, в которых наблюдается найденная закономерность);
- МинимальныйИнтервал (признак установки минимального интервала последовательности; должна быть установлена единица измерения интервала, кратность);
- МаксимальныйИнтервал (признак установки максимального интервала последовательности; должна быть установлена единица измерения интервала, кратность);
- ИнтервалЭквивалентностиВремени (признак установки интервала эквивалентности времени; должна быть установлена единица измерения интервала, кратность);
- МинимальнаяДлина (минимальная длина искомых последовательностей);
- ПоискПоИерархии (признак осуществления поиска по иерархии; распространяется на колонки с типом Элемент).

Ряд свойств связан с таким понятием, как типЕдиницыИнтервалаВремениАнализаданных. Это системное перечисление содержит следующие значения:

- ♦ минута ТекущаяМинута;
- ♦ час Текущийчас;
- день Текущийдень;
- неделя ТекущаяНеделя;
- декада ТекущаяДекада;
- месяц ТекущийМесяц;
- квартал ТекущийКвартал;
- полугодие ТекущееПолугодие;
- ГОД ТекущийГод.

Рассмотрим использование поиска последовательностей на примере нашей информационной базы. Данные из колонки Заказчик будут определять принадлежность к определенной цепочке событий (определяют последовательность анализа). Поле Услуга будет являться элементом получаемой последовательности. В листинге 9.4 показана процедура, которую необходимо разместить вместо предыдущей в модуле формы нашего отчета.

Результат выполнения процедуры показан на рис. 9.12, где указан интервал между двумя последовательными услугами.

Листинг 9.4. Процедура формирования отчета с поиском последовательностей

Процедура КнопкаСформироватьНажатие(Кнопка) Анализ = Новый АнализДанных; Анализ.ТипАнализа=Тип("АнализДанныхПоискПоследовательностей");

Анализ данных

Запрос = Новый Запрос; Запрос.Текст = "ВЫБРАТЬ Заказчик, Услуга, Период | ИЗ РегистрНакопления.РегистрацияПродаж"; Анализ.ИсточникДанных= Запрос.Выполнить(); Анализ.НастройкаКолонок.Период.ТипКолонки= ТипКолонкиАнализаДанныхПоискПоследовательностей.Время; РезультатАнализа = Анализ.Выполнить(); Построитель=Новый ПостроительОтчетаАнализаДанных(); Построитель.Макет=Неопределено; Построитель.ТипАнализа=Тип("АнализДанныхПоискПоследовательностей"); ТабДок = Новый ТабличныйДокумент; Построитель.Вывести(РезультатАнализа, ТабДок); ТабДок.Показать(); КонецПроцедуры

10	1С:П	редпр	ият	гие (учебн	ая версия) - Конфигура	ация					
Ī	<u>Ф</u> айл	Правк	a	Таблица О	перации <u>С</u> ервис <u>О</u> кна	Справка					
=	n <u>-</u> 1		X	B. A.			▼ 33 5	a 🖪 🖻		2 M M+ M-	43
			0/0					*** "!;	▼ - = = = = = = = = = = = = = = = = = =	ME IN IN IN	
Į	🖥 Tat	блица									_ 🗆 ×
<u>(</u>	123		1	2	3		4	5		6	7 🔼
		23	1	N≘	Количество случаев	Процент с	лучаев		Средний	Минимальный	Максимальн
		24	j	Состав			•		интервал	интервал	интервал
	P	25		1		2		100,00			
				Услуга =	• Установка дачных домин	(ОВ,					
		26		Услуга =	 Установка двереи Установка окон 				1	1 1 1 1 1 2 1	1 400 21 4
		27		Yonyra -	- Лолвесные потолки				г мес. глд.	т мес. тэ д.	Пиес. 21 д.
	۱ <u>۵</u>	28	í	2		1		50,00			
				Услуга =	 Установка дачных домин 	(ОВ,					
		29		Услуга =	= Установка дверей						
		30		Услуга=	Подвесные потолки				1 мес. 21 д.	1 мес. 21 д.	1 мес. 21 д.
	님	31		Услуга =	= Установка окон Г	4		CO 00	20 c.	20 c.	20 c.
	17	32		J Venura -		1		50,00			+ III
		33		Услуга =	= Установка дверей						
		34	í	Услуга =	- Установка окон				1 мес. 21 д.	1 мес. 21 д.	1 мес. 21 д.
	LL	35		Услуга =	Подвесные потолки				20 c.	20 c.	20 c.
		36									
		37									
		38									
	<	aĦ		· ·			1	1			Σ
P	<mark>\</mark> Доку	менты	Про	одажа товар	💷 Отчет Отчет3	A. I	Габлица				
Д	іля пол	учения	под	цсказки наж	кмите F1						CAP NUM

Рис. 9.12. Отчет с отражением последовательностей услуг

Статистика в учебном центре

В этом разделе мы рассмотрим пример, касающийся подсчета ряда статистических показателей. Для выполнения практических действий, связанных с демонстрацией анализа данных, создадим новую конфигурацию. Тематика разрабатываемой информационной базы будет связана со статистикой записи учащихся на различные учебные курсы, которые предлагает наш условный учебный центр. Первым объектом, который мы создадим в новой конфигурации, будет справочник ВидРаботы, отражающий виды трудовой деятельности преподавателей нашего учебного центра. Справочник не будет иметь дополнительных реквизитов, и его заполнение в режиме 1С:Предприятие показано на рис. 9.13.

🏪 1С:Предприятие	(учебная версия) - Конфигурация 📃 🗵 🗙											
🗄 <u>Ф</u> айл <u>П</u> равка Опе	рации <u>С</u> ервис <u>О</u> кна Сп <u>р</u> авка											
🛭 🗋 🝙 X 🖻 🛍 🚳 🎽 📰 🛅 餐 M M+ M- 🎘 🧅												
🗐 Справочник Вид работы 📃 🗆 🗙												
Действия • 🔄 🛬 ≼ 🍇 🏹 🎦 🗃 • 🌾 😥 🕐												
Код Наименование 🕶 🖂												
00000001	Основная											
- 00000002	По совместительству											
Справочник вид расоты												
Для получения подска	зки нажмите F1 CAP NUM //											

Рис. 9.13. Заполнение справочника ВидРаботы

Будем считать, что в нашем учебном центре имеются два основных вида работы преподавателей:

- по совместительству;
- основная.

Далее создадим еще один справочник — преподаватели (рис. 9.14), который также не будет иметь дополнительных реквизитов.

🏪 1С:Предприятие	учебная версия) - Конфигурация 📃 🗖 🗙											
🛛 <u>Ф</u> айл <u>П</u> равка Опер	🗄 🙅 айл Правка Операции Сервис Окна Справка											
I 🗋 🛁 🔛 🔏 📾 📾 🖉 🗮 📰 📰 🧟 M M+ M- 🎘 🧅												
🗐 Справочник Преподаватели 📃 🗆 🗙												
Действия 🕶 😫 🛬	🛃 🚰 🎁 🛅 - 🏹 🔯 🕐											
Код	Наименование 🔹 🔼											
- 00000003	Аникина Л.В.											
- 000000002	Иванов П.С.											
- 000000001	Петров О.В.											
🗐 Справочник Вид раб	оты 🎒 Справочник Преподаватели											
Для получения подсказки нажмите F1 CAP NUM												

После этого добавим справочник курсы, в котором разместим один дополнительный реквизит — Преподаватель (тип данных — СправочникСсылка.Преподаватели). И еще создадим справочник учащиеся, в котором не нужно добавлять дополнительные реквизиты.

Кроме справочников, нам потребуется документ ЗаписьнаКурсы. На рис. 9.15 представлены реквизиты, которые следует в указанном документе разместить. Здесь Учащийся имеет тип данных СправочникСсылка.Учащиеся, Курс — Справочник-Ссылка.Курсы, а числочасов — числовой тип данных. В данном документе определяется информация о количестве часов занятий конкретных курсов, которые собираются посещать учащиеся.

🛃 Конфигуратор (учебная в	ерсия) - Конфигурация 📃 🔲 🗙										
<u>Ф</u> айл <u>П</u> равка Конфигурация	Отладка Администрирование <u>С</u> ервис <u>О</u> кна										
Справка											
🗟 Документ ЗаписьНаКурс	сы _ 🗆 🗙										
Основные											
🕨 Данные											
Нумерация	Е Реквизиты										
Движения											
Последовательности	1 1 2 3 3 1 1 4 5 1 1 5 1 1 1 1 1 1 1 1 1 1 1 1 1										
Журналы											
Формы	Н-11 І абличные части										
Макеты	— Цуанцийся										
Подсистемы											
	— ЧислоЧасов										
A V											
Действия 👻 <Назад	Далее> Закрыть Справка										
Е! Служебные сообщения											
Для получения подсказки нажми	rre F1 CAP NUM ru 🗸 🥢										

Рис. 9.15. Закладка Данные окна редактирования документа Запись НаКурсы

Для фиксации записей учащихся на курсы мы создадим еще один объект конфигурации — регистр накопления _{Нагрузка}. На закладке **Основные** укажем, что данный регистр будет регистром оборотов. Представление об измерениях и ресурсе регистра дает рис. 9.16.

Теперь необходимо обеспечить проведение документа — формирование движений по регистру накопления. Для этого в окне редактирования объекта конфигурации перейдем на закладку Движения. Здесь отметим в списке регистров конфигурации единственный регистр, который мы создали. После отметки регистра сразу становится доступным конструктор движений, которым мы и воспользуемся. С помощью кнопки Заполнить выражения напротив названий полей регистра формируются соответствующие выражения (рис. 9.17).

Теперь все действия в окне конструктора выполнены, осталось только щелкнуть по кнопке **OK**. Это приводит к генерации конструктором программной процедуры

<mark>∦Конфигуратор (учебн</mark> Файл Правка Конфигу Сп <u>р</u> авка	ная версия) - Конфигурация(_)× рация Отладка Администрирование Сервис Окна
Регистр накоплени Основные	ия Нагрузка _ 🗆 🗙
 Данные Регистраторы Формы Макеты Подсистемы Права Интерфейсы 	В Измерения ↓ Учащийся ↓ Курс Ресурсы ЧислоЧасов Реквизиты
Обмен данными Действия 👻 <Наза	ад Далее> Закрыть Справка
Служебные сообщения Для получения подсказки і] нажмите F1 САР NUM ги⊸

Рис. 9.16. Закладка Данные окна редактирования регистра Нагрузка

Конструктор движения ре	гистров		□ × □				
열혈철수							
Регистры		Реквизиты докум	ента				
💽 РегистрНакопления.Нагр	узка	💻 Дата					
		Номер					
		Текстрокане Текстрокане	речень. 9 чащиися				
		ТекСтрокаПе	речень Курс				
Табличная часть: Перечень	▼		< <u>Н</u> азад Далее>				
Поле	Выражение		Заполнить выражения				
🛴 Учащийся	ТекСтрокаПеречень.Учащийся						
L→ Kypc	ТекСтрокаПеречень.Курс		Одистить выражения				
🧯 ЧислоЧасов	ТекСтрокаПеречень.ЧислоЧасов						
			ОК				
			Отмена				
			Справка				

Рис. 9.17. Заполнение выражений в окне конструктора движений регистра

(листинг 9.5), которая формирует движения по регистру при проведении документа.

Листинг 9.5. Процедура, выполняемая при проведении документа

```
Процедура ОбработкаПроведения (Отказ, Режим)
Для Каждого ТекСтрокаПеречень Из Перечень Цикл
// регистр Нагрузка
Движение = Движения.Нагрузка.Добавить();
Движение.Период = Дата;
Движение.Учащийся = ТекСтрокаПеречень.Учащийся;
Движение.Курс = ТекСтрокаПеречень.Курс;
Движение.ЧислоЧасов = ТекСтрокаПеречень.ЧислоЧасов;
КонецЦикла;
//}}____КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

На этом этап подготовительных действий можно считать практически завершенным, осталось только перейти в режим 1С:Предприятие, заполнить справочники и создать несколько документов.

Таким образом, мы подготовили необходимую базу для проведения статистического анализа. Создадим несколько отчетов в стиле предыдущих примеров этой главы.

Первый отчет будет касаться получения общей статистики, и для этого на закладке **Модуль** необходимо разместить текст процедуры, приведенной в листинге 9.6.

Листинг 9.6. Процедура для формирования общей статистики (вариант 1)

```
Процедура КнопкаСформироватьНажатие (Кнопка)
Анализ = Новый АнализДанных;
Анализ. ТипАнализа=Тип ("АнализДанныхОбщаяСтатистика");
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Курс.Преподаватель,
| Учащийся, СУММА (ЧислоЧасов) КАК СуммаЧасов
I N3
     РегистрНакопления.Нагрузка КАК Нагрузка
| СГРУППИРОВАТЬ ПО Учащийся, Курс. Преподаватель ";
Анализ.ИсточникДанных= Запрос.Выполнить ();
РезультатАнализа = Анализ.Выполнить ();
Построитель=Новый ПостроительОтчетаАнализаДанных ();
Построитель.Макет=Неопределено;
Построитель. ТипАнализа=Тип ("АнализДанныхОбщаяСтатистика");
ТабДок = Новый ТабличныйДокумент;
Построитель. Вывести (РезультатАнализа, ТабДок);
ТабДок.Показать ();
КонецПроцедуры
```

После размещения процедуры перейдем в режим 1С:Предприятие и сформируем необходимый отчет, фрагмент которого представлен на рис. 9.18.

Учитывая, что используемый регистр является регистром оборотов, процедуру в листинге 9.6 можно заменить более эффективной (листинг 9.7).

Листинг 9.7. Процедура для формирования общей статистики (вариант 2)

```
Процедура КнопкаСформироватьНажатие (Кнопка)
Анализ = Новый АнализДанных;
Анализ.ТипАнализа=Тип ("АнализДанныхОбщаяСтатистика");
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Курс.Преподаватель, Учащийся, ЧислоЧасовОборот
| ИЗ РегистрНакопления.Нагрузка.Обороты";
Анализ.ИсточникДанных= Запрос.Выполнить();
РезультатАнализа = Анализ.Выполнить();
Построитель=Новый ПостроительОтчетаАнализаДанных();
Построитель.Макет=Неопределено;
Построитель.ТипАнализа=Тип ("АнализДанныхОбщаяСтатистика");
ТабДок = Новый ТабличныйДокумент;
Построитель.Вывести (РезультатАнализа, ТабДок);
ТабДок.Показать();
```

КонецПроцедуры

🏪 1C	Пре	цприя	тие	: (y4	ебная версия) - Конфигура	ация									_	
і <u>Ф</u> аи	и <u>п</u> р	авка	iac / B	олица	а Операции <u>с</u> ервис <u>О</u> кна С	л <u>р</u> авка				-	a interior di			- (7.9tm		
	-	- d	5 =		B 😅 🛛 🤉 ୯ M 🗌			•	<u>ж</u> ж	· • • • •	-	<u> </u>	[M+ N	1- X -		
A 1	абли	ща													_ [⊐ ×
12	30		1	2	3				4	5		6		7	8	^
		1	μ													_
		2		0	бщая статистика	3										
		3	1		•											
P		4	1	Ин	формация о данных											
		5			Колицество объектов:				4							
L		7			KONPACCI DO OU BERTOD.				4							
Θ		8	1	He	прерывные поля											
ΠT	в пепрерывные поля									_						
		10	-	<u></u>	Поле	Значен	ий	Минин	имум Максимум		имум	Cpe,	Среднее		мах	<u>C</u> 1
L		11		Сум	мачасов		4		5		30	<u>и</u>	16,00		25	_
A		12	1	Ли	скретные поля											
ΠŤ		14		-												_
Ę)	15		Ку	рсПреподаватель											
		15	-		Колицество значений:				4							
		18	1		Количество уникальных значе	ений:			2							
		19	1		Мода:			Петров	0. B .							
		20														
	曱	21		Та	блица частот											
		22							Относи	ельная	Наког	пленная	-	Накопле	нная	-
		23			Значение		Час	тота	час	гота	ча	стота	отно	осительна	и частот	a
		24		Пет	ров О.В.			3		75,00			3		75,	,00
	L	25		Ани	кина Л.В.	1		1		25,00			4		100,	,00
<		1.28	1			i										>
	uer C	FREUCE			Tafauna											
	HOF L	гапист	DAR		Антарлица											
Для п	олуче	ния по,	дска	ззки	нажмите F1									C	AP NUN	1 //

Рис. 9.18. Отчет с отражением общей статистики

В разработанном отчете дискретными являются поля, определяющие информацию о преподавателе и учащемся. В представленном на рис. 9.18 фрагменте информация по этим полям не видна, однако если в режиме 1С:Предприятие прокрутить окно со сформированным отчетом, то мы все увидим (рис. 9.19).



Рис. 9.19. Диаграмма в отчете с отражением общей статистики

Попробуем теперь представить общую статистику по виду работы преподавателей и числу заказанных часов занятий. Изменим текст программной процедуры (листинг 9.8), что обеспечит нам в режиме 1С:Предприятие соответствующий отчет. Результат в этом случае будет выглядеть так, как показано на рис. 9.20.

Листинг 9.8. Процедура для формирования общей статистики (вариант 3)

```
Процедура КнопкаСформироватьНажатие(Кнопка)
Анализ = Новый АнализДанных;
Анализ.ТипАнализа=Тип("АнализДанныхОбщаяСтатистика");
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Курс.Преподаватель.ВидРаботы, ЧислоЧасовОборот
| ИЗ РегистрНакопления.Нагрузка.Обороты";
Анализ.ИсточникДанных= Запрос.Выполнить();
```

РезультатАнализа = Анализ.Выполнить(); Построитель=Новый ПостроительОтчетаАнализаДанных(); Построитель.Макет=Неопределено; Построитель.ТипАнализа=Тип("АнализДанныхОбщаяСтатистика"); ТабДок = Новый ТабличныйДокумент; Построитель.Вывести(РезультатАнализа, ТабДок); ТабДок.Показать(); КонецПроцедуры



Рис. 9.20. Диаграмма, отражающая статистику по виду работы

Рассмотрим пример, который позволяет выявить учащихся, одновременно записывающихся на курсы (в одном документе). Учитывая, что уже имеющаяся форма вполне подходит и для этой ситуации, нам просто потребуется скорректировать процедуру из листинга 9.8. Ее новый вариант представлен в листинге 9.9. Здесь Регистратор будет объектом анализа, а учащийся — элементом. Готовый отчет приведен на рис. 9.21.

Листинг 9.9. Процедура для формирования поиска ассоциаций

```
Процедура КнопкаСформироватьНажатие(Кнопка)
Анализ = Новый АнализДанных;
Анализ.ТипАнализа=Тип("АнализДанныхПоискАссоциаций");
```

340

Запрос = Новый Запрос; Запрос.Текст = "ВыБРАТЬ Регистратор, Учащийся ИЗ РегистрНакопления.Нагрузка"; Анализ.ИсточникДанных= Запрос.Выполнить(); РезультатАнализа = Анализ.Выполнить(); Построитель=Новый ПостроительОтчетаАнализаДанных(); Построитель.Макет=Неопределено; Построитель.ТипАнализа=Тип("АнализДанныхПоискАссоциаций"); ТабДок = Новый ТабличныйДокумент; Построитель.Вывести(РезультатАнализа, ТабДок); ТабДок.Показать(); КонецПроцедуры

뜸 1С:П	редпр	лият	гие (учебн	ая верс	:ия) - І	Конфигу	рация					>	<
<u>∎</u> айл	<u>П</u> равк	a.	Табли	ица Он	перации	<u>С</u> ерви	ю <u>О</u> кна	Справ	ка					
I 🗋 🚄		Ж	e.	ß	a 🗟	5	e 29	>> •			🥂 м	M+	м- 😤 .	
🛓 Таб	лица												_ 🗆 ×	T
123		1	2				3				4		^	
	28													
P	29		Час	сто в	стреч	аемь	іе груг	пы						
	24	-	No		Konung		qoeinm	Проце	UT C MA	1300				
	21	-	Coci	TOP	ROMAG	CIDO C	лучась	проце	п слу	IGCD				
	32	1	0001	1			1			16.67				
	34	1	V	ນລິມເມເລີດ	і га = Пем	зина А	<u>п</u>			10,01				
	35	1	V.	чашийс	a = Kuni	иппов (<u>) П</u>							
	36	1	<u> </u>	101 000 1710	in inder									
₽	37	1	Acc	соция	ативны	ые пр	авила	<u>ı</u>						I
	30	-						-					F	
	39	-	исхо	одныи	нарор			Сл	едстви	e			ť	
117	40	-	1 	และแม่นั่ง	a – Kum	40000 (,	/	04 - F	los source	<u>а п</u>	+	
LL	41		+	чащини	ла – кири	00008 0	///.		чащии	<u>ca -)</u>	ICI0I3PING	<u>(A)</u> [.		
	42	-												
	43	1												
	45	1											~	
<	40												>	
Щ у Отчет	ринина Статистика 💦 Таблица													
Для полу	чения	под	сказн	ки наж	мите F1							CAP	NUM	//

Рис. 9.21. Отчет с отражением ассоциативной статистики

Статистические отчеты к главе 2

В этом разделе мы вернемся к информационной базе, разработанной в *главе* 2, и добавим к имеющимся объектам конфигурации несколько отчетов со статистическими показателями.

Начнем изменения с создания объекта конфигурации — отчета под именем отчетПоОбщейСтатистике. В нем нам понадобится простая форма с кнопкой для формирования отчета. Технические действия сводятся к тому, что на закладке Модуль мы будем размещать необходимые процедуры. Начнем с формирования отчета по статистике поступлений товаров. Для формирования запроса мы воспользуемся табличной частью документа заказ. Необходимый текст программной процедуры представлен в листинге 9.10.

Листинг 9.10. Процедура для формирования статистики по поступлениям

```
Процедура КнопкаСформироватьНажатие (Кнопка)
Анализ = Новый АнализДанных;
Анализ.ТипАнализа=Тип("АнализДанныхОбщаяСтатистика");
Запрос = Новый Запрос;
Запрос.Текст = "ВЫЕРАТЬ Товар, Количество ИЗ Документ.Заказ.Перечень";
Анализ.ИсточникДанных= Запрос.Выполнить();
РезультатАнализа = Анализ.Выполнить();
Построитель=Новый ПостроительОтчетаАнализаДанных();
Построитель.Макет=Неопределено;
Построитель.ТипАнализа=Тип("АнализДанныхОбщаяСтатистика");
ТабДок = Новый ТабличныйДокумент;
Построитель.Вывести(РезультатАнализа, ТабДок);
ТабДок.Показать();
КонецПроцедуры
```

Фрагмент отчета, включающего полученную статистику, представлен на рис. 9.22.



Рис. 9.22. Отчет с отражением статистики по заказанным товарам

10	10-0	DOGRD										
. V8	16.11	редпр	иян	ие (учеоная версия) - конфигура	щия						-	
≣.	<u> Ф</u> айл	<u>П</u> равк	a T	аблица Операции <u>С</u> ервис <u>О</u> кна С	правка							
	1		Ж	ЪС 🛋 🙆 🖢 С н		• 8	" 🙀 1	5 🖻 🗸	1	M M+ M-	- 🎇 🚽	
	🔓 Tat	блица										
	123		1	2 3		4		5	6	7		8 [
		34		Товар = Мякая мебель								
		35		Товар = Стенка (мод. 15-7)								
	LL	36		Товар = Кухня Уют-З			L					
		37										
	Ę	38		<u>Ассоциативные правила</u>								
		40		Исхолный набор	Спелствие			Процент с та	ев Пол	стоверность	Значимо	сть
		40		1	слодотоло			80	1 00	100.00	ond mino	1 25
	I۲	42		Товар = Мяжая мебель	Tosan = Ky	XH9 VIDT-3			,00	100,00		1,20
		43		Товар = Стенка (мод. 15-7)								
	占	44		2				80	0.00	100.00		1.25
	IΤ	45		- Товар = Мягкая мебель	Товар = Ст	генка (мод. 15-	7)					
		46			Товар = Ку	/хня Уют-З						
	۱Å.	47		3		•		80	0.00	100,00		1,25
	ΙT	48		Товар = Күхня Уют-З	Товар = М	ягкая мебель						
		49			Товар = Ст	генка (мод. 15-	7)					
	1.5		i +						!			
	/ Отчет			общей стат 🚡 Таблица								
Д	ля полі	учения	пода	жазки нажмите F1							CAP NU	м //

Рис. 9.23. Отчет с отражением вместе заказываемых товаров

Теперь создадим отчет для поиска ассоциаций при заказах товаров. Нас интересуют товары, присутствующие вместе в документе заказ. В листинге 9.11 представлена процедура формирования данной статистики, а на рис. 9.23 приводится отчет, формируемый в режиме 1С:Предприятие.

Листинг 9.11. Процедура для поиска ассоциаций при заказах товаров

```
Процедура КнопкаСформироватьНажатие (Кнопка)
Анализ = Новый АнализДанных;
Анализ.ТипАнализа=Тип ("АнализДанныхПоискАссоциаций");
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ Ссылка, Товар ИЗ Документ.Заказ.Перечень";
Анализ.ИсточникДанных= Запрос.Выполнить();
РезультатАнализа = Анализ.Выполнить();
Построитель=Новый ПостроительОтчетаАнализаДанных();
Построитель.Макет=Неопределено;
Построитель.ТипАнализа=Тип ("АнализДанныхПоискАссоциаций");
ТабДок = Новый ТабличныйДокумент;
Построитель.Вывести (РезультатАнализа, ТабДок);
ТабДок.Показать();
КонецПроцедуры
```

На рассмотренных примерах мы выяснили, что анализ данных позволяет увидеть явно не бросающиеся в глаза закономерности в имеющихся данных и представить их в виде легко воспринимаемого результата. При большом объеме информации найти неочевидные закономерности достаточно сложно. И в этом случае средства автоматизации позволяют легко помочь пользователю. Полученный результат анализа можно эффективно использовать при принятии решений управленческим составом организации. глава 10



Знакомимся с версией 1С:Предприятие 8.2

В предыдущих девяти главах книги мы рассмотрели многообразие технических ресурсов, которые заложены в платформу 1С:Предприятие 8.1. Фактически рассмотренная программная платформа позволяет реализовать решение любой прикладной задачи. Однако прогресс не стоит на месте и в 2009 году фирма "1С" выпустила очередную версию своей знаменитой системы — 1С:Предприятие 8.2. Несмотря на новизну этой версии, она уже активно используется в ряде организаций. Можно сказать, период практического использования новой версии программы 1С:Предприятие 8.2 начался, в ближайшие годы она должна занять лидирующее место при решении задач автоматизации различных участков работы в организациях.

Важно подчеркнуть, что 1С:Предприятие 8.2 значительно отличается от версий 8.0 и 8.1. Даже переход от версии 8.0 к 8.1 не был таким качественным. В новую версию заложено большое количество новых подходов к разработке и новинок в пользовательском интерфейсе.

К настоящему моменту существует огромное количество прикладных решений на платформе 1С:Предприятие 8.1. Понятно, что для разработчиков и пользователей переход на версию 8.2 должен быть легкий и максимально "безболезненный". Именно так он и реализован в новой версии программного продукта. Для выполнения разработок с использованием новых ресурсов, а также для совместимости с предыдущими разработками (на платформе 1С:Предприятие 8.1) в версии 8.2 можно использовать два режима:

- старый (совместимый с версией 8.1);
- новый режим, который называется режимом управляемого приложения.

Таким образом, если мы располагаем информационной базой, созданной на версии 8.1, то ее следует сконвертировать. Такая конвертация производится системой автоматически (разумеется, при этом программа спросит у пользователя согласия на выполнение подобной процедуры). После выполнения конвертации вы сможете работать со своей разработкой уже на новой платформе, и никакая доработка в этом случае не потребуется. После запуска вашего прикладного решения уже в новой версии 8.2 внешний вид приложения для пользователя будет таким же, как прежде в знакомой программе 1С:Предприятие 8.1. Наличие режима совместимости с версией 8.1 очень важно и позволяет без каких-либо препятствий постепенно перейти на полноценную работу в новой версии.

В режиме управляемого приложения используется управляемый интерфейс и управляемые формы. В этом случае разработчикам и пользователям доступны все новые визуальные и функциональные возможности системы. Если вы сконвертировали свою разработку (из версии 8.1), то после преобразования разработчику предоставляется возможность использования и новых возможностей программы. Удобство такого подхода заключается в постепенном переходе к новому интерфейсу, который предоставляет пользователям комфортную работу в стиле современных программных продуктов.

Вид нового интерфейса

Наверняка, первое, что хочется спросить, когда речь идет о новой версии — как выглядит окно программы для пользователя (в режиме 1С:Предприятие). На рис. 10.1 показан вид прикладного решения для задачи, рассмотренной в *главе 1*. Напомним, что там речь шла об учете оплаты учащихся за учебные курсы. Видно, что по сравнению с переходом от версии 8.0 к версии 8.1 здесь изменения весьма значительные. Совершенно новый интерфейс приложения невольно провоцирует вопрос — это действительно система 1С:Предприятие? В верхней части на рис. 10.1 расположена панель разделов, которая для рассматриваемого прикладного решения включает следующие разделы:

- Рабочий стол (непременный атрибут любой разработки);
- Директор;
- Менеджер по оплатам;
- Управление.

	- 🏠 Ko	(1С:Предприятие)	☆ 🖻 🔳 🖼	M M+ M- (i	×
Рабочий стол	 Директор	Менеджер по оплатам	— Управление		
Должности Клиенты Курсы		Клиенты	5 🖉 🗵 Q,H	айти Все де	йствия 🕶 ③
Специалисты		Наименование	🚊 Код		
		– Николаев (C. Ф. 000000002		+
		😑 Сидоров Е.	П. 000000001		
		📟 Торин Д.В.	00000003		-
		4	<u>n</u>		
История	ę.				-5

Рис. 10.1. Интерфейс системы 1С:Предприятие 8.2

Кроме рабочего стола, остальные вышеперечисленные разделы являются специфическими для решаемой прикладной задачи. В данном случае три раздела являются следствием разделения использования разработанных объектов конфигурации для трех потенциальных пользователей разработки. При необходимости можно создать столько разделов, сколько требуется.

Примечание

Рабочий стол представляет собой стандартный раздел приложения, содержащий часто используемые объекты (это могут быть справочники, документы или отчеты).

Содержание панели разделов, которую мы видим на рис. 10.1, является следствием формирования трех подсистем в режиме конфигуратора. Подсистемы позволяют выделить в прикладном решении специфические функциональные ресурсы для различных категорий пользователей. Так, для директора в работе требуются определенные отчеты, а бухгалтер работает со своими документами. В связи с этим объект конфигурации **Подсистема** занимает в платформе 1С:Предприятие 8.2 ключевое место.

На рис. 10.2 (в левой части) показано содержимое окна Конфигурация, в котором отражены три подсистемы (три объекта конфигурации, которые мы создали для представления прикладного решения пользователям). Для создания новой подсистемы, аналогично созданию встречавшихся ранее в книге объектов конфигурации, достаточно щелкнуть правой кнопкой мыши на названии Подсистемы и в открывшемся контекстном меню выбрать пункт Добавить. Это действие приводит



Рис. 10.2. Окно конфигуратора системы 1С:Предприятие 8.2

к отображению на экране окна редактирования объекта конфигурации (рис. 10.2) и последующие действия разработчика касаются выполнения ряда несложных визуальных настроек.

Наряду с созданием подсистем основные действия, как и в прежней версии, заключаются в создании необходимых справочников, документов, отчетов и т. д. При этом каждый создаваемый справочник, документ или отчет следует отнести к одной либо нескольким подсистемам. Если обратиться к рис. 10.1, то на нем мы видим интерфейс подсистемы **Управление** в режиме 1С:Предприятие. В эту подсистему включены знакомые по *главе 1* справочники:

- 🔶 Должности;
- ♦ Клиенты;
- 🔶 Курсы;
- ♦ Специалисты.

В центральной части рис. 10.1 располагается одна из форм справочников форма списка справочника клиенты. Заметим, что в разработках на платформе 1С:Предприятие 8.1, описанных в предыдущих главах, мы использовали свойство Способ редактирования, с помощью которого можно было определить форму для редактирования (форму списка или форму элемента). В версии 8.2 не поддерживается такой режим работы. Редактировать данные в форме списка теперь нельзя при редактировании автоматически открывается форма элемента.

В левой части на рис. 10.1 отображен важный элемент нового интерфейса — панель навигации. На данной панели отражены объекты, которыми располагает пользователь при выбранном интерфейсе. Фактически содержание навигации представляет собой совокупность команд (команды в данном контексте — это действия, которые может выполнять пользователь). При использовании на панели навигации команд открытия списков автоматически отображаются соответствующие управляемые формы списков в рабочей области окна приложения. На рис. 10.1 рабочую область окна приложения занимает форма списка справочника Клиенты.

Заметим, что картина, которую мы видим на рис. 10.1, является следствием того, что щелчком мыши мы выбрали интерфейс Управление. Как результат этого дей-



Рис. 10.3. Интерфейс подсистемы Менеджер по оплатам

ствия на панели навигации нашли место справочники, которые мы включили в данную подсистему при их разработке.

На рис. 10.3 приведен вид интерфейса **Менеджер по оплатам**. Здесь в центральной части окна (рабочей области) мы видим панель действий. Команды, размещенные на панели действий, позволяют быстро создать элемент справочника, добавить новый документ, сформировать отчет. Это ускоряет работу пользователя, сокращая технические действия до минимума.

Информационная панель

Рассмотрим окно режима 1С:Предприятие на рис. 10.4, где в нижней части окна приложения присутствует строка с информацией о фамилиях, а также о названии одного из курсов. Эта строка представляет собой еще один элемент интерфейса — информационную панель. Основное назначение информационной панели заключается в формировании сведений о последних действиях, выполненных в системе. Если выполненное действие было связано с записью данных, то, щелкнув по гиперссылке оповещения, мы откроем форму, содержащую данные. Применительно к рис. 10.4 с помощью первой (слева) гиперссылки оповещения производится открытие формы элемента справочника клиенты с внесенной записью о Николаеве С.Ф.

Примечание

Последние оповещения об изменении объектов информационной базы отображаются в правой части информационной панели для быстрого доступа к ним.



Рис. 10.4. Содержание информационной панели с отражением последних действий

В левой части информационной панели присутствует важная кнопка **История**, с помощью которой можно открыть историю работы пользователя с программой (рис. 10.5). Это является особенностью новой версии системы. Важно отметить, что такое запоминание не просто пассивное — система позволяет выполнить возврат

к событиям из окна **История**. Например, щелчок по гиперссылке **Внесение оплаты** от 24.11.2010 8:41:52 приводит к открытию формы соответствующего документа. Таким образом, всегда можно вернуться и проверить внесенные данные.

Если один конкретный объект информационной базы изменялся несколько раз, то в истории будет храниться единственная запись о последнем изменении. Важно отметить, что окно **История** хранит не более 200 записей, и поэтому при внесении новых данных старые удаляются.

Открыть	
Чаименование	Дата
<u> Тетров А.П.</u>	24.11.2010 11:23:09
Базы данных	24.11.2010 11:22:49
Николаев С.Ф.	24.11.2010 11:22:41
Внесение оплаты 000000001 от 24.11.2010 8:41:52	24.11.2010 8:41:52
MS Excel	24.11.2010 8:41:39
Сомпьютерная графика	24.11.2010 8:40:39
Горин Д.В.	24.11.2010 8:14:09
Сидоров Е.П.	24.11.2010 8:13:34

Рис. 10.5. История работы пользователя с программой

Примечание

В окно **История** попадают только действия, которые пользователь выполнил интерактивно. В случае если объекты системы были изменены программно, то подобные события в истории не фиксируются.

Системные команды и главное меню

Обратимся теперь к заголовку окна приложения (см. рис. 10.4), где присутствует область системных команд. В левой части данной области располагается кнопка для перехода к главному меню. Правее находятся команды для перемещения по истории работы в основном окне (кнопки **Вперед** и **Наза**д) и меню **Избранное**.

В главное меню входят типовые команды, характерные для приложения в целом. Эти команды не зависят от конкретной конфигурации.

Команды Вперед и Назад позволяют легко перемещаться по истории работы в текущем пользовательском сеансе. Если необходимо, то в выпадающем списке (сам список открывается с помощью щелчка мышью по пиктограмме с изображением стрелки) можно наблюдать пункты программы, которые открывал пользователь (рис. 10.6). Далее правее в области заголовка располагается пиктограмма, обеспечивающая переход к избранным ссылкам пользователя (в состав избранных ссылок можно добавить необходимый отчет или текущую точку навигации программы).

В правой части заголовка окна приложения располагаются пиктограммы, соответствующие вспомогательным командам:

- переход по ссылке;
- калькулятор;
- календарь;
- информация о системе.

Рабочий стол Должности Клиенты Килен	 Управление - Курсы Управление - Специалисты Управление - Курсы Управление Менеджер по оплатам - Клиенты Менеджер по оплатам Рабочий стол 		ние ,Найти 🍋 Все действия + 🥝		
Специалисты	-	Наименование	A	Код	A
	r 7	- MS Excel		00000003	r.
		📟 Базы данных	000000001		
		📟 Компьютерная графика	000000002		
					-

Рис. 10.6. Список с разделами, использованными пользователем

Особенности режима 1С:Предприятие

Одной из особенностей версии 8.2 является отсутствие меню **Операции** при работе пользователя в режиме 1С:Предприятие. Этим разделом меню программы 1С:Предприятие 8.1 мы пользовались на протяжении всех глав книги. В этом случае в версии 8.1 мы сразу получали доступ к объектам, созданным системой на основании разработанных объектов конфигурации. Такая организация была весьма удобна. Однако не очень хорошо, если пользователю невысокой квалификации легко доступен подобный ресурс. В этом случае, открыв меню **Операции**, он получал доступ даже к тем объектам, которые ему не предназначены.

По логике команда, аналогичная команде **Операции**, должна быть доступна только ограниченной категории пользователей. В новой версии подобная команда автоматически не отображается в меню. Для включения доступа ко всем объектам необходимо в главном меню выбрать раздел **Сервис**, в котором следует щелкнуть мышью на пункте **Параметры**. После этого в открывшемся окне (рис. 10.7) следует установить флажок **Отображать команда Все функции**». Как результат этой установки в главном меню появится команда **Все функции** (рис. 10.8). Если ею воспользоваться, то мы увидим окно (рис. 10.9) с перечнем объектов.

В меню Операции предыдущей версии программы существовал набор системных команд типа Удаление помеченных объектов, Проведение документов

Параметры × 🖓 Отладка в текушем сеансе разрешена Устанавливать режим разрешения отладки при запуске Отображать показатели производительности Имитировать задержку при вызовах сервера: 1,45 💲 Задержка при вызове (с.): Задержка при передаче данных (с./Кбайт): 0.45 2 0.15 单 Задержка при получении данных (с./Кбайт): 🕑 Отображать команду «Все функции» 0K Справка Отмена

и т. д. Сейчас эти команды собраны в уже знакомом окне Все функции (ветка Стандартные).

Рис. 10.7. Окно для установки параметров системы

д К	Файл Правка Сервис Окна Справка Все функции	• • •	Менеджер по оплатам Курсы Ф Создать С Ф Х О Найти Э Ви	се действия •	(7)
Курсы Специа	листы		Наименование	Код	
		5	- MS Excel	00000003	
			😑 Базы данных	000000001	
			😑 Компьютерная графика	000000002	
					V

Рис. 10.8. Главное меню приложения

При рассмотрении примеров в ряде глав книги активно использовалось свойство Способ редактирования. В этом случае мы могли установить для режима 1С:Предприятие тот или иной вариант редактирования справочников. Так, можно остановиться на редактировании в диалоге или в списке (кроме того, имеется возможность сохранения выбора за пользователем в режиме 1С:Предприятие, где он может самостоятельно переключать формы). Теперь свойства Способ редактирования в платформе 8.2 нет, что связано с тем, что не поддерживается режим редактирования ссылочных данных в списке. Данные в этом случае всегда редактируются в отдельной форме.





Режим разработки

В версии 8.2 проектирование прикладного решения начинается с проектирования состава подсистем. Разбивка на подсистемы ориентирована на пользователей. При таком подходе прикладное решение становится более понятным. Для создания новой подсистемы (либо редактирования существующей) следует воспользоваться окном редактирования объекта конфигурации, где на закладке **Состав** определяется набор объектов конфигурации, входящих в данную подсистему (рис. 10.10).

В верхнем списке на рис. 10.10 представлены все имеющиеся объекты конфигурации, а в нижнем только те, которые входят в состав подсистемы. Для редактирования состава подсистем в версии 8.2 появился новый редактор под названием **Все подсистемы**. Вызвать данный редактор можно с помощью контекстного меню, которое открывается щелчком мыши по разделу **Подсистемы** (рис. 10.10). Главное назначение данного нового редактора заключается в формировании состава командного интерфейса. Также этот новый редактор позволяет изменять состав объектов, входящих в подсистемы.

Для работы по организации интерфейса подсистем предусмотрен командный интерфейс, окно которого открывается с помощью контекстного меню, вызываемо-

го щелчком правой кнопкой мыши в окне Конфигурация по пункту с названием интересующей подсистемы (рис. 10.11). Его основное назначение заключается в редактировании видимости команд.



Рис. 10.10. Окно редактирования подсистемы



Рис. 10.11. Командный интерфейс

В версии 8.2 появился новый вид форм — управляемые формы. При этом поддерживается механизм работы и со старыми формами (они нам знакомы по версии 1С:Предпиятие 8.1). В дереве объектов конфигурации формы не отличаются друг от друга. Однако при открытии формы разница сразу заметна. Для работы с новыми формами в версии 8.2 добавлен новый редактор, который имеет структуру, отличающую от знакомого по главам книги редактора форм.

В платформе 8.2 теперь нет необходимости рисовать форму вручную. Разработчик только описывает состав формы (управляемой), а система сама располагает нужные элементы на форме.

Идеология использования форм такова — при создании новых решений на платформе 8.2 следует придерживаться использования новых форм. Однако если функциональности новых оказывается недостаточно, то можно использовать старые (неуправляемые) формы.

Одна из основных идей платформы 8.2 заключается в удобстве разработки прикладных решений и максимальной автоматизации при создании интерфейса.

Пример программирования формы документа

Для того чтобы представить элементы сходства и отличия разработки прикладных решений для платформ 8.1 и 8.2, рассмотрим в этом разделе пример, в котором будет использовано программирование формы документа в системе 1С:Предприятие 8.2. Аналогичные примеры для платформы 8.1 нам встречались на протяжении всей книги.

Будем считать, что в конфигурации мы располагаем справочниками организации (включает список организаций, с которыми наша фирма сотрудничает) и товары (содержит перечень товаров). Так, указанные организации выступают в качестве наших поставщиков, и у каждой организации для нас предусмотрена скидка, которая со временем меняется. Для подобной информации следует создать периодический регистр сведений Скидкиоторганизаций. В качестве значения параметра Периодичность установим вариант Один день.

После этого на закладке Данные окна редактирования объекта конфигурации СкидкиОтОрганизаций зададим измерение Организация (тип данных — Справочник-Ссылка.Организации). В качестве ресурса регистра сведений укажем числовой параметр Скидка (будет содержать значение скидки для каждой фирмы в процентах).

Кроме этого, нам потребуется создать документ ПоступленияОтОрганизаций. В окне редактирования объекта конфигурации ПоступленияОтОрганизаций на закладке Данные необходимо определить структуру документа. В данном случае в документе создадим табличную часть Перечень, в которой поместим пять реквизитов:

- ♦ Товар (ТИП ДАННЫХ СправочникСсылка. Товары);
- Поставщик (ТИП Данных СправочникСсылка.Организации);
- Сумма (тип данных число);

- Скидка (тип данных число);
- СуммаСоСкидкой (тип данных число).

В данном документе указывается, какой товар на какую сумму с учетом скидки поступил от конкретного поставщика. Дальнейшая задача заключается в том, чтобы добавить следующий "функционал" в документ ПоступленияОтОрганизаций:

- при указании поставщика (выборе из справочника) в поле поставщик автоматически в поле Скидка должна быть внесена скидка для данной организации из периодического регистра сведений СкидкиОтОрганизаций;
- после внесения суммы поставки товара должна автоматически рассчитываться сумма поставки по данному товару с учетом скидки (заполняться поле суммаСо-Скидкой).

Как и в рассмотренных ранее примерах для платформы 1С:Предприятие 8.1, нам потребуется создать пользовательскую форму документа поступленияОтОрганизаций. Для этого следует в окне редактирования объекта конфигурации ПоступленияОтОрганизаций воспользоваться конструктором формы. Результат работы конструктора (рис. 10.12) приводит к генерации формы, несколько отличающейся от тех, что мы видели в книге при работе с платформой 1С:Предприятие 8.1.



Рис. 10.12. Управляемая форма документа в режиме конфигуратора

Теперь в версии 8.2 при разработке формы разработчику недоступна возможность визуальной дорисовки, и он только указывает, какие элементы необходимо разместить на форме. Далее непосредственное размещение элементов производит сама система, исходя из заложенных в нее критериев.

Если обратиться к рис. 10.12, то видно, что сами элементы формы располагаются в левой верхней части окна. При этом если необходимо включить новый элемент, то следует воспользоваться кнопкой **Добавить**.

Как мы определили ранее, перед нами стоят две задачи расширения "функционала" формы документа. Первая из них заключается в написании программы для обработки события ПриИзменении для поля Поставщик (точнее ПереченьПоставщик, как дополнил конструктор формы). А именно, как только в режиме 1С:Предприятие пользователь выберет конкретного поставщика, то сразу же после этого должна быть внесена соответствующая скидка из регистра сведений СкидкиОтОрганизаций. Для перехода к окну написания программного кода необходимо сначала обратиться к окну свойств поля ПереченьПоставщик, а затем в разделе События с помощью щелчка мыши по изображению лупы напротив события ПриИзменении перейти к написанию программной процедуры. Здесь в качестве обработчика рассматриваемого события напишем текст, представленный на листинге 10.1.

Листинг 10.1. Обработка события ПриИзменении для поля ПереченьПоставщик

&НаКлиенте

Процедура ПереченьПоставщикПриИзменении (Элемент)

Стр=Элементы.Перечень.ТекущиеДанные;

Стр.Скидка=ОбщиеФункции.ПолучитьСкидку (Объект.Дата,Стр.Поставщик); КонецПроцедуры

Здесь сначала с помощью программного объекта мы получаем доступ к данным текущей заполняемой строки табличной части документа, а затем используем пользовательскую функцию получить Скидку(), в которую передаем дату документа и фирму-поставщика. Эту функцию далее мы создадим, а ее "функционал" заключается в извлечении информации о скидке из периодического регистра сведений Скидки Оторганизаций.

Заметим, что функция получить Скидку () может потребоваться не только для нашего документа. В этом случае логично ее поместить в общий раздел, который в дереве объектов конфигурации называется **Общие модули** (подраздел раздела **Общие**). Здесь нам потребуется сначала создать общий модуль, который назовем общиефункции. В окне свойств данного модуля следует обратить внимание на установки, показанные на рис. 10.13.

Далее в созданный модуль общиефункции поместим функцию получитьСкидку (), представленную в листинге 10.2. Теперь если сохранить конфигурацию и перейти в режим 1С:Предприятие, то мы увидим, что при заполнении поля поставщик в документе поступленияОтОрганизаций происходит автоматическая подстановка актуальной скидки. Таким образом, первая задача в плане автоматизации документа решена.

Свойства: Об	щиеФункции Х	
월 🛯 🕽	<	
•Основные:		
Имя	ОбщиеФункции	
Синоним	Общие функции	
Комментарий		
Модуль <u>О</u>	ткрыть	
Глобальный 🗌		
Клиент (управл	пяемо:	Рис. 10.13. Окно свойств общего молупя Общие Функции
Сервер		
Внешнее соед	инениє	
Вызов сервер-	a 🔽	
Привилегиров	анный 🗌	
Повторное исг	тользо Не использовать 🗾 💌	
Разрешает вызо	ов сервера	
📩 Дополните	ельно: 😤 Свойства: Общи	

Листинг 10.2. Функция для извлечения информации о скидке от поставщика

```
Функция ПолучитьСкидку (ДатаРасчета, ЭлементПоставщики) Экспорт
Отбор=Новый Структура ("Организация", ЭлементПоставщики);
Инфо=РегистрыСведений.СкидкиОтОрганизаций.ПолучитьПоследнее (ДатаРасчета,
Отбор);
Возврат Инфо.Скидка;
КонецФункции
```

Перейдем теперь к автоматизации вычисления суммы со скидкой после ручного заполнения поля Сумма. Для этого в окне редактирования формы документа ПоступленияОтОрганизаций необходимо написать обработчик события ПриИзменении для поля переченьСумма. Для перехода к окну написания программного кода нужно сначала обратиться к окну свойств поля переченьСумма, а затем щелчком мыши по изображению лупы напротив события ПриИзменении перейти к написанию программы. Текст, который необходимо в данном случае разместить, представлен в листинге 10.3.

Листинг 10.3. Процедура для автоматического вычисления суммы со скидкой

&НаКлиенте

Процедура ПереченьСуммаПриИзменении (Элемент)

Стр=Элементы.Перечень.ТекущиеДанные;

```
Стр.СуммаСоСкидкой= Стр.Сумма-Стр.Сумма*Стр.Скидка/100;
```

КонецПроцедуры

В результате проведенных действий получен необходимый сервис для работы пользователя с документом ПоступленияОтОрганизаций. Так, при внесении постав-

щика автоматически производится извлечение из регистра сведений СкидкиОтОрганизаций актуальной скидки от поставщика, а также при заполнении суммы производится расчет суммы с учетом скидки.

Большинство программных конструкций, встретившихся нам в листингах 10.1 и 10.2, не вызывает вопросов. Однако первая строка листинга 10.1 «наклиенте требует определенного комментария.

Дело в том, что система 1С:Предприятие 8.2 включает два программных компонента:

• Клиентское приложение;

• Сервер 1С:Предприятие.

Клиентское приложение представляет собой программу, которая обеспечивает интерактивное взаимодействие системы с пользователем. Сервер 1С:Предприятие — это один из рабочих процессов, функционирующих в составе кластера серверов 1С:Предприятие 8. Он обеспечивает взаимодействие Клиентского приложения с базой данных. Также Сервер 1С:Предприятие может исполнять код на встроенном языке.

Важный момент в плане программирования заключается в том, что на сервере и на клиенте доступны разные свойства, методы и объекты встроенного языка.

Другой важный момент касается структуры формы (в нашем примере это была форма документа). В платформе 1С:Предприятие 8.2 форма состоит из двух взаимодействующих между собой частей. Одна часть выполняется в контексте сервера, а другая — в контексте клиента. Для этого все процедуры и функции формы должны включать указание контекста, в котором они будут выполняться. И для такого указания используются директивы компиляции, с одной из которых мы встретились — «Наклиенте. Данная директива указывает на то, что этот программный компонент выполняется в контексте клиентской части приложения. В свою очередь директива «НаСервере говорит о том, что выполнение программного компонента должно производиться в контексте сервера.

Варианты клиентских приложений

В версии 8.1 существовал один вариант платформы — мы устанавливали программу 1сv8.exe с одного дистрибутива, который включал все компоненты. Данная программа обеспечивала интерактивное взаимодействие пользователя с компьютером.

Версия 8.2 включает в себя три различных приложения:

- ◆ "толстый" клиент 1сv8.ехе;
- ♦ "тонкий" клиент 1сv8с.ехе;
- Web-клиент.

"Толстый" клиент представляет собой программу, в определенной степени идентичную той, с которой мы встретились в версии 8.1. Отличие заключается в том, что "толстый" клиент может работать в режиме управляемого приложения.

В этом случае разработчику и пользователю доступны управляемый интерфейс и управляемые формы. Несколько вычурное название объясняется его функциональностью. Это приложение реализует всю функциональность встроенного языка.

"Тонкий" клиент представляет собой новое клиентское приложение. Оно работает с информационной базой только в новом режиме управляемого приложения (управляемый интерфейс и управляемые формы).

"Тонкий" клиент требует существенно меньших аппаратных ресурсов. Это приложение может общаться по локальной сети и через Интернет. Название "тонкий" объясняется тем, что данное приложение умеет исполнять ограниченный набор функциональности встроенного языка.

Web-клиент выполняется в среде интернет-браузера. И для работы с Webклиентом необходимо лишь открыть браузер на своем компьютере и указать адрес Web-сервера, на котором выставлена информационная база. В этом случае Webклиент автоматически начнет выполняться.
Заключение

Итак, последняя глава книги завершена. Главный результат заключается в том, что вы на примерах различной степени сложности познакомились с технологией разработки конфигураций для информационной системы 1С:Предприятие 8.1. В ходе рассмотрения примеров вы освоили практические приемы работы с объектами конфигурации. Теперь перед вами открываются широкие возможности для внесения изменений в готовые решения, а также для создания собственных разработок.

Хотелось бы добавить несколько слов для тех читателей, которые не располагают программой 1С:Предприятие 8.1 (например, организация, где они работают, использует до сих пор весьма популярную предыдущую версию программного продукта — 1С:Предприятие 7.7). В этом случае можно воспользоваться очень недорогой версией для обучения. Замечу, что все разработки в данной книге выполнены с помощью этого программного продукта.

Хочу выразить большую благодарность издательству "БХВ-Петербург", которое приняло решение о выпуске данного (четвертого) издания моей книги. Скоро в издательстве "БХВ-Петербург" планируется выпуск еще одной моей книги, в которой будут рассматриваться приемы разработки прикладных решений на платформе 1С:Предприятие 8.2.

ПРИЛОЖЕНИЕ

Описание компакт-диска

На прилагаемом к книге компакт-диске вы найдете информационные базы, разработанные в книге, а также файлы приложений Microsoft Word и Microsoft Excel, которые упоминаются в ряде глав.

В табл. П1 приводятся пояснения относительно содержания имеющихся на компакт-диске папок.

Название папки	Описание
Glava_1	Информационная база, разработанная на основании примеров, описанных в <i>главе 1</i>
Glava_2	Информационная база, разработанная на основании примеров, описанных в <i>главе 2</i>
Glava_3	Информационная база, разработанная на основании примеров, описанных в <i>главе</i> 3
Glava_4	Информационная база, разработанная на основании примеров, описанных в <i>главе 4</i>
Glava_5	Информационная база, разработанная на основании примеров, описанных в <i>главе 5</i>
Glava_6 (part1)	Информационная база, разработанная на основании примеров, описанных в первой части <i>главы 6</i>
Glava_6 (part2)	Информационная база, разработанная на основании примеров, описанных во второй части <i>главы 6</i>
Glava_7 (part1)	Информационная база, разработанная на основании примеров, описанных в первой части <i>главы</i> 7
Glava_7 (part2)	Информационная база, разработанная на основании примеров, описанных во второй части <i>главы</i> 7
Glava_7 (part3)	Информационная база, разработанная на основании примеров, описанных в третьей части <i>главы</i> 7
Glava_8 (part1)	Информационная база, разработанная на основании примеров, описанных в первой части <i>главы</i> 8
Glava_8 (part2)	Информационная база, разработанная на основании примеров, описанных во второй части <i>главы 8</i>

Таблица П1. Папки и файлы сопроводительного компакт-диска

Название папки	Описание
Glava_8 (part3)	Информационная база, разработанная на основании примеров, описанных в третьей части <i>главы</i> 8
Glava_9 (part1)	Информационная база, разработанная на основании примеров, описанных в первой части <i>главы 9</i>
Glava_9 (part2)	Информационная база, разработанная на основании примеров, описанных во второй части <i>главы</i> 9
Glava_9 (part3)	Информационная база, разработанная на основании примеров, описанных в третьей части <i>главы</i> 9
Arhiv	Архивы информационных баз
Add_Glava_5	Файлы к рассмотренному в <i>главе 5</i> примеру автоматического заполнения документа, созданного в приложении Microsoft Word
Add_Glava_8	Файлы к <i>главе</i> 8, созданные в приложениях Microsoft Word и Microsoft Excel

Для установки информационных баз необходимо:

- 1. Скопировать папку, содержащую интересующую вас информационную базу (см. табл. П1), с компакт-диска на жесткий диск. Скажем, для примеров главы 1 полный путь местонахождения информационной базы может быть следующим C:\Glava_1.
- 2. В окне запуска информационной системы 1С:Предприятие 8.1 нажать кнопку Добавить.
- 3. В открывшемся окне выбрать вариант Добавление в список существующей информационной базы.
- 4. Подобрать любое название создаваемой информационной базе. Это может быть, скажем, "Информационная база с примерами к 1-й главе".
- 5. Задать каталог информационной базы. В качестве этого каталога надо указать папку, которую вы перенесли с компакт-диска на жесткий диск вашего компьютера.

На этом установка завершена, и в окне запуска можно перейти либо в режим конфигуратора, либо в режим 1С:Предприятие.

Литература

- 1. Белоусов П. С., Островерх А. В. 1С:Предприятие от 8.0 до 8.1. М.: 1С-Паблишинг, 2008. — 288 с.: ил.
- 2. Габец А. П., Гончаров Д. И. 1С:Предприятие 8.0. Простые примеры разработки. — М.: 1С-Паблишинг, 2005. — 432 с.: ил.
- 3. Кашаев С. М. Программирование в Microsoft Excel на примерах. СПб.: БХВ-Петербург, 2007. — 320 с.: ил. + Видеокурс (на CD-ROM).
- 4. Кашаев С. М. 1С:Предприятие 8. Учимся программировать на примерах. СПб.: БХВ-Петербург, 2008. 336 с.: ил. + CD-ROM.
- 5. Кашаев С. М. Офисные решения с использованием Microsoft Excel 2007 и VBA. СПб.: Питер, 2008. 320 с.: ил. + CD-ROM.
- 6. Кашаев С. М. Офисные решения с использованием Microsoft Excel 2007 и VBA. СПб.: Питер, 2008. 320 с.: ил. + CD-ROM.
- 7. Михайлов А. В. 1С:Предприятие 7.7/8.0: системное программирование. СПб.: БХВ-Петербург, 2005. 336 с.: ил.
- 8. Радченко М. Г. 1С:Предприятие 8.0. Практическое пособие разработчика. Примеры и типовые приемы. — М.: 1С-Паблишинг, 2006. — 656 с.: ил.
- Радченко М. Г. 1С:Предприятие 8.1. Практическое пособие разработчика. Примеры и типовые приемы. — М.: 1С-Паблишинг; СПб.: Питер, 2007. — 512 с.: ил.
- 10. Профессиональная разработка в системе 1С:Предприятие 8 / Под ред. М. Г. Радченко. М.: 1С-Паблишинг; СПб.: Питер, 2006. 816 с.: ил.
- 11. Хрусталева Е. Ю. Разработка сложных отчетов в 1С:Предприятии 8. М.: 1С-Паблишинг, 2008. — 512 с.: ил. + CD-ROM.

Предметный указатель

Α

Automation Client/Server 299

С

Component Object Model (COM) 299

..

Μ

Microsoft Excel:

- ♦ Cells 302
- CreateObject 311
- ♦ Run() 306
- ♦ Sheets 302

A

Агрегатная функция 188

Б

Булево 53

В

Вид:

- 🛇 данных, непрерывный 326
- ◊ регистра 109

Г

Гиперссылка оповещения 348

Microsoft Word 167

- ActiveDocument 311
- Ocuments 311
- ♦ InsertAfter 312
- Paragraphs 311
 Count 312
- ♦ Range 311

0

OLE 169 ◊ контейнер 299 ◊ сервер 300 OLE Automation 299

Д

Движения 110, 281 Диалог 42 Документ 66, 91

3

Записать() 163 Запрет незаполненных значений 146 Запрос 84, 174, 178 NULL 209
BO3P 180
BЫБОР 213
BЫБРАТЬ 175
ГДЕ 184
ДатаВремя() 217

Запрос (прод.):

- ◊ ДобавитьКДате() 217
- ◊ ИЕРАРХИЯ 182
- ◊ ИМЕЮЩИЕ 196
- ◊ ИТОГИ 197
- ♦ KAK 179
- ◊ КОЛИЧЕСТВО() 188
- ◊ ЛЕВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ 208
- ◊ логические операторы 186
- ◊ МАКСИМУМ() 188
- ◊ МЕЖДУ 194
- ◊ МИНИМУМ() 188
- ◊ ОБЪЕДИНИТЬ 200
- ◊ ПЕРВЫЕ 180
- ◊ ПОДСТРОКА() 216
- ◊ ПОЛНОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ 212
- ◊ ПРАВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ 210
- ◊ ПРЕДСТАВЛЕНИЕ() 219
- ♦ РАЗЛИЧНЫЕ 183
- ◊ РАЗНОСТЬДАТ() 218
- ◊ СГРУППИРОВАТЬ ПО 85, 190
- ◊ СОЕДИНЕНИЕ 207
- ♦ CPEДHEE() 188
- ◊ СУММА() 188
- ◊ УБЫВ 179
- ◊ УПОРЯДОЧИТЬ ПО 179

И

Иерархический справочник 31 Измерение 108 Имя 25 Индексирование 57, 146 Информационная панель 348 История 348

К

Ключевое слово:

- ◊ Для Каждого ... Цикл 65
- ◊ Если 56
- ◊ Иначе 56
- ◊ Истина 57
- ◊ КонецЕсли 56
- ◊ КонецПроцедуры 50
- ◊ Ложь 57
- ◊ Пока ... цикл 51

◊ Процедура 50 ◊ Тогла 56 Кнопка 149 Кол 32 Количество уровней иерархии 31 Коллекция: Параметры 51 ◊ ЭлементыФормы 57 Командный интерфейс 352 Комментарий 25 Конструктор: движений 281 ◊ запроса 226 ◊ настроек 232 ◊ печати 46, 63 Конфигуратор 21, 24

Л

Логические операторы 186

Μ

- Макеты 45 Макросы Excel 305 Максимальное значение 44 Массив 168 Метаданные 17 Метод: ◊ Вставить() 168 ◊ Выбрать() 51, 99, 121, 178
- ◊ Вывести() 51, 329
 ◊ Выполнить() 178
- Добавить() 73, 114, 178
- ◊ Записать() 98, 160
- ◊ Итог() 123
- ◊ Остатки() 123
- ◊ Очистить() 73, 99, 178
- ◊ Показать() 52
- ◊ Получить() 134, 275
- ◊ ПолучитьМакет() 50
- ◊ ПолучитьОбъект() 103
- ◊ ПолучитьПоследнее() 134, 275
- ◊ Прочитать() 163
- ◊ Следующий() 51, 99, 178
- ◊ СоздатьДокумент() 97, 106
- ◊ СоздатьМенеджерЗаписи() 163
- ◊ СоздатьНаборЗаписей() 160
- ◊ СрезПоследних() 279
- ◊ ТипЗаписи() 198
- ◊ УстановитьПараметр() 130, 193

Минимальное значение 44 Модуль 42, 46 ◊ объекта 107

Η

Надпись 148 Наименование 32 Независимый режим записи 126, 140

0

Область системных команд 349 Обороты 109, 118, 280 Обработка 95, 147 выбора 288 Общая статистика 325 Объединение результатов запросов 200 Объект: ОСОМОбъект 169 ◊ АнализДанных 328 ДокументОбъект 103 ОпостроительОтчетаАнализаДанных 329 ◊ Структура 57, 102 ТабличныйДокумент 76 Окно редактирования объекта конфигурации 25 Операторы: ◊ Для 19 ◊ Для каждого 19 ♦ Если 20 Новый 20 ◊ Пока 20 Прервать 20 Основные формы списка 39 Остатки 109, 122 Отладка 28 Отчет 77

Π

Панель навигации 347 Параметр отчета 80 Параметры виртуальной таблицы 283 Перечисление 25 Период 132, 156, 195 Периодичность 144, 269 Подсистема 346 Подчинение регистратору 126, 140 Поле ввода 148 Приход 114 Произвольная форма 147 Процедура: ◊ ОбработкаПроведения() 114 ◊ ПриОткрытии() 151

Ρ

Рамка группы 147 Расход 114 Регистр накопления 108 ◊ оборотный 118 ◊ остатков 109 Регистр сведений 125, 126, 140 ◊ периодический 132 Регистры накопления 111 Редактирование в диалоге 37 Режим: ◊ 1С:Предприятие 17, 22 ◊ конфигуратора 17 Результат анализа данных 321 Реквизит 32, 92 Ресурс 108

С

Свойство: ◊ Зашита 52 ◊ Индексировать 56 ОтображатьЗаголовки 52 ◊ ОтображатьСетку 51 ◊ Параметры 51 ◊ ТипАнализа 328 ◊ ТолькоПросмотр 52 ЭтоГруппа 211 Секшия отчета 78 Синоним 25 Система компоновки данных 222 Событие: ПередУдалением 107 ПриЗаписи 107 ◊ ПриИзменении 72, 152, 275 ◊ ПриОткрытии 151 Список значений 176 Справочник 27 иерархический 31 Ссылка 50, 87 Схема компоновки данных 222

Т

Табличная часть 33, 68 Текст шапки 43 ◊ АнализДанныхПоискАссоциаций 329
◊ ТабличныйДокумент 50
Тип данных:
◊ Null 19
◊ Булево 19, 53, 57
◊ Запрос 84
◊ число 19
Тип движения регистра 114
"Толстый" клиент 358
"Тонкий" клиент 359

У

Управляемые формы 354

Φ

Флажок 53 Форма 30 ◊ диалога *См*. Форма элемента ◊ списка 36

◊ элемента 36

Функция:

- ◊ Год() 152
- ◊ КонецДня() 99
- ◊ Месяц() 152
- ◊ НачалоДня() 99
- ◊ Сообщить() 123
- ◊ СтрДлина() 301
- ◊ Строка() 178
- ◊ ТекущаяДата() 98, 152

Ц

Цикл 51, 65

ш

Шапка 67, 92

Я

Язык запросов 17