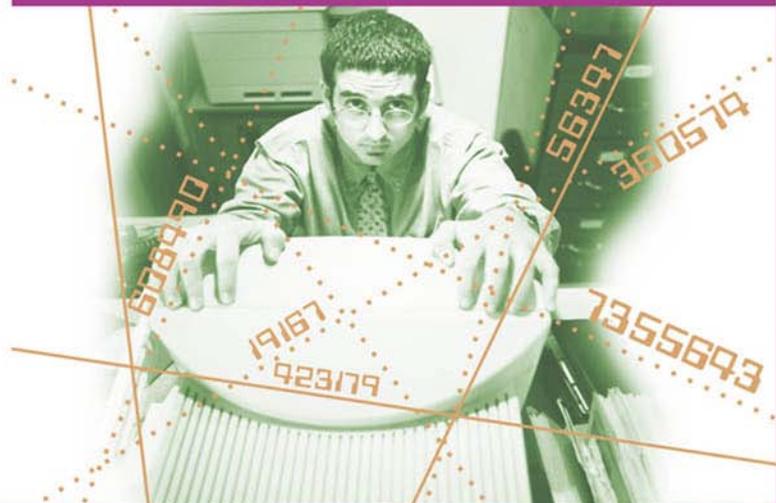


1С: Предприятие

Секреты конструирования

- Администрирование БД
- Элементы интерфейса
- Атрибуты и методы метаданных
- Структура информационной базы
- Структура метаданных



1С

Наталья Рязанцева
Дмитрий Рязанцев

***1С*: Предприятие** **Секреты конструирования**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06
ББК 32.973.26-018.2
Р99

Рязанцева Н. А., Рязанцев Д. Н.

Р99 1С:Предприятие. Секреты конструирования. — СПб.:
БХВ-Петербург, 2005. — 368 с.: ил.

ISBN 978-5-94157-628-9

Книга посвящена изучению приемов и методов конструирования конфигурации в Конфигураторе "1С:Предприятие" с применением компонент, используемых при разработке конфигураций "Бухгалтерский учет", "Оперативный учет" и "Расчет". Для лучшего понимания логики конструирования приведена структура информационной базы всех трех компонент. Материал охватывает все стадии создания конфигурации: от постановки задачи до администрирования базы данных. Представлена структура метаданных, назначение метаданных иллюстрируется уникальными примерами разработки новой конфигурации "Коммунальные услуги". Также рассматриваются разнообразные отчеты, используемые при модифицировании уже существующих конфигураций. Описан механизм администрирования баз данных. Для создания элементарных процедур приведен синтаксис встроенного языка. Рассматриваются ошибки и сложности, которые могут возникнуть в процессе работы.

Для 1С-программистов и пользователей

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Татьяна Лапина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Игоря Цырульниковой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 15.07.05.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 29,67.

Тираж 4000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 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

ISBN 978-5-94157-628-9

© Рязанцева Н. А., Рязанцев Д. Н., 2005
© Оформление, издательство "БХВ-Петербург", 2005

Оглавление

Предисловие	7
Введение	9
Постановка задачи	10
Анализ информации.....	10
Построение алгоритма.....	12
Выбор компоненты	13
Блок-схема задачи.....	13
Глава 1. Конфигурация.....	15
Глава 2. Файловая реализация конфигурации.....	32
Основные файлы конфигурации.....	33
Структура файла 1Cv7.md	33
Реквизиты	33
Форма	37
Структура файла 1Cv7.dd.....	40
Основные файлы базы данных.....	42
Общие обязательные таблицы для всех конфигураций	42
Структура файла 1susers.dbf	42
Структура файла 1ssystem.dbf.....	43
Структура файла 1sconst.dbf.....	43
Структура файла 1sjourn.dbf.....	45
Структура файла 1scrdoc.dbf.....	46
Структура файла 1sdnlock.dbf.....	47
Файлы компоненты "Расчет"	47
Файлы компоненты "Оперативный учет"	50
Структура файла 1sstream.dbf	52
Файлы компоненты "Бухгалтерский учет"	54
Структура файла 1saccs.dbf.....	54
Структура файла 1soper.dbf.....	56

Структура файла lsentry.dbf.....	58
Структура файла lsbkttlc.dbf.....	60
Структура файла lsbkttl.dbf.....	62
Структура файла lscorent.dbf.....	64
Структура файла lsacsel.dbf.....	65
Структура файла lstoper.dbf.....	66
Структура файла lssbsel.dbf.....	67
Файлы компоненты "Расчет"	68
Глава 3. Элементы диалога.....	71
Элементы диалога.....	71
Реквизиты.....	73
Действия с элементами диалога.....	82
Выбор элемента	82
Перемещение	83
Изменение размеров.....	83
Автоматическая установка размеров.....	84
Выравнивание	85
Выравнивание по образцу.....	85
Центрирование элемента.....	86
Удаление элементов диалога	86
Отмена сделанных изменений.....	86
Определение порядка обхода элементов диалога.....	87
Работа с таблицей.....	87
Определение свойств элементов диалога.....	88
Кнопки диалога свойств.....	89
Вкладка <i>Общие</i>	89
Вкладка <i>Тип</i>	91
Вкладка <i>Дополнительно</i>	92
Вкладка <i>Шрифт</i>	98
Вкладка <i>Описание</i>	99
Диалог	99
Слои	100
Синтаксис и конструкции встроенного языка	106
Процедура.....	107
Список параметров.....	108
Предварительное объявление процедуры.....	109
Функция	110
Оператор условного выполнения.....	112
Оператор цикла	115
Оператор <i>Попытка...Исключение</i>	116
Оператор <i>ВызватьИсключение</i>	119
Оператор <i>Перейти</i>	119
Оператор <i>Продолжить</i>	121

Оператор <i>Прервать</i>	121
Оператор <i>Возврат</i>	121
Директивы.....	122
Алгоритм исполнения модулей встроенного языка	124
Процедуры и функции элементов формы	124
Предопределенные процедуры	127
Статус возврата.....	127
Предопределенные процедуры справочников	127
Предопределенные процедуры документов	135
Предопределенные процедуры журнала документов.....	144
Предопределенные процедуры журнала расчетов.....	145
Предопределенные процедуры контекста формы	151
Глобальный модуль.....	159
Предопределенные процедуры Глобального модуля	161
Навигация в теле модуля	171
Синтакс-Помощник.....	171
Глава 4. Назначение метаданных	174
Справочники.....	176
Подчинение.....	179
Реквизиты.....	180
Документы	183
Реквизиты.....	185
Правила нумерации документа	185
Проведение документа.....	187
Ввод на основании	189
Константы	192
Журналы	196
Перечисления	208
Отчеты.....	210
Встроенные отчеты	212
Шаблон	212
Процедура.....	249
Оперативные отчеты	250
Диалог	250
Шаблон	258
Процедура.....	259
Внешние отчеты	263
Регламентированные отчеты	271
Планы счетов.....	274
Редактирование плана счетов	277
Ввод нового счета (субсчета)	279
Операция.....	284
Проводка	287

Регистры.....	289
Виды расчетов.....	292
Группы расчетов.....	295
Правила пересчета.....	296
Журналы расчетов.....	297
Глава 5. Пользователи	301
Рабочий каталог	303
Интерфейсы	307
Конструктор меню	313
Подчиненный интерфейс.....	314
Панели инструментов интерфейса.....	315
Права	318
Права Администратора.....	328
Дата запрета редактирования.....	328
Панели инструментов Конфигуратора	329
Настройка параметров.....	333
Глава 6. Взаимодействие конфигураций.....	338
Восстановление базы данных	338
Некорректность бухгалтерских итогов	338
Тестирование и исправление информационной базы	340
Ошибка запуска Конфигуратора.....	342
Ошибка открытия файлов.....	342
Обновление конфигурации.....	342
Методические рекомендации по модификации конфигурации	345
Сохранение конфигурации.....	350
Восстановление конфигурации.....	353
Глава 7. Несколько слов о распределенной ИБ.....	355
Префиксы.....	355
Автообмен.....	358
Настройка параметров автообмена.....	359
Настройка параметров миграции объектов метаданных	363
Настройка параметров миграции операций.....	364
Настройка параметров миграции регистров.....	365

Предисловие

Мечта каждого пользователя продуктов "1С:Предприятие" — уметь вносить мелкие изменения в отчеты и печатные документы, не обращаясь по таким "пустякам" к программистам. Мечта каждого начинающего программиста — освоить быстро все нюансы работы с Конфигуратором, не затрачивая уйму времени на подтверждение своих догадок экспериментальным путем. Оплата труда большинства программистов сегодня зависит от скорости разработки и выполнения поставленной задачи.

Книга составлена таким образом, чтобы в первых главах дать представление о логике функционирования всех составляющих Конфигуратора в целом на простом примере, не загружая читателя излишними деталями.

Описание общих элементов интерфейса — шаг к разработке удобного пользовательского интерфейса.

Во избежание дилетантских действий с файлами конфигурации и базы данных приводится описание их структуры.

Детальное описание структуры метаданных позволяет корректно работать с базой данных, добавляя новые объекты и изменяя существующие.

Особенно важно овладеть навыками проектирования новых разнообразных отчетов и изменения существующих. Описание синтаксиса встроенного языка на базе примеров — неотъемлемая часть данного процесса.

Администрирование базы данных, пользователей и прав — тот раздел Конфигуратора, который вполне доступен пользователям, не владеющим навыками программирования, то есть простым бухгалтерам.

Круг читателей этой книги настолько широк, насколько широк спектр интересов пользователей продуктов "1С:Предприятие". Каждый найдет в ней тот раздел, который поможет овладеть навыками работы в своей предметной области.

Введение

С чего начинается конструирование любого объекта? С построения макета объекта и модели его функционирования. Для организации учета деятельности предприятия необходим информационный макет и задание алгоритма обработки его элементов. Таким макетом с встроенным алгоритмом взаимодействия элементов является конфигурация. Для создания конфигурации предназначен программный инструмент 1С:Предприятие — Конфигуратор.

Конфигурацией в системе 1С:Предприятие называется совокупность трех взаимосвязанных составных частей:

- структуры метаданных;
- набора пользовательских интерфейсов;
- набора прав.

Метаданными (данными о данных) в системе 1С:Предприятие называется совокупность объектов данных, настроенных на хранение и обработку информации о хозяйственной деятельности конкретного предприятия.

Пользовательским интерфейсом в системе 1С:Предприятие называется совокупность команд главного меню и панелей инструментов, настроенных на работу с конкретными объектами данных — документами, справочниками, журналами и т. д. Как правило, пользовательский интерфейс создается для конкретной категории пользователей. Цель создания интерфейса — обеспечить быстрый доступ пользователей к той информации, которая необходима им в соответствии с их обязанностями.

Права в системе 1С:Предприятие определяют полномочия пользователей на работу с информацией, которая обрабатывается в системе. Совокупность предоставляемых пользователю прав определяется, как правило, кругом его обязанностей.

Основными элементами конфигурации являются метаданные. Метаданные определяют классы объектов определенного вида. Для четкой постановки

задачи необходимо очень хорошо знать назначение и место каждого из метаданных в любой конфигурации. Набор метаданных строго определен для каждой конфигурации. Разработчик может задавать сколько угодно своих объектов, но в рамках заданного вида метаданных. Добавить новый вид метаданных невозможно.

В данной книге мы шаг за шагом пройдем построение конфигурации от постановки задачи до ее реализации. Конечная цель — создание конфигурации с учетом приобретенных в процессе изучения навыков.

Постановка задачи

Прежде всего необходимо определить информационные потоки; характер задачи с точки зрения оперативности получения, ввода и хранения информации; отчетность и периодичность ее формирования. Как правило, заказчик формулирует задачу нечетко, даже если она ставится в рамках уже созданной конфигурации. Формализация задачи в виде детальной блок-схемы алгоритма представляет шаблон, который реализовать средствами Конфигуратора значительно проще. С другой стороны, владея инструментом Конфигуратора, можно детализацию блок-схемы не доводить до абсурда.

Анализ информации

Информация об организации, ФИО директора и главного бухгалтера, телефоны организации и другие постоянные величины, необходимые для расчетов и формирования отчетной документации задаются в конфигурации метаданными типа "Константа".

Для начала определяем основной объект учета. В случае коммунальных услуг основным объектом учета является *квартира*, поскольку все начисления за коммунальные услуги производятся на конкретный адрес. Хотя квартиры, подлежащие учету, относятся к классу постоянной информации, информация о них меняется на протяжении периода учета, следовательно, подлежит редактированию. Кроме того, список квартир также может изменяться.

К коммунальным услугам относится и капитальный ремонт домов. Следовательно, необходимо вести информацию о типе домов, дате ввода домов в эксплуатацию, их состоянии и плановых сроках капитального ремонта. Таким образом, определился еще один объект учета — *дома*.

Для формирования справок "Выписка из лицевого счета" необходимо вести учет жильцов, даты рождения и родственные отношения, следовательно, еще один объект учета — *жильцы*.

Для расчетов с поставщиками и клиентами традиционно необходимы объекты учета — *контрагенты, услуги и тарифы*.

Все объекты учета задаются в конфигурации метаданными типа "Справочник".

На следующем этапе определяется документооборот предприятия. Все движения суммовых и количественных показателей по объектам учета производятся первичными документами. Документы имеют либо стандартную форму, утвержденную Госкомстатом России, либо внутреннюю форму предприятия, утвержденную руководством данного предприятия. Для нашей учебной конфигурации определены следующие документы:

- "Показания водомеров";
- "Ввод начального сальдо";
- "Услуги сторонних организаций";
- "Заявка";
- "Оплата".

Все первичные документы задаются в конфигурации метаданными типа "Документ".

В процессе формирования документов должны быть предусмотрены соответствующие расчеты с формированием соответствующих выходных документов. В нашем случае это расчеты начислений за коммунальные услуги. Все выходные документы также задаются в конфигурации метаданными типа "Документ".

В обилии документов должна быть предусмотрена группировка документов по типу и параметры отбора. Для быстрого доступа к нужным документам предназначены объекты метаданных типа "Журнал". Для нашей конфигурации запрашивается группировка:

- Начисления;
- Оплата;
- Ремонт.

Финалом функционирования задачи является пакет регламентированных документов и аналитические отчеты. Определим для нашей конфигурации следующие основные отчеты:

- Наряд на ремонт;
- Извещение-квитанция;
- Извещения-квитанции;

- Расчетная ведомость;
- Ведомость должников;
- Анализ состояния домов;
- Выписка из лицевого счета.

Все отчеты задаются в конфигурации метаданными типа "Отчет".

Построение алгоритма

Для представления полного алгоритма задачи иногда бывает полезно раскручивать алгоритм с конца. После анализа выходной информации логическая цепочка выстраивается из требований входной информации каждого этапа. В нашем случае выходной информацией по квартирам является пакет извещений на оплату, расчетная ведомость и накопление сальдо по квартирам. Исходим из информации, которая должна быть в извещении-квитанции на оплату:

- Получатель — информация об организации;
- Плательщик — информация о владельце квартиры;
- Адрес квартиры;
- Сальдо на начало периода начисления;
- Общая сумма начисления;
- Общий метраж;
- Количество жильцов;
- Список видов платежей;
- Список тарифов по видам платежей;
- Список сумм, начисленных по всем видам платежей;
- Список сумм, начисленных по всем видам платежей с учетом льгот;
- Список перерасчетов по тем видам начислений, по которым производился перерасчет.

Вся информация в извещении-квитанции хранится в справочниках, кроме сумм начислений и сальдо на начало периода оплаты. Суммы начислений и сальдо могут накапливаться в таблицах следующих видов:

- Бухгалтерские итоги;
- Виды расчетов;
- Регистры.

Правильный вид таблицы накопления по объектам учета влияет на эффективность работы задачи. Выбранный вид накопления итогов определяет компоненту, которая будет использоваться при расчетах.

Выбор компоненты

Технологическая платформа, кроме механизмов, используемых во всех продуктах 1С:Предприятия, включает три основные функциональные компоненты. Функциональные компоненты включаются в состав продуктов системы 1С:Предприятие, использующих специфические возможности компонент.

Компонента "Бухгалтерский учет" предназначена для ведения учета на основе бухгалтерских операций. Она обеспечивает ведение планов счетов, ввод проводок, получение бухгалтерских итогов. Компонента используется для автоматизации бухгалтерского учета в соответствии с любым законодательством и методологией учета.

Компонента "Расчет" предназначена для выполнения сложных периодических расчетов. Она может использоваться для расчета заработной платы любой сложности, расчетов по ценным бумагам и других видов расчетов.

Компонента "Оперативный учет" предназначена для учета наличия и движения средств в самых различных разрезах в реальном времени. Она используется для учета запасов товарно-материальных ценностей, взаиморасчетов с контрагентами и т. д. Компонента позволяет отражать в учете операции хозяйственной жизни предприятия непосредственно в момент их совершения.

Поскольку в нашем случае явно выраженный периодический расчет (ежемесячно) с выставлением абонентской платы за предоставление постоянных услуг, целесообразно выбрать для расчетов компоненту "Расчет". Для обеспечения ведения бухгалтерского учета одновременно с расчетами можно использовать две или три компоненты одновременно в одной конфигурации, однако такие сложные конфигурации используют очень много ресурсов и требуют высокой производительности сервера и сети. В отсутствие данных возможностей лучше использовать отдельные конфигурации с выгрузкой проводок в конфигурацию "Бухгалтерский учет".

В нашем случае будут использованы одновременно в одной конфигурации компоненты "Бухгалтерский учет" и "Расчет". Для иллюстрации использования метаданных типа "Регистр" введем подзадачу обслуживания заявок на ремонт с компонентой "Оперативный учет".

Блок-схема задачи

Квинтэссенцией предварительных действий по постановке задачи является блок-схема задачи (рис. В1).

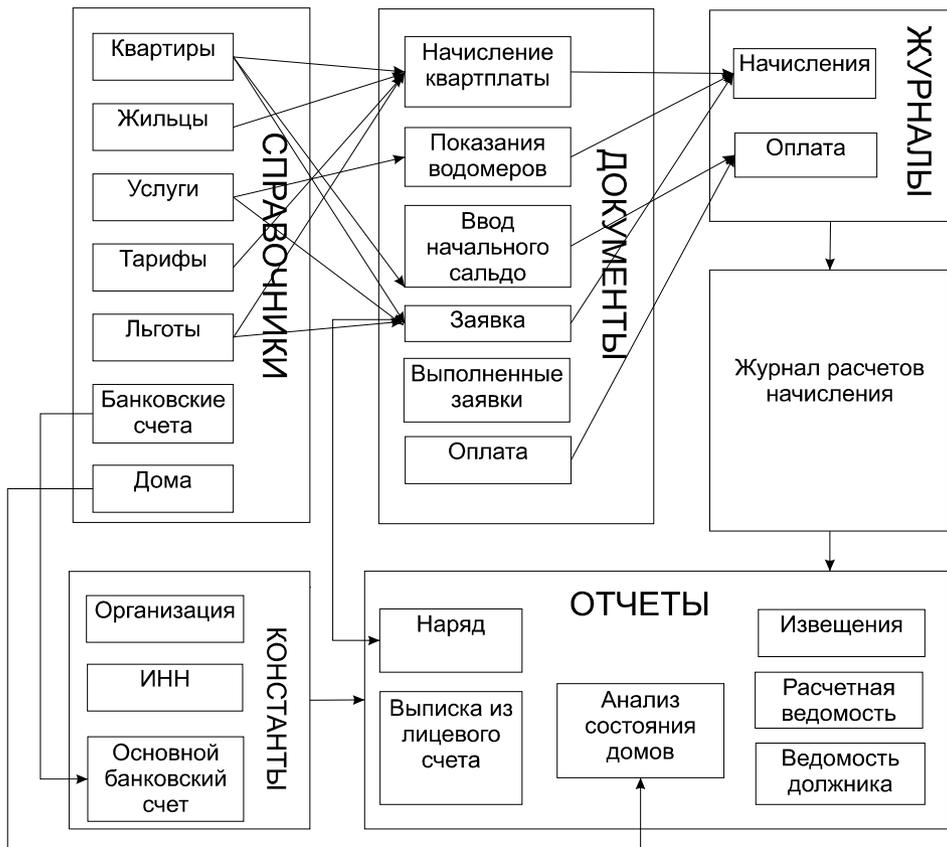


Рис. В1. Блок-схема задачи по начислению квартплаты

ГЛАВА 1

Конфигурация

Создадим новую конфигурацию "Коммунальные услуги". Для этого в окне **Запуск 1С:Предприятия** необходимо в верхнем левом углу выбрать режим **Конфигуратор** и нажать кнопку **Добавить** (рис. 1.1).

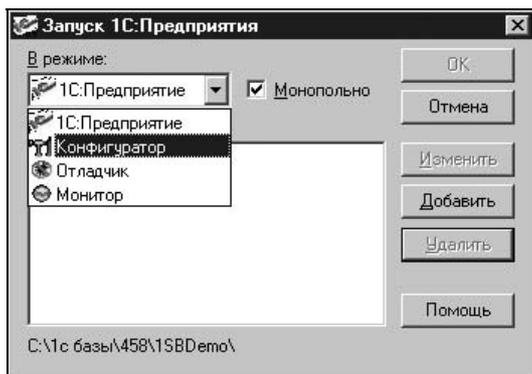


Рис. 1.1. Окно запуска Конфигуратора 1С:Предприятие

В окне **Регистрация Информационной Базы** необходимо задать название конфигурации и каталог, в котором будет размещаться наша конфигурация (рис. 1.2).

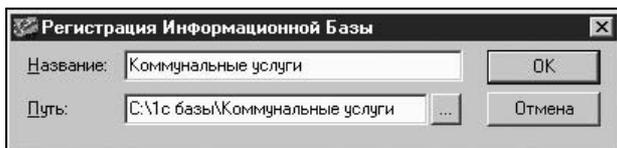


Рис. 1.2. Окно задания параметров конфигурации в списке

По нажатию кнопки **ОК** в процессе запуска в окне **Кодовая страница таблиц информационной базы** необходимо выбрать из списка язык, который будет использован в конфигурации (рис. 1.3).

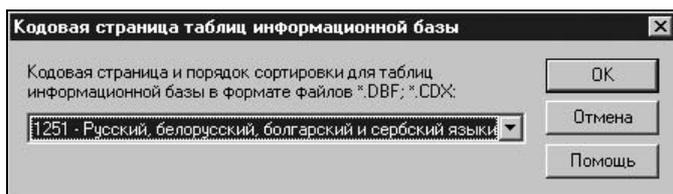


Рис. 1.3. Окно задания языка конфигурации

В процессе запуска будет создан набор необходимых служебных каталогов и файлов, которые определяют конфигурацию. Для открытия созданной конфигурации необходимо в меню **Конфигурация** выбрать пункт **Открыть конфигурацию** (рис. 1.4).

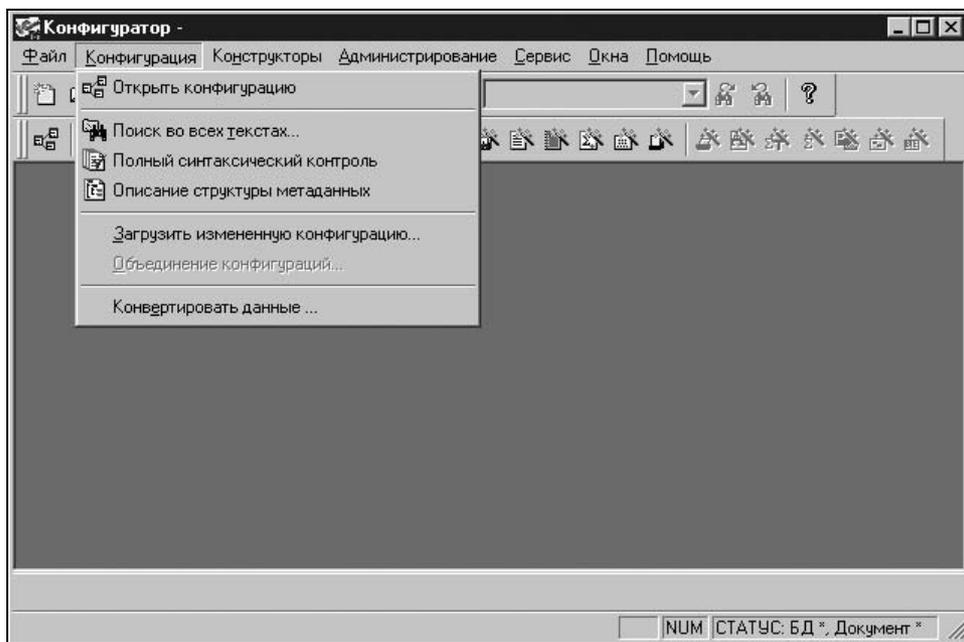


Рис. 1.4. Окно Конфигуратор

Конфигурация открывается в окне **Конфигурация** на вкладке **Метаданные**. В отличие от принятого стандарта расположения вкладок в верхней части окна, в данном случае вкладки располагаются в нижней части окна (рис. 1.5).

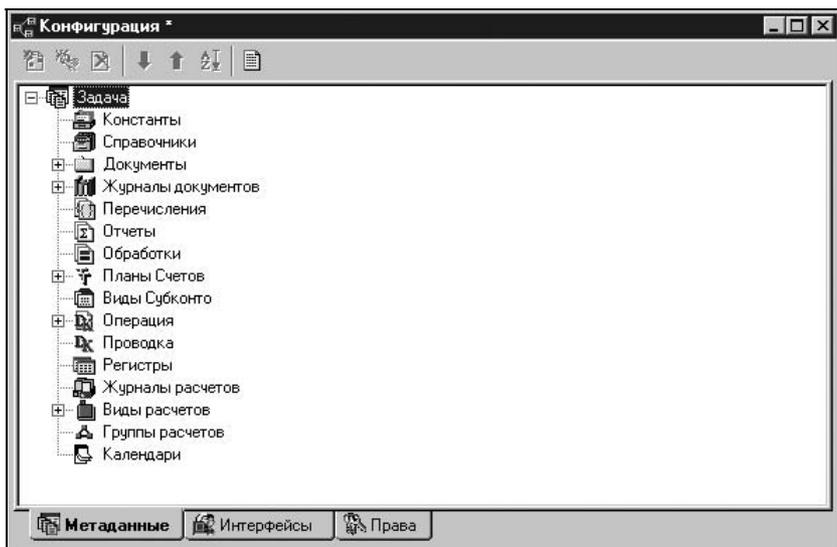


Рис. 1.5. Окно конфигурации на вкладке **Метаданные**

Новая конфигурация по умолчанию называется **Задача**, для нее определен по умолчанию ряд параметров, которые можно изменить в окне **Задача** (рис. 1.6), которое открывается двойным щелчком мыши на данном пункте меню.

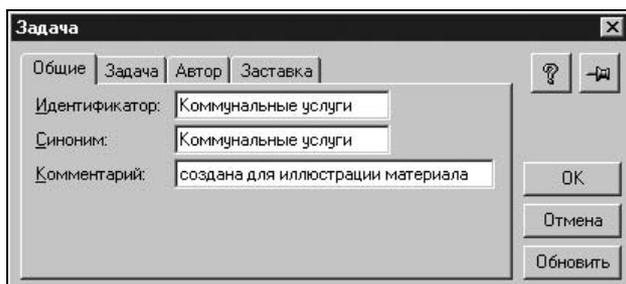


Рис. 1.6. Окно задания общих параметров конфигурации на вкладке **Общие**

На вкладке **Задача** в группе **Основной язык** переключателем задается основной язык.

Реквизит **Округление** устанавливает правила округления результатов числовых выражений. Если выбрана установка **Округлять 1,5 до 1**, то при округлении, например, денежных сумм до целых рублей, суммы до 50 копеек включительно будут отбрасываться, а свыше 50 копеек — округляться до целого рубля. При такой установке сумма 1 рубль 50 копеек будет округлена до

1 рубля, а сумма 1 рубль 51 копейка — до 2 рублей. Если выбрана установка **Округлять 1,5 до 2**, то суммы до 50 копеек будут отбрасываться, а свыше 50 копеек включительно — округляться до целого рубля. Соответственно, сумма 1 рубль 49 копеек будет округлена до 1 рубля, а сумма 1 рубль 50 копеек — до 2 рублей.

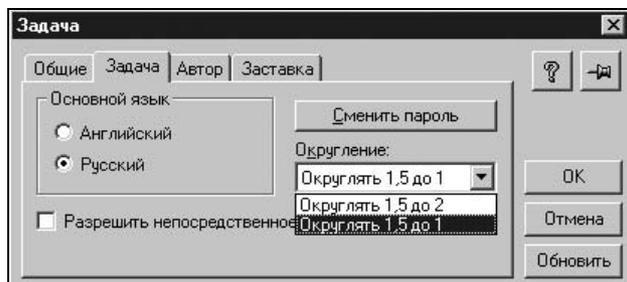


Рис. 1.7. Окно задания общих параметров конфигурации на вкладке **Задача**

Признак **Разрешить непосредственное удаление объектов** включает или отключает механизм контроля ссылочной целостности. Если признак отключен, удаление данных заменяется пометкой на удаление, физическое удаление помеченных объектов выполняется в специальном режиме **Удаление помеченных объектов**. В данном режиме, а также в режиме **Поиск ссылок на объекты** производится отбор объектов, связанных ссылками на удаляемый объект. Если опция **Разрешить непосредственное удаление объектов** включена, механизм контроля ссылочной целостности в системе 1С:Предприятие отключен. В этом случае, если это необходимо, возможность непосредственного удаления объектов может быть ограничена определенным категориям пользователей с помощью настройки прав доступа к конкретным видам объектов метаданных. Установки данного признака на самом деле недостаточно для непосредственного удаления объектов. Для установки режима непосредственного удаления объектов необходимо выполнение следующих условий:

1. Признак **Разрешить непосредственное удаление объектов** должен быть включен.
2. У пользователя должны быть установлены права на непосредственное удаление объектов.
3. В настройке параметров (вкладка **Общие** меню **Сервис** главного меню системы 1С:Предприятие) переключатель **Режим удаления объектов** должен быть установлен соответственно в состояние **Непосредственное удаление**.

Внимание!

Средствами встроенного языка существует возможность непосредственного удаления объектов при любых установках в конфигурации. Поэтому элементы конкретной конфигурации могут выполнять непосредственное удаление в обход механизма контроля ссылочной целостности. В этом случае ответственность за целостность данных лежит на специалисте, выполнявшем конфигурирование системы.

Нажатием кнопки **Сменить пароль** вызывается окно **Смена пароля** для ввода пароля, который пользователь должен указать при открытии окна **Конфигурация** (рис. 1.8).

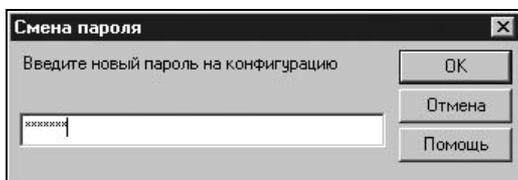


Рис. 1.8. Окно ввода пароля конфигурации

После ввода пароля запрос повторяется для подтверждения пароля (рис. 1.9).



Рис. 1.9. Окно подтверждения пароля конфигурации

Задать информацию об авторе и логотип на вкладке **Автор** можно только с помощью кнопки **Изменить** (рис. 1.10).

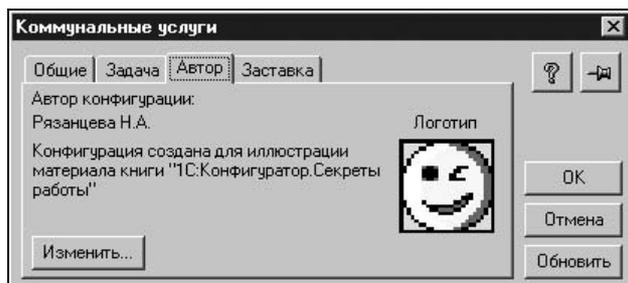


Рис. 1.10. Окно задания общих параметров конфигурации на вкладке **Автор**

Диалог позволяет редактировать информацию об авторе конфигурации и назначать пароль на изменение этой информации. Реквизит **Краткая информация** задает текст, который будет выдаваться в заставке системы в режиме запуска "1С:Предприятие". Длина текста ограничена примерно 60-ю символами. Реквизит **Развернутая информация** задает текст, который будет выдаваться в диалоге **О программе**. Длина текста ограничена размерами поля.

С помощью кнопки **Выбрать** в группе **Логотип** можно выбрать графический файл, содержащий логотип автора конфигурации. Логотип автора конфигурации будет выдаваться в диалоге **О программе**. В группе **Заставка** можно выбрать графический файл, который изменит центральную часть заставки системы 1С:Предприятие, которая выдается в режиме запуска 1С:Предприятие.

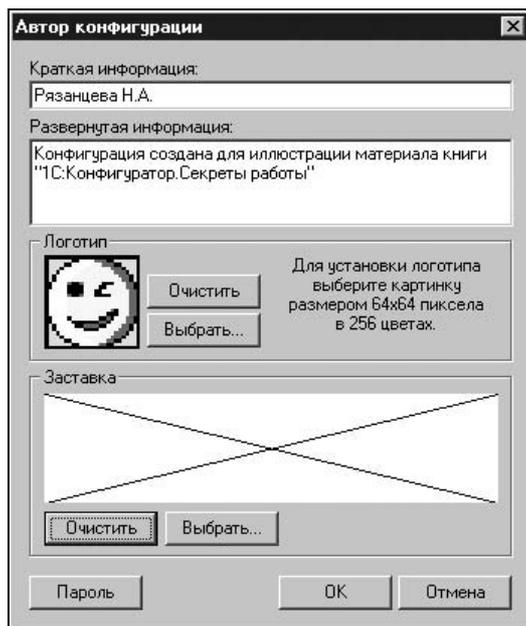


Рис. 1.11. Окно задания информации об авторе и логотипа

В новой конфигурации только определены метаданные, но объектов метаданных нет. Виды метаданных задают ключевые реквизиты файлов базы данных, остальные реквизиты задает разработчик конфигурации. В совокупности метаданные задают базу данных конфигурации.

Проследим простейшую цепочку задания объекта метаданных, отображения его в пользовательском меню и наборе пользовательских прав по редактированию записей. Зададим справочник **Дома**, предварительно отключив при-

знак использования конструкторов для создания новых справочников (меню **Конструкторы**). Для ввода нового объекта метаданных, необходимо выполнить следующие действия:

1. Установить указатель на соответствующем виде метаданного в списке метаданных.
2. В меню **Действия** выбрать пункт **Новый элемент** (рис. 1.12).

Меню **Действия** можно заменить контекстным меню (нажатие правой кнопки мыши).

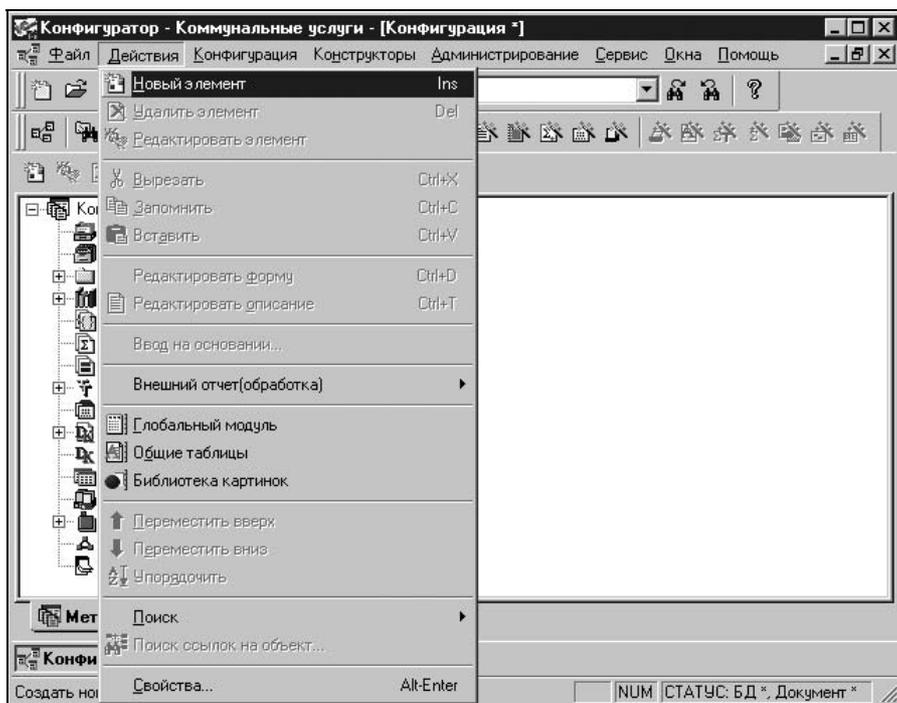


Рис. 1.12. Меню Действия

Создадим новый **Справочник Дома**. После задания нового элемента на строке списка метаданных **Справочники** в окне конструктора справочника заполним информацию о справочнике (рис. 1.13).

Зададим реквизиты в справочнике, не вникая в их смысл (рис. 1.13):

- Идентификатор** — Дома;
- Синоним** — Дома;
- Комментарий** — Инвентарное дело.

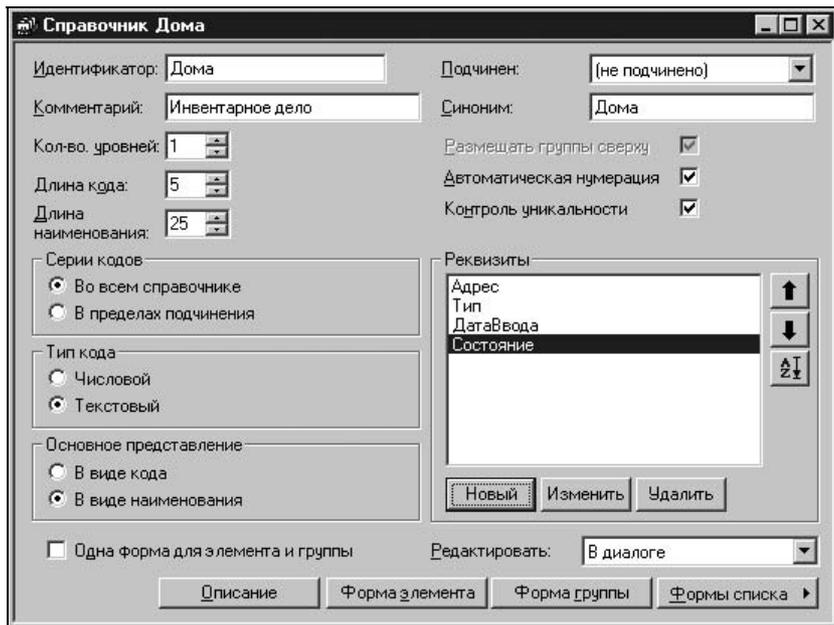


Рис. 1.13. Окно задания реквизитов справочника

Группа **Реквизиты** заполняется с помощью кнопки **Новый**. При нажатии кнопки **Новый** в окне **Свойства Реквизита** заполним реквизиты **Адрес**, **Тип**, **ДатаВвода**, **Состояние** (рис. 1.14).

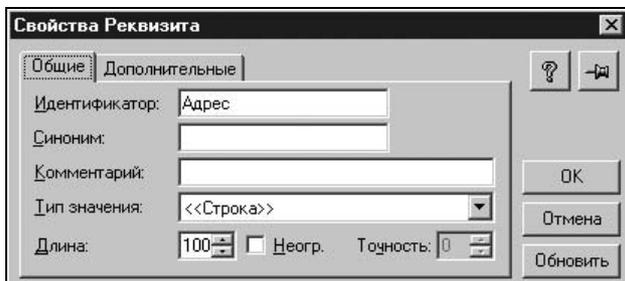


Рис. 1.14. Окно задания свойств реквизита справочника

Внимание!

Во всех полях **Идентификатор** задаются имена реквизитов. В именах недопустимы специальные символы. Имя представляет собой строку длиной не более 128 символов, состоящую из букв, цифр и знаков подчеркивания. Имя не должно начинаться с цифры и содержать пробелы.

В простейшем виде структура объекта метаданных **Справочник Дома** определена. Формы диалогов ведения объектов метаданных задаются автоматически, но могут быть изменены разработчиком для каждого объекта метаданных индивидуально. Из пункта **1С:Предприятие** меню **Сервис** запустим созданную конфигурацию. Кнопка запуска конфигурации  вынесена на панель инструментов **Конфигурация**. После запуска программы доступ к справочнику **Дома** можно получить из пункта **Справочники** меню **Операции** (рис. 1.15).

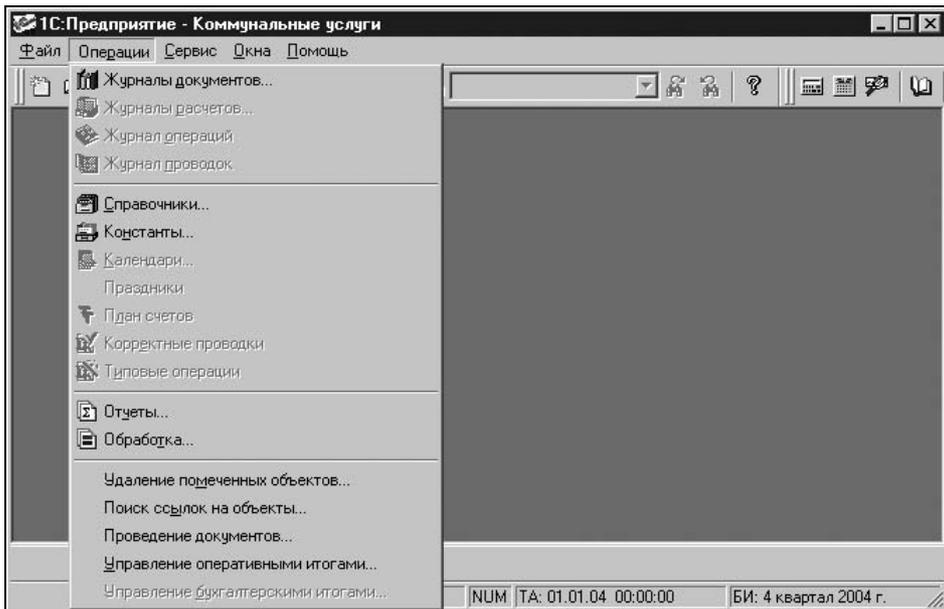


Рис. 1.15. Окно программы 1С:Предприятие меню **Операции**

Выбор из предложенного списка справочников (рис. 1.16) в реальной конфигурации не очень удобен. Список справочников большой, кроме того, в нем могут быть служебные справочники, доступ к которым ограничен. Список объектов метаданных типа "Документ" тоже велик.

Для обеспечения быстрого доступа к справочнику создадим новый пользовательский интерфейс, т. е. произвольный набор меню. Для создания нового интерфейса необходимо на вкладке **Интерфейсы** выбрать пункт **Новый интерфейс** меню **Действия**. В окне **Свойства пользовательского интерфейса** зададим название интерфейса (рис. 1.17).

В списке интерфейсов появился интерфейс с названием **Главный инженер**. Выбором пункта **Редактировать** меню **Действия** или двойным щелчком

мыши на названии интерфейса в списке можно задавать меню главного инженера (рис. 1.18).

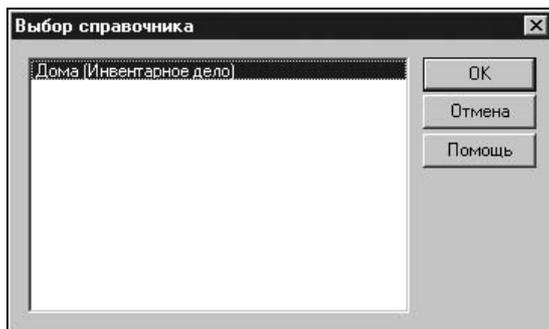


Рис. 1.16. Окно выбора справочника

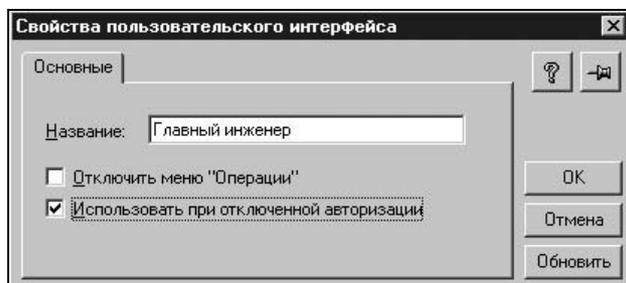


Рис. 1.17. Окно свойств пользовательского интерфейса

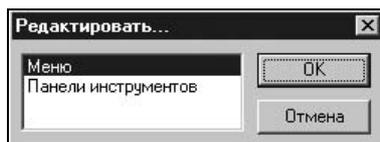


Рис. 1.18. Окно выбора вида редактирования интерфейса

Вспользуемся конструктором для задания набора меню. Для этого в интерфейсе **Главный инженер** в окне **Редактор меню** выберем пункт **Конструктор** меню **Действия** для вызова диалога редактирования интерфейса (рис. 1.19).

Конструктор меню предлагает типовой вариант построения меню конфигурации "1С:Предприятие" (рис. 1.20).

Нажатием кнопки **Построить** завершим пока создание интерфейса.

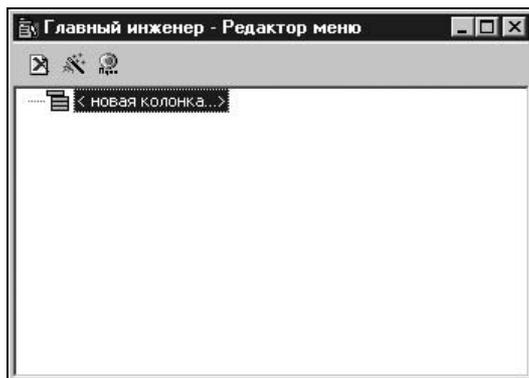


Рис. 1.19. Окно редактирования интерфейса

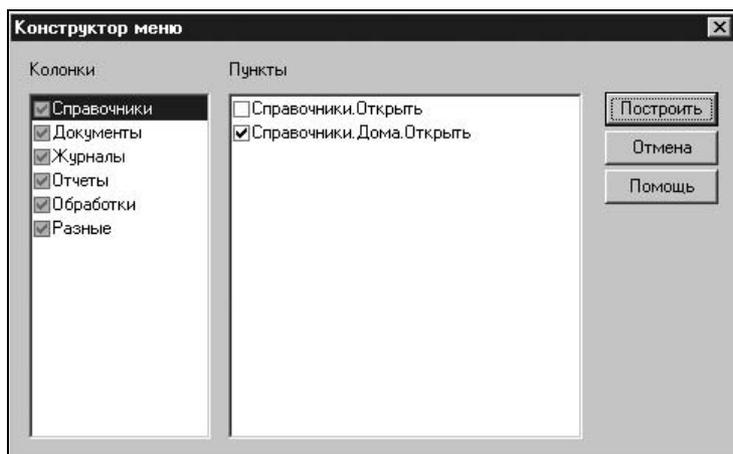


Рис. 1.20. Окно конструктора меню

В результате в конфигурации появилась колонка **Справочники** и меню с пунктом **Дома** (рис. 1.21).

А теперь предположим, что главный инженер может редактировать инвентарное дело дома, а расчетчик не должен иметь доступ к данному справочнику. Для этого создадим новый интерфейс с названием **Расчетчик** и с помощью конструктора создадим меню, в котором отключим признак вхождения в меню справочника **Дома**. А теперь предположим, что в круг обязанностей расчетчика входит ведение справочника **Квартиры**, но главный инженер может только просматривать данный справочник, но не может вносить исправления. В меню должны присутствовать оба справочника. Данное ограничение осуществляется с помощью механизма задания прав.



Рис. 1.21. Окно конфигурации с заданным меню

Создадим справочник "Квартиры", доступ к которому разрешен и тому и другому пользователю (рис. 1.22).

Добавим в оба интерфейса в колонку **Справочники** пункт **Квартиры**. Для этого двойным щелчком мыши на строке <новый> в окне **Редактор меню** вызовем окно **Свойства элемента меню**, зададим название пункта и выберем в качестве объекта **Справочники. Квартиры** (рис. 1.23).

Таким образом, в списке есть два интерфейса: в меню главного инженера присутствуют оба справочника, в меню расчетчика только справочник **Квартиры**. С помощью механизма задания набора прав ограничим главного инженера в редактировании справочника "Квартиры". Для этого зададим его набор прав и назовем его ГИ на вкладке **Права** окна **Конфигурация** (рис. 1.24).

Внимание!

На вкладке **Права** по умолчанию всегда присутствует набор прав **Администратор**. Во избежание ограничения прав на редактирование всех наборов прав не рекомендуется удалять данный набор.

При создании нового набора прав даются полные полномочия для редактирования всех объектов метаданных. Для редактирования набора прав необходимо установить указатель на нужный набор и выбрать пункт **Редактировать** меню **Действия** или дважды щелкнуть мышью на нужном наборе.

Справочник Квартиры

Идентификатор: Подчинен:

Комментарий: Синоним:

Кол-во. уровней: Размещать группы сверху

Длина кода: Автоматическая нумерация

Длина наименования: Контроль уникальности

Серии кодов:

- Во всем справочнике
- В пределах подчинения

Тип кода:

- Числовой
- Текстовый

Основное представление:

- В виде кода
- В виде наименования

Реквизиты:

- Договор
- КоличествоКомнат
- МетражОбщий
- МетражЖилой
- Льгота
- КоличествоЖильцов
- Дом
- ДоговорПриобретения
- Телефон
- ВидЧистаяВода

Новый Изменить Удалить

Одна форма для элемента и группы Редактировать:

Описание Форма элемента Форма группы Формы списка

Рис. 1.22. Окно справочника "Квартиры"

Главный инженер - Редактор меню

Свойства элемента меню

Общие Параметры Акселератор

Название: Тип:

Объект:

Команда:

Подсказка:

Документы

Журналы

Отчеты

Обработки

Журналы расчетов

OK Отмена Обновить

Рис. 1.23. Окно Свойства элемента меню

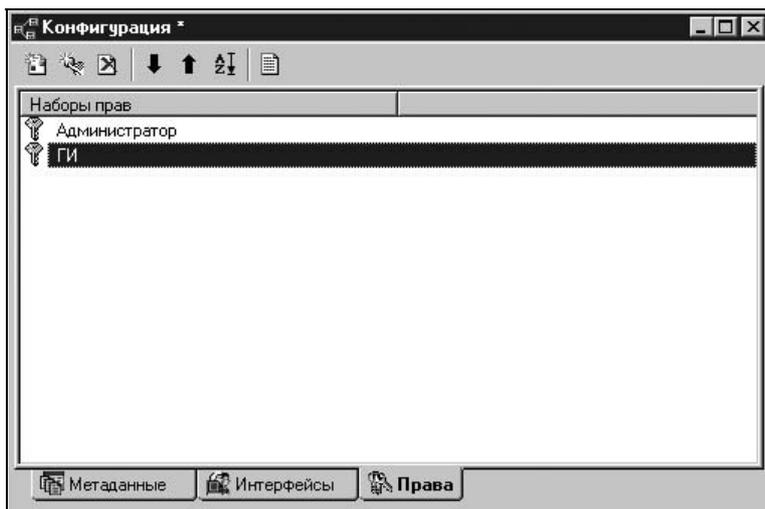


Рис. 1.24. Окно конфигурации на вкладке **Права**

Установим указатель на справочнике **Квартиры** и выберем пункт **Свойства** в меню **Действия**. В окне **Свойства элемента прав** оставим только признак **Чтение** (рис. 1.25).

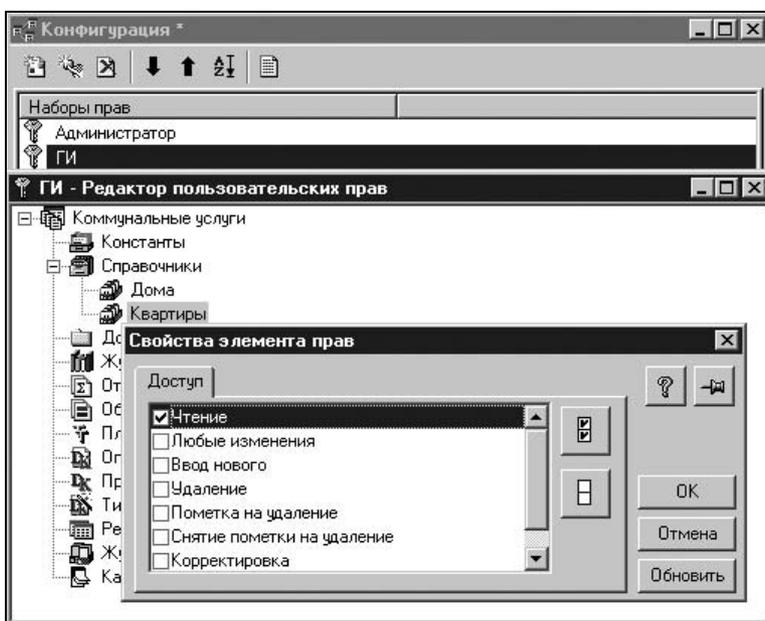


Рис. 1.25. Окно **Свойства элемента прав** справочника "Квартиры"

Добавим также набор прав **Расчет** с полными полномочиями. Осталось определить механизм запуска интерфейса главного инженера и интерфейса расчетчика.

Механизм запуска разных интерфейсов осуществляется с помощью определения круга пользователей и их полномочий (ролей). Зададим двух пользователей: ГЛИ (главный инженер) и РСЧ (расчетчик). Для этого выберем пункт **Пользователи** меню **Администрирование** (кнопка пользователей  вынесена на панель инструментов **Администрирование**) и нажмем клавишу <Insert>, что равносильно выбору пункта **Новый** меню **Действия** (рис. 1.26).

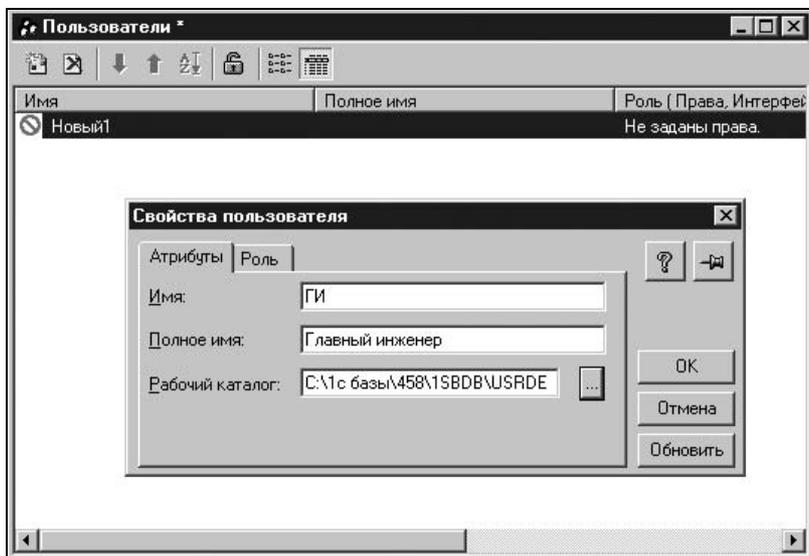


Рис. 1.26. Окно Пользователи на вкладке Атрибуты

На вкладке **Атрибуты** окна пользователей при задании рабочего каталога пользователя рекомендуется в каталоге <каталог БД>/USRDEF создать папку с именем пользователя и выбрать ее в качестве рабочей папки (рис. 1.27).

На вкладке **Роль** окна **Свойства пользователя** предлагается назначить данному пользователю интерфейс выбором из списка существующих интерфейсов и набор прав из списка созданных наборов (рис. 1.28).

Назначим пользователю ГЛИ интерфейс **Главный инженер** и набор прав **ГИ**, а пользователю РСЧ интерфейс **Расчетчик** и набор прав **Расчет**. Чтобы не нарушалась конфиденциальность для каждого пользователя, может быть задан пароль выбором пункта **Сменить пароль** меню **Действия**.

Внимание!

В список пользователей обязательно должен быть включен пользователь с правами администрирования. В противном случае список пользователей не сохраняется (рис. 1.29).



Рис. 1.27. Окно выбора рабочего каталога пользователя

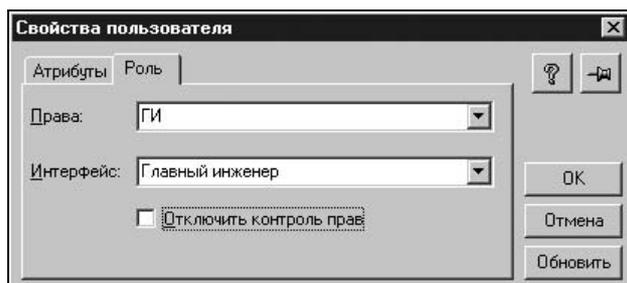


Рис. 1.28. Окно Свойства пользователя на вкладке Роль

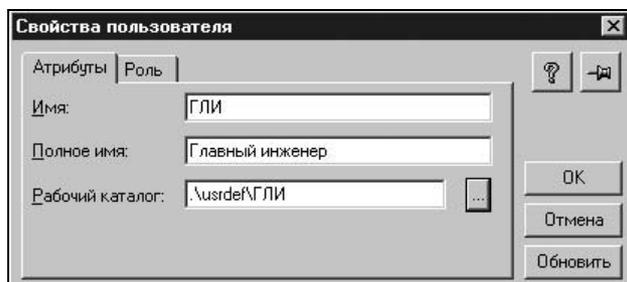


Рис. 1.29. Сообщение об отсутствии администратора

После определения круга пользователей и их полномочий при запуске конфигурации программа предложит выбрать пользователя из списка (рис. 1.30).

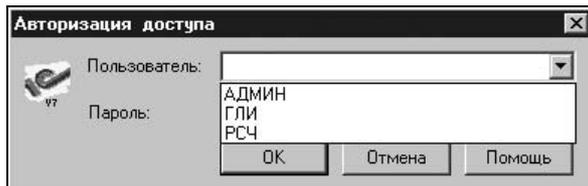


Рис. 1.30. Окно авторизации доступа при запуске программы

Любая попытка главного инженера редактировать справочник **Квартиры** будет пресекаться специальным сообщением (рис. 1.31).

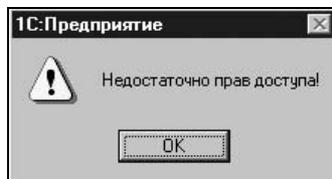


Рис. 1.31. Сообщение об ограничении прав доступа

ГЛАВА 2

Файловая реализация конфигурации

Программное обеспечение "1С:Предприятие" организовано по принципу максимального разделения исполняющей системы и данных. Исполняющая система интерпретирует конфигурацию, которая задает интерфейс пользователя с базой данных (рис. 2.1).

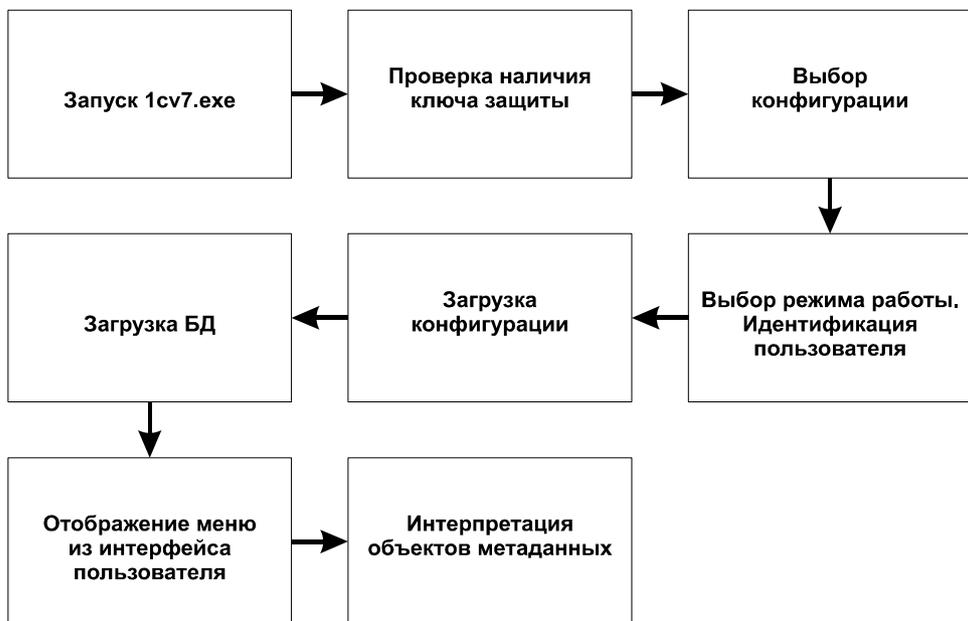


Рис. 2.1. Схема запуска и интерпретации конфигурации

Основные файлы конфигурации

Исполняющий модуль 1Cv7.exe использует для интерпретации два основных файла, описывающих конфигурацию: файл конфигурации 1Cv7.md и словарь базы данных 1Cv7.dd. Исполняющий модуль следит за целостностью информации, любые изменения конфигурации фиксируются в обоих файлах. Все программные процедуры конфигурации хранятся в файле 1Cv7.md в специальном формате и доступны программисту только в Конфигураторе. Разделение описательной информации на два файла удобно для программистов при работе в Конфигураторе, оно освобождает программиста от ведения структуры базы данных. Взаимосвязь данных файлов видна в их структурах.

Структура файла 1Cv7.md

Информацию о конфигурации в файле можно рассматривать в двух аспектах: описание метаданных и обработки событий. Метаданные описываются отдельными фреймами, в которых вложены описания формы, реквизитов, элементов интерфейса и печатных форм, которые обозначены разделителями { и }.

Реквизиты

Рассмотрим созданный нами справочник "Дома". Описание справочника "Дома" представлено в файле 1Cv7.md следующим фрагментом:

```
{ "12", "Дома", "Инвентарное  
дело", "Дома", "0", "5", "1", "1", "2", "25", "1", "1", "1", "13", "13", "0", "1", "1",  
{"Params",  
{"14", "Адрес", "", "", "S", "100", "0", "0", "0", "0", "0", "1", "0", "0", "1", "0", "0"},  
{"15", "Тип", "", "", "S", "10", "0", "0", "0", "0", "0", "1", "0", "0", "1", "0", "0"},  
{"22", "ДатаВвода", "", "", "D", "0", "0", "0", "0", "0", "0", "1", "0", "0", "1", "0", "0"},  
{"23", "Состояние", "", "", "S", "10", "0", "0", "0", "0", "0", "1", "0", "0", "1", "0", "0"}},  
{"Form",  
{"13", "ФормаСписка", "", ""}}},
```

Каждому объекту метаданных присваивается уникальный номер, по которому осуществляются все ссылки. Во фрагменте видно, что номер справочника — 12, номер реквизита "Адрес" — 14. При добавлении нового объекта или реквизита используется текущий номер, поэтому номера реквизитов в рамках одного объекта могут быть непоследовательными. Данные номера используются и в файле 1Cv7.dd.

Рассмотрим соответствие параметров, заданных в Конфигураторе, параметрам в приведенном фрагменте на примере реквизита "Адрес" (рис. 2.2 и рис. 2.5).

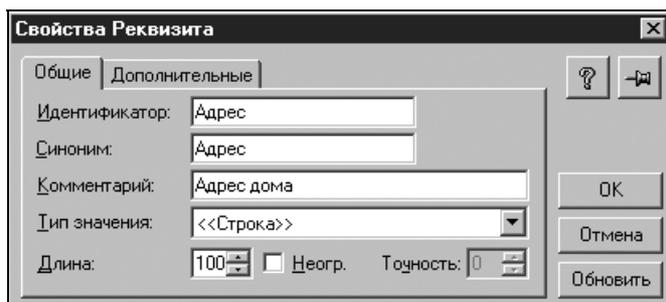


Рис. 2.2. Диалоговое окно **Свойства Реквизита** на вкладке **Общие**

Из фрагмента видно, что:

- Номер объекта** — первый параметр;
- Идентификатор** — второй параметр после номера;
- Комментарий** — третий параметр;
- Синоним** — четвертый параметр;
- Тип значения** — пятый параметр;
- Длина** — шестой параметр;
- Точность** — седьмой параметр;
- Вид заданного типа реквизита** — восьмой параметр.

Тип значения в модуле в данном случае обозначен "S". Данный параметр может принимать следующие значения:

- S — строка;
- N — число;
- O — документ;
- B — справочник;
- E — перечисление;
- T — счет;
- U — неопределенный.

Конкретный вид заданного типа реквизита задается восьмым параметром. В нашем справочнике "Дома" реквизит "Состояние" задается обычной строкой. Ограничим содержание данного реквизита вариантом из перечисленных

возможных состояний домов. Для этого создадим новый объект метаданных типа "Перечисление" с идентификатором "СостоянияДомов" (рис. 2.3).

Назначим реквизиту **Состояние** тип значения — **Перечисление.СостоянияДомов** (рис. 2.4).

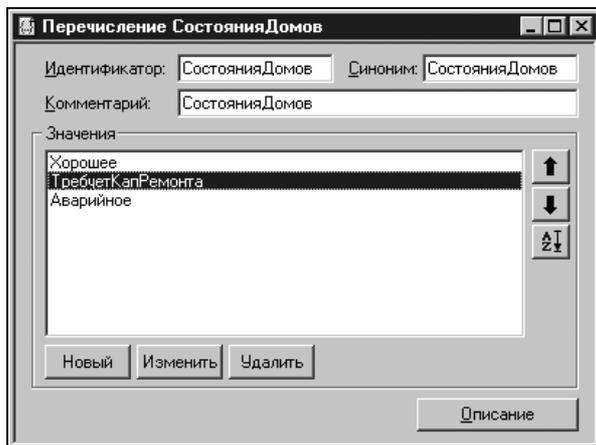


Рис. 2.3. Диалоговое окно перечисления

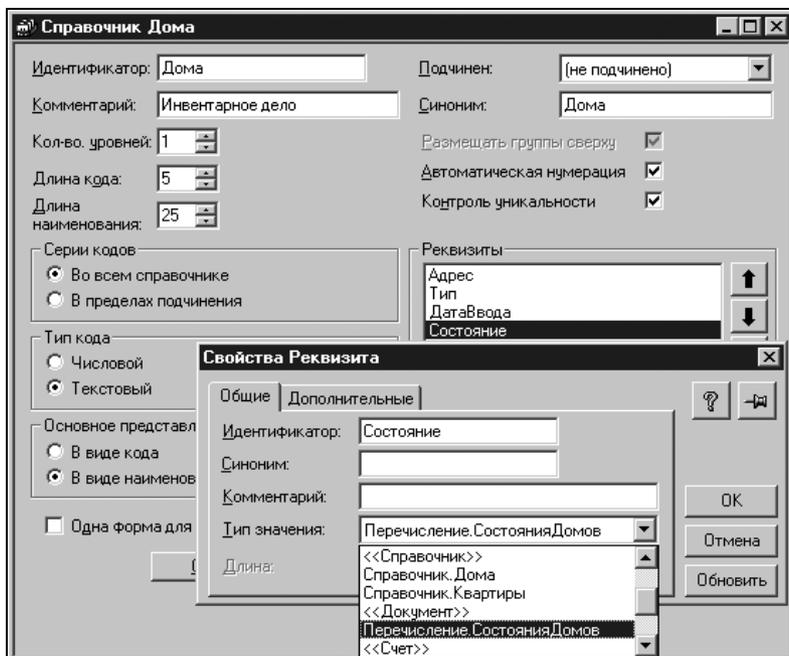


Рис. 2.4. Диалоговое окно справочника "Дома"

Фрагмент описания нашего справочника в файле 1Cv7.md изменится в части описания реквизита "Состояние" следующим образом:

```
{ "12", "Дома", "Инвентарное
дело", "Дома", "0", "5", "1", "1", "2", "25", "1", "1", "1", "13", "13", "0", "1", "1",
{"Params",
{"14", "Адрес", "", "", "S", "100", "0", "0", "0", "0", "0", "1", "0", "0", "1", "0", "0"},
{"15", "Тип", "", "", "S", "10", "0", "0", "0", "0", "0", "1", "0", "0", "1", "0", "0"},
{"22", "ДатаВвода", "", "", "D", "0", "0", "0", "0", "0", "0", "1", "0", "0", "1", "0", "0"},
{"23", "Состояние", "", "", "E", "0", "0", "26", "0", "0", "0", "1", "0", "0", "1", "0", "0"}},
{"Form",
{"13", "ФормаСписка", "", ""}}}
```

Реквизит "Состояние" описан как объект типа "E" со ссылкой на объект с номером 26. В данном случае используется все тот же номер объекта метаданных, присвоенный при создании перечисления "СостоянияДомов" и хранящийся в его описании:

```
{ "26", "СостоянияДомов", "СостоянияДомов", "СостоянияДомов",
{"EnumVal",
{"27", "Хорошее", "Хорошее", "Хорошее", "Хорошее"},
{"28", "ТребуетсяКапРемонта", "ТребуетсяКапРемонта", "Требуется кап. ремонта",
"Требуется кап. ремонта"},
{"29", "Аварийное", "Аварийное", "Аварийное", "Аварийное"} } } }
```

Остальные параметры приведены в соответствии следующим образом:

- Разделять триады** — девятый параметр;
- Неотрицательный** — десятый параметр;
- Периодический** — одиннадцатый параметр;
- Для элемента** — двенадцатый параметр;
- Не используется** — тринадцатый параметр;
- Сортировка** — четырнадцатый параметр;
- Ручное изменение** — пятнадцатый параметр;
- Изменяется Документами** — шестнадцатый параметр;
- Отбор по реквизиту** — семнадцатый параметр.

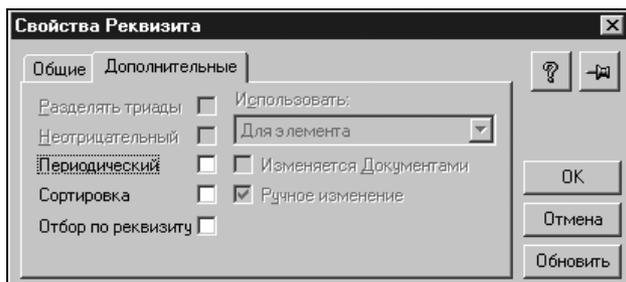


Рис. 2.5. Диалоговое окно **Свойства Реквизита** на вкладке **Дополнительные** для реквизита справочника

Форма

Форма представления объекта метаданных пользователю должна быть удобной не только в разрезе представления информации на экране, но и в разрезе организации контроля и обработки редактируемых данных. Для этого в составе объекта метаданных предназначена форма.

При создании справочника "Дома" в окне **Справочник Дома** (см. рис. 1.13) представлен набор кнопок для задания формы представления диалога с пользователем: **Форма элемента**, **Форма группы**, **Форма списка**. Кнопка **Форма элемента** предназначена для задания формы окна редактирования элемента справочника. При открытии пустой формы автоматически предлагается список реквизитов справочника для размещения их в окне. Разместим все реквизиты в окне автоматически с помощью кнопки **Вставить** (рис. 2.6).

Программа предложит вариант автоматического расположения реквизитов (рис. 2.7).

Зададим свое наименование окна формы элемента. Для этого выделим форму элемента справочника "Дома" (рис. 2.7) и в меню **Действия** выберем пункт **Свойства**. В окне свойств формы зададим заголовок **Иллюстрация формы справочника** (рис. 2.8).

Описание формы элемента справочника "Дома" представлено в файле 1Cv7.md следующим фрагментом:

```
{ "Frame",
{"-11", "0", "0", "0", "400", "0", "0", "0", "204", "1", "2", "1", "34", "MS Sans Serif",
"378", "165", "Иллюстрация формы справочника", "", "", "0", "", "1", "1", "6", "25",
"-1", "0", "0",
{"0",
{"Основной", "1"}}, "1", "1"}},
```

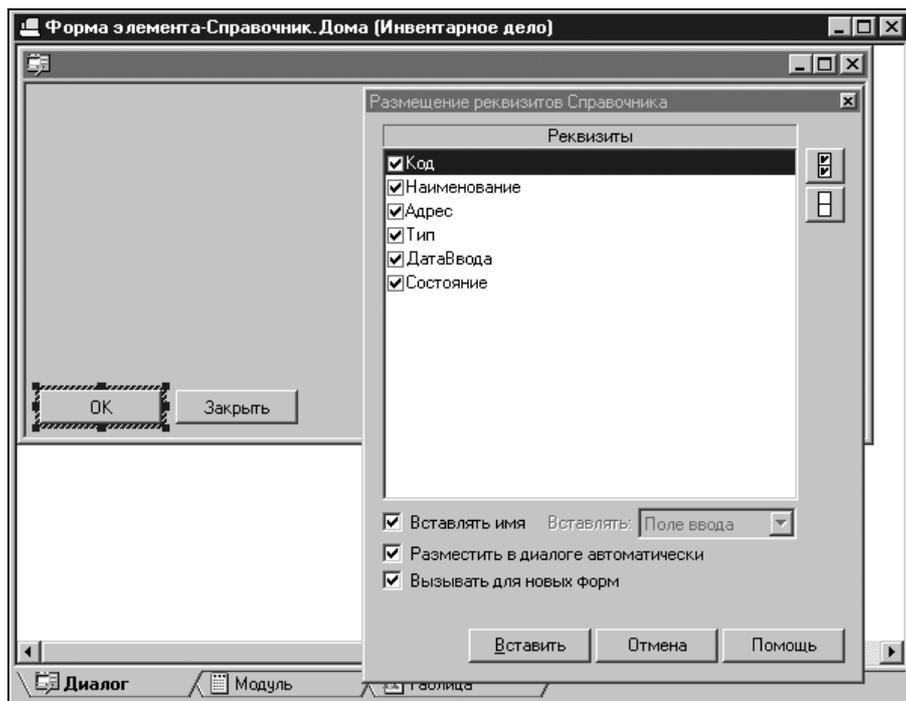


Рис. 2.6. Диалоговое окно размещения реквизитов справочника "Дома"

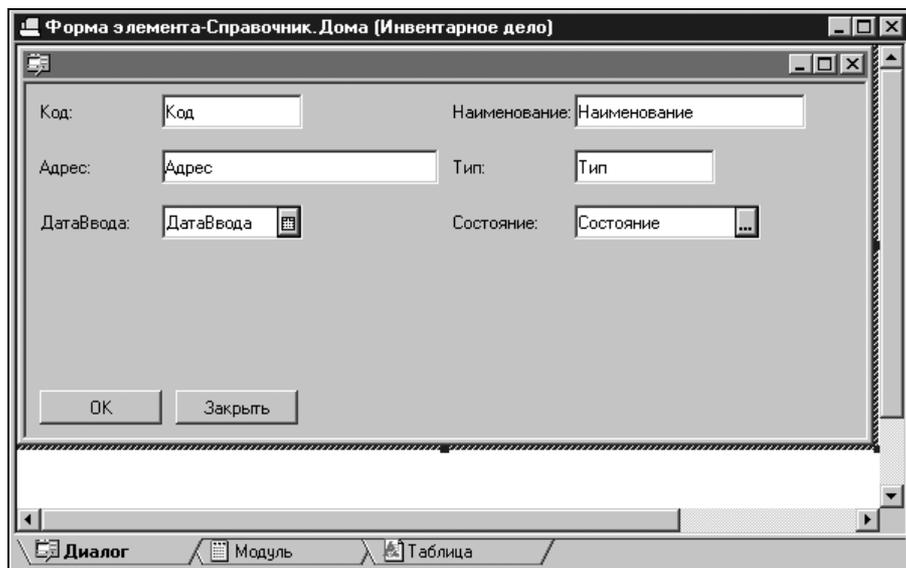


Рис. 2.7. Диалоговое окно формы элемента справочника "Дома"

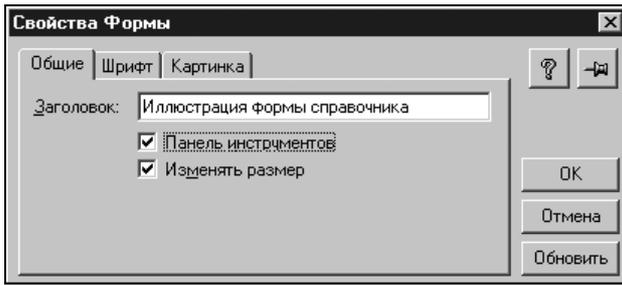


Рис. 2.8. Диалоговое окно свойств формы элемента справочника "Дома"

```

{"Controls",
{"Закреть","B1",-1,"0","Основной","{}0","0"}},
{"Код:","STATIC","1342177792","10","20","18","14","0","0","4165","","","0",
"U","0","0","0","192","","","0",-
11,"0","0","0","0","0","0","0","0","0","0","0","MS Sans Serif",-1,-1,"0",
"Основной","{}0","0"}},
{"Cnt_Ver","10001"}}
{"OK","BUTTON","1342181387","10","142","54","14","0","0","4153","","",
#Записать? Закреть","","-1","U","0","0","0","0","0","0","0","0","0","0","0",
11,"0","0","0","0","0","0","0","0","0","0","0","0","MS Sans Serif",-1,-1,"0",
"Основной","{}0","0"}},
{"Состояние","BMASKED","1350565888","246","66","82","14","1","0","4154",
"","Состояние","23","E","0","0","26","0","0","","","0",-
11,"0","0","0","0","0","0","0","0","0","0","0","0","MS Sans Serif",-1,-
1,"0","Основной","{}0","0"}},
{"Состояние:","STATIC","1342177792","192","66","42","14","0","0","4155",
"","","0","U","0","0","0","0","192","","","0",-
11,"0","0","0","0","0","0","0","0","0","0","0","0","MS Sans Serif",-1,-
1,"0","Основной","{}0","0"}},
{"ДатаВвода","BMASKED","1350565888","64","66","62","14","1","0","4156",
"","ДатаВвода","22","D","0","0","0","0","32","","","0",-
11,"0","0","0","0","0","0","0","0","0","0","0","0","MS Sans Serif",-1,-1,"0",
"Основной","{}0","0"}},
{"ДатаВвода:","STATIC","1342177792","10","66","42","14","0","0","4157",
"","","0","U","0","0","0","0","192","","","0",-
11,"0","0","0","0","0","0","0","0","0","0","0","0","MS Sans Serif",-1,-1,"0",
"Основной","{}0","0"}},
{"Тип","1CEDIT","1350565888","246","43","62","14","1","0","4158","","",
"Тип","15","S","10","0","0","0","0","0","","","0",-11,"0","0","0","0","0","0",
"0","0","0","0","0","0","MS Sans Serif",-1,-1,"0","Основной","{}0","0"}},

```

```
{
  "Тип": "STATIC", "1342177792", "192", "43", "18", "14", "0", "0", "4159", "", "", "",
  "0", "U", "0", "0", "0", "0", "192", "", "", "", "0", "-
  11", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "MS Sans Serif", "-1", "-1", "0",
  "Основной", {"""0""}, {"""0""}},
  {"Адрес": "1CEDIT", "1350565888", "64", "43", "122", "14", "1", "0", "4160", "", "",
  "Адрес": "14", "S", "100", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
  "0", "0", "0", "0", "0", "0", "0", "0", "MS Sans Serif", "-1", "-1", "0", "Основной",
  {"""0""}, {"""0""}},
  {"Адрес": "STATIC", "1342177792", "10", "43", "26", "14", "0", "0", "4161", "", "", "",
  "0", "U", "0", "0", "0", "0", "192", "", "", "", "0", "-
  11", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "MS Sans Serif", "-1", "-1", "0",
  "Основной", {"""0""}, {"""0""}},
  {"Наименование": "1CEDIT", "1350565888", "246", "20", "102", "14", "1", "0",
  "4162", "", "", "Наименование", "-2569", "S", "25", "0", "0", "0", "0", "0", "0", "0", "0", "-11",
  "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "MS Sans Serif", "-1", "-1", "0",
  "Основной", {"""0""}, {"""0""}},
  {"Наименование": "STATIC", "1342177792", "192", "20", "54", "14", "0", "0",
  "4163", "", "", "", "0", "U", "0", "0", "0", "0", "192", "", "", "", "0", "-
  11", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "MS Sans Serif", "-1", "-1", "0",
  "Основной", {"""0""}, {"""0""}},
  {"Код": "1CEDIT", "1350565888", "64", "20", "62", "14", "1", "0", "4164", "", "", "Код",
  "-2568", "S", "5", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
  "0", "0", "0", "0", "0", "MS Sans Serif", "-

```

Из фрагмента видно, что описание размеров и свойств окна формы и всех его элементов интерфейса подобно описанию реквизитов справочника.

Поскольку информация об объекте метаданных хранится в файле 1Cv7.md во фрагментах (фраймах), недостаточно формализованных для быстрого поиска в базе данных, в системе организован словарь базы данных, который находится в файле 1Cv7.dd.

Структура файла 1Cv7.dd

На основе файла конфигурации 1Cv7.md формируется словарь базы данных 1Cv7.dd. База данных состоит из совокупности файлов формата DBASE. Отсутствие файла справочника "Дома" в каталоге базы данных не вызывает сообщение об ошибке, программа создаст новый файл и его индексы для пустого справочника на основании словаря базы данных. Отсутствие же описания справочника в словаре вызовет фатальную ошибку при обращении к справочнику в процессе работы.

Информация о структуре данных справочника хранится в словаре базы данных 1Cv7.dd в следующем фрагменте:

#==TABLE no 8 : Справочник Дома

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=SC12	Справочник Дома	A	SC12	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=ID	ID object	C	9	0
F=CODE	object code	C	5	0
F=DESCR	object description	C	25	0
F=ISMARK	Flag Object is Marke	C	1	0
F=VERSTAMP	Version stamp	C	6	0
F=SP14	(P)Адрес	C	100	0
F=SP14	(P)Тип	C	10	0
F=SP22	(P)ДатаВвода	D	8	0
F=SP23	(P)Состояние	C	9	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=IDD	of ID	0	ID	IDD
I=CODE	of CODE	0	CODE(UPPER)	CODE
I=DESCR	of DESCR	0	DESCR(UPPER)	DESCR

Фрагмент условно делится на три части:

- описание таблицы — TABLE;
- описание полей — Fields;
- описание индексов — Indexes.

Из описания видно, что файл справочника называется sc12.dbf. Имена справочников состоят из префикса sc и номера объекта метаданных в файле конфигурации 1Cv7.md; имена документов, соответственно, из префикса dh и номера. В части описания полей, в которых задаются стандартные для DBASE-файлов тип, длина, точность, представляют интерес поле ID и способ задания имен полей. ID — внутренний код элемента, он недоступен пользователю для редактирования и предназначен для сохранения ссылок при изменении кода. Все ссылки на элементы справочников, документов, перечислений и т. д. делаются с помощью ID. Поля CODE и DESCR — атрибуты справочника **Код** и **Наименование**. Реквизитам присваиваются имена, состоящие из префикса sp и номера из файла конфигурации 1Cv7.md.

Основные файлы базы данных

Кроме справочников и документов, которые необходимы в любой конфигурации, в словаре описывается ряд таблиц метаданных, а также системные таблицы. Для конкретных журналов констант не задаются отдельные файлы. Информация журналов и констант ведется в общих файлах, в отличие от справочников и документов. Поддержка корректной работы в локальной сети, распределение пользователей, а также системная информация осуществляется с помощью файлов, описание которых представлено в начале словаря базы данных.

Общие обязательные таблицы для всех конфигураций

Семь первых таблиц словаря базы данных являются обязательными для всех конфигураций независимо от используемых библиотек. В конце словаря базы данных представлены специфические таблицы, которые характерны для определенных предметно-ориентированных конфигураций, использующих метаданные типа "Операция", "Регистр" или "Расчет". Использование свойств и методов этих объектов метаданных связано с наличием специальных библиотек Account.dll, Trade.dll и Salary.dll. При проектировании конфигурации необходимо учитывать характерные особенности каждой компоненты. В данной главе рассматриваются особенности данных компонент с точки зрения построения файловой системы.

Структура файла 1susers.dbf

В данном файле фиксируются активные пользователи конфигурации в сети.

#===TABLE no 0 : Соединений

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SUSERS	Соединений	A	1SUSERS	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=USRSCNT	Number of users	N	4	0
F=NETCHGCN	Count of changes	N	10	0

Во фрагменте описания файла в словаре базы данных реквизит "Номер пользователя" USRSCNT используется в режиме **Монитор** для связи с файлом описания пользователей users.usr.

Внимание!

Файл users.usr создается при создании первого пользователя и находится в каталоге конфигурации USRDEF. В конфигурации, где не используется список пользователей, данного файла и соответствующего каталога нет.

Структура файла 1ssystem.dbf

Данный файл системного назначения. В нем хранится информация о текущей дате и времени, а также такой важный параметр, как "точка актуальности" (TA). В поле FLAGS хранятся флаги текущего состояния системы.

#==TABLE no 1 : Системная

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SSYSTEM	Системная	A	1SSYSTEM	1

#-----Fields-----

# Name	Descr	Type	Length	Precision
F=CURDATE	Date of TM	D	8	0
F=CURTIME	Time of TM	C	6	0
F=EVENTIDTA	ID Event On TA	C	9	0
F=DBSIGN	DB Sign	C	3	0
F=DBSETUUID	UUID of DB set	C	36	0
F=SNAPSHPER	Snap Shot Period	C	1	0
F=ACCDATE	Date of Account Total	D	8	0
F=FLAGS	FLAGS	N	10	0

Структура файла 1sconst.dbf

В файле 1sconst.dbf хранятся значения констант и значения периодических реквизитов документов, поэтому структура файла достаточно сложная.

#==TABLE no 2 : Константы

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SCONST	Константы	A	1SCONST	1

#-----Fields-----

# Name	Descr	Type	Length	Precision
F=OBJID	ID obj(0-cons)	C	9	0
F=ID	ID parameter	C	4	0
F=DATE	Fix date	D	8	0
F=PARTNO	no of part	N	3	0

F=VALUE	valume	C	23	0
F=DOCID	ID Document	C	9	0
F=TIME	Time	C	6	0
F=ACTNO	Action No	N	6	0
F=LINENO	LineNo	N	4	0
F=TVALUE		C	3	0

#----Indexes-----

#	Name	Descr	Unique	Indexed fields	DBName
I=IDD			0	ID,OBJID,DATE,TIME,DOCID,PARTNO	IDD
I=DOC			0	DOCID,ACTNO,PARTNO	DOC

Назначение полей данной таблицы различное для констант и справочников. Для констант в поле ID хранится номер константы, как объекта метаданных, под которым он описан в файле 1Cv7.md. Поле OBJID используется в периодических константах для ведения истории значения константы. Для периодических реквизитов справочников в поле ID также хранится номер реквизита справочника, как объекта метаданных, под которым он описан в файле 1Cv7.md, но в поле OBJID хранится внутренний идентификатор элемента справочника. Если же значение реквизита было изменено документом, то в поле DOCID хранится внутренний идентификатор этого документа. Рассмотрим фрагмент файла, в котором хранится изменение истории значений периодических реквизитов справочника "Основные средства" с помощью документа "Ввод в эксплуатацию" и в диалоговом окне справочника (рис. 2.9).

OBJID	ID	DATE	PARTNO	VALUE	DOCID	TIME	ACTNO	LINENO	TVALUE
9	8TW	28.04.2003	0	2	1P3	65ZOMB	1	0	
9	9HT	28.04.2003	0	1	1P3	65ZOMB	2	0	
9	AQL	28.04.2003	0	0	1P3	65ZOMB	3	0	
9	AGN	28.04.2003	0	0,00	1P3	65ZOMB	4	0	
9	AQO	28.04.2003	0	0	1P3	65ZOMB	5	0	
9	8TX	28.04.2003	0	AQU	1P3	65ZOMB	6	0	
A	8TW	28.04.2003	0	2	1P7	661TSO	1	0	
A	9HT	28.04.2003	0	1	1P7	661TSO	2	0	
A	AQL	28.04.2003	0	0	1P7	661TSO	3	0	
A	AGN	28.04.2003	0	0,00	1P7	661TSO	4	0	
A	AQO	28.04.2003	0	0	1P7	661TSO	5	0	
A	8TX	28.04.2003	0	AQU	1P7	661TSO	6	0	
B	8TW	20.06.2003	0	1	1Q6	BTVSGO	1	0	
B	9HT	20.06.2003	0	1	1Q6	BTVSGO	2	0	
B	AQL	20.06.2003	0	0	1Q6	BTVSGO	3	0	
0	KIN		0	11111111111111	0	0	0	0	
9	8TW	28.04.2003	0	2	0	0	0	0	
9	9HT	28.04.2003	0	1	0	0	0	0	
9	AQL	28.04.2003	0	0	0	0	0	0	
A	8TW	01.01.1980	0	1	0	0	0	0	

Запись: 1163 из 1949

Рис. 2.9. Таблица 1SCONST

На рис. 2.9 видно, что значение реквизита VALUE = 1, заданное 01.01.1980, с ID = 8TW ("Подразделение" в справочнике "Основные средства") элемента справочника с номером OBJID = A было изменено документом с DOCID = 1P7 28.04.2003 на значение VALUE = 2.

Структура файла 1sjourn.dbf

Все журналы хранятся в файле 1sjourn.dbf. Рассмотрим фрагмент словаря базы данных, описывающий структуру данного файла.

#==TABLE no 3 : Журналы

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SJOURN	Журналы	A	1SJOURN	1

#-----Fields-----

# Name	Descr	Type	Length	Precision
F=IDJOURNAL	ID of Journal	C	4	0
F=IDDOC	ID Document	C	9	0
F=IDDOCDEF	ID Def Document	C	4	0
F=APPCODE	Application code	N	3	0
F=DATE	date	D	8	0
F=TIME	Time	C	6	0
F=DNPREFIX	Prefix Document No	C	18	0
F=DOCNO	Document No	C	10	0
F=CLOSED	Flag Document is Clo	N	1	0
F=ISMARK	Flag Document is Mar	C	1	0
F=ACTCNT	Action counter	C	6	0
F=VERSTAMP	Version stamp	C	6	0

#----Indexes-----

# Name	Descr	Uniq	Index fields	DBName
I=IDDOC	Id Doc	0	IDDOC	IDDOC
I=ACDATETIM	Date+Time+ID	0	DATE,TIME,IDDOC	ACDATETIME
I=DOCNO	Prefix+No	0	DNPREFIX,DOCNO(UPPER)	DOCNO
I=DOCTYPE	Type+Date+Time	0	IDDOCDEF,DATE,TIME,IDDOC	DOCTYPE
I=JOURNAL	Journal+Date+T	0	IDJOURNAL,DATE,TIME,IDDOC	JOURNAL

Необходимо напомнить, что все поля с именами, начинающимися с символов ID, содержат внутренние коды записей, что позволяет поддерживать целостность информации о ссылках.

Для однозначной идентификации документа в конкретном журнале служат поля:

- IDJOURNAL — идентификатор журнала;
- IDDOC — идентификатор документа;
- IDDOCDEF — идентификатор описания документа.

Поле APPCODE носит скорее служебный характер (рис. 2.10). Информация о том, как был создан документ — непосредственно в журнале, вводом на основании другого документа или обработкой — необходима для работы некоторых режимов. Значения этого поля могут быть полезны для всякого рода "разборок":

- 4 — документ был создан какой-либо обработкой;
- 20 — документ был создан непосредственно в журнале как автономный;
- 52 — документ был создан на основании другого документа.

IDJOURNAL	IDDOC	IDDOCDEF	APPCODE	DATE	TIME	DNPREFIX	DOCNO	CLOSE	ISMARK	ACTCNT	VERSTAMP
902	270	9NP	20	22.07.2003	9D8RWG	125172003	7-655	4		0	0
902	272	9NP	20	23.07.2003	9HK61C	125172003	7-656	4		0	0
902	27K	9NP	20	25.07.2003	98HY7K	125172003	7-659	4		0	0
902	2BP	9NP	20	28.07.2003	9H13LS	125172003	7-661	4		0	0
902	2BU	9NP	20	28.07.2003	BCPTYO	125172003	7-663	4		0	0
902	2C3	9NP	20	30.07.2003	AG6B5C	125172003	7-665	4		0	0
902	2C4	9NP	20	31.07.2003	72XDSW	125172003	7-666	4		0	0
902	27L	9NP	20	01.08.2003	5ON41S	125172003	8-133	4		0	0
902	2C5	9NP	20	01.08.2003	5UM38W	125172003	8-134	4		0	0
902	282	9NP	4	01.08.2003	759EHS	125172003	8-0001	0		0	0
902	283	9NP	4	01.08.2003	75BJNK	125172003	8-0002	0		0	0
902	284	9NP	4	01.08.2003	75DOTC	125172003	8-0003	0		0	0
902	285	9NP	4	01.08.2003	75FTZ4	125172003	8-0004	0		0	0
902	286	9NP	4	01.08.2003	75HZ4W	125172003	8-0005	0		0	0
902	287	9NP	4	01.08.2003	75K4AO	125172003	8-0006	0		0	0
902	288	9NP	4	01.08.2003	75M9GG	125172003	8-0007	0		0	0
902	289	9NP	4	01.08.2003	75OEM8	125172003	8-0008	0		0	0
902	28A	9NP	4	01.08.2003	75QJSD	125172003	8-0009	0		0	0
902	28B	9NP	4	01.08.2003	75SOXS	125172003	8-0010	0		0	0
902	28C	9NP	4	01.08.2003	75UU3K	125172003	8-0011	0		0	0
902	28D	9NP	4	01.08.2003	75WZ9C	125172003	8-0012	0		0	0
902	28E	9NP	4	01.08.2003	75Z4F4	125172003	8-0013	0		0	0
902	28F	9NP	4	01.08.2003	7619KW	125172003	8-0014	0		0	0
902	28G	9NP	4	01.08.2003	763EQO	125172003	8-0015	0		0	0
902	28H	9NP	20	01.08.2003	765JWG	125172003	8-0016	4		0	0
902	28I	9NP	4	01.08.2003	767P28	125172003	8-0017	0		0	0

Рис. 2.10. Таблица 1SJOURN

Структура файла 1scrdoc.dbf

Все документы, введенные на основании других документов, участвуют в так называемой структуре подчиненности. Для отражения полной структуры подчиненности документов предназначен файл 1scrdoc.dbf. Рассмотрим фрагмент словаря базы данных, описывающий структуру данного файла.

#==TABLE no 4 : Ссылки документов

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SCRDOC	Ссылки документов	A	1SCRDOC	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=MDID	Md ID of select	C	4	0
F=PARENTVAL	Parent Value	C	23	0
F=CHILDDATE	Child date	D	8	0
F=CHILDTIME	Child Time	C	6	0
F=CHILDID	Child Doc ID	C	9	0
F=FLAGS	Flags of refers	N	2	0

#----Indexes-----

# Name	Descr	Uniq	Indexed fields	DBName
I=CHILD	Child Referenc	0	CHILDID,MDID,PARENTVAL	CHILD
I=PARENT	Parent Referen	0	MDID,PARENTVAL,CHILDDATE,CHILDTIME,CHILDID	PARENT

Заданное подобным образом индексирование позволяет быстро восстанавливать схему подчиненности документа.

Структура файла 1sdnlock.dbf

Данный файл используется при работе с распределенной базой данных.

#==TABLE no 5 : Номеров документов

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SDNLOCK	Номеров документов	A	1SDNLOCK	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=DNPREFIX	Prefix object	C	28	0
F=DOCNO	Object No	C	10	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=DOCNO	Prefix+No	0	DNPREFIX,DOCNO	DOCNO

Файлы компоненты "Расчет"

Компонента "Расчет" предназначена для ведения учета объектов с большим количеством разнообразных операций, так называемых "расчетов". Поскольку виды расчетов могут задаваться пользователем, необходимы специфиче-

ские журналы расчетов. Журналы расчетов хранятся в отдельных файлах подобно документам и справочникам. Имена журналов состоят из префикса сj и номера объекта метаданных в файле конфигурации 1Cv7.md. Рассмотрим "Журнал расчета заработной платы". Описание журнала представлено в файле 1Cv7.md следующим фрагментом:

```
{ "CJ",
  {"447", "Зарплата", "Журнал расчета заработной платы", "Журнал зарплата",
  "16", "2", "2450419",
  {"CJParams",
  {"448", "Дни", "Отработанные дни", "Дни", "N", "6", "2", "0", "0", "0"},
  {"449", "Часы", "Отработанные часы", "Часы", "N", "7", "2", "0", "0", "0"},
  {"1089", "НомерСтрокиДокумента", "Номер строки документа", "Номер строки документа", "N", "4", "0", "0", "1", "0"},
  {"456", "СтрокаИсправления", "Номер строки документа-исправления", "Строка документа-исправления", "N", "3", "0", "0", "0", "0"}, "15", "2", "1",
  {"FF", "202"},
  {"Form",
  {"2703", "ФормаСписка", "", ""},
  {"2707", "ПоСотруднику", "Открывается из спр. Сотрудники", ""}, "2703", "2703"},
  {"757", "Дополнительный", "Расчет дополнительной заработной платы", "Журнал дополнительной зарплаты", "16", "2", "2450419",
  {"CJParams",
  {"752", "Дни", "Дни", "Дни", "N", "6", "2", "0", "0", "0"},
  {"753", "Часы", "Часы", "Часы", "N", "7", "2", "0", "0", "0"},
  {"1236", "НомерСтрокиДокумента", "Номер строки документа", "Номер строки документа", "N", "4", "0", "0", "0", "0"}, "15", "2", "0",
  {"FF", "202"},
  {"Form",
  {"755", "ФормаСписка", "", ""}, "755", "755"},
```

Соответствующее описание данного журнала в словаре базы данных представлено фрагментом:

```
#==TABLE по 123 : Журнал расчетов Зарплата
```

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=CJ447	Журнал расчетов Зарплата	A	CJ447	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=IDDOC		C	9	0
F=IDS		C	9	0
F=IDALG		C	4	0
F=ORDER		N	3	0
F=RESULT	Result	N	15	2
F=DATEB	date1	D	8	0
F=DATEE	date2	D	8	0
F=PERIOD		C	9	0
F=RECALC		N	3	0
F=ID		C	9	0
F=DP		N	1	0
F=IDPARDOC		C	9	0
F=IDRECALC		C	9	0
F=FF202	(P)ОсновнойЭлемент	C	9	0
F=SP448	(P)Дни	N	7	2
F=SP449	(P)Часы	N	8	2
F=SP1089	(P)НомерСтрокиДокуме	N	5	0
F=SP456	(P)СтрокаИсправления	N	4	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=IDDOC+PER		0	IDDOC,PERIOD,IDS,ORDER	IDDOC+PERIO
I=PERIOD+ID		0	PERIOD,IDS, ORDER,DATEB	PERIOD+IDS+
I=IDS+PERIO		0	IDS,PERIOD, ORDER,DATEB	IDS+PERIOD+
I=ID		0	ID	ID
I=IDS+DATEE		0	IDS,DATEE,ID	IDS+DATEE+I
I=DATEE+ID		0	DATEE,ID	DATEE+ID
I=IDPARDOC		0	IDPARDOC	IDPARDOC
I=IDRECALC		0	IDRECALC	IDRECALC
I=FF202	FF202	0	FF202,PERIOD,IDS, ORDER,DATEB	FF202

В представленном фрагменте особый интерес представляет индексация файла.

Файлы компоненты "Оперативный учет"

Компонента "Оперативный учет" предназначена для ведения учета накопительных объектов, так называемых "регистров". Регистры хранятся в отдельных файлах, только содержимое каждого регистра хранится в двух файлах. Остатки или обороты в зависимости от заданного в Конфигураторе вида хранятся в файлах, имена которых состоят из префикса rg и номера объекта метаданных. Расшифровки движений документов по регистру хранятся в файлах, имена которых состоят из префикса ga и номера объекта метаданных. Номера объектов метаданных находятся в файле конфигурации 1Cv7.md. Рассмотрим регистр Банк. Описание регистра представлено в файле 1Cv7.md следующим фрагментом:

```
{ "Registers",
  {"639", "Банк", "Остатки средств на банковских счетах", "", "0", " ", "0", "1",
  {"Props",
  {"4066", "Фирма", "", "", "В", "0", "0", "4014", "0", "0", "0", "0"}},
  {"641", "БанковскийСчет", "", "Банковский
счет", "В", "0", "0", "1710", "0", "0", "0", "0"}},
  {"Figures",
  {"644", "СуммаВал", "Остаток средств в валюте счета", "Сумма (вал.)", "N",
  "15", "2", "0", "0", "0"}},
  {"642", "СуммаУпр", "Остаток средств по упр. учету", "Сумма (упр.)", "N", "15",
  "2", "0", "0", "0"}},
  {"643", "СуммаРуб", "Сумма по бухгалтерскому учету", "Сумма (руб.)", "N",
  "15", "2", "0", "0", "0"}},
  {"Flds",
  {"686", "КодОперации", "", "Код операции", "E", "0", "0", "345", "0", "0", "0"}},
  {"4274", "ДвижениеДенежныхСредств", "", "Движение денежных
средств", "В", "0", "0", "4264", "0", "0", "0"}},
```

Соответствующее описание накопительной части данного регистра в словаре базы данных представлено фрагментом:

```
#==TABLE no 144 : Регистр Банк
```

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=RG639	Регистр Банк	A	RG639	1

```
#-----Fields-----
```

# Name	Descr	Type	Length	Precision
F=PERIOD	Period Registr	D	8	0

F=SP4066	(P)Фирма	C	9	0
F=SP641	(P)БанковскийСчет	C	9	0
F=SP644	(P)СуммаВал	N	16	2
F=SP642	(P)СуммаУпр	N	16	2
F=SP643	(P)СуммаРуб	N	16	2
#----Indexes-----				
# Name	Descr	Unique	Indexed fields	DBName
I=PROP	PERIOD+PROP	0	PERIOD,SP4066,SP641	PROP

Следует обратить особое внимание на то, что суммы в поле SP642 хранятся в валюте, назначенной для управленческого учета, в данном случае в USD (рис. 2.11).

PERIOD	SP4066	SP641	SP644	SP642	SP643
01.04.2002	1	4	35100	35100	1092283,92
01.04.2002	1	U	414843,28	13420,55	414843,28
01.04.2002	2	4	-2080	-2080	-64998,19
01.04.2002	2	2	75000	2480,03	75000
01.04.2002	2	U	2000	64,7	2000
01.05.2002	1	4	33900	33900	1054940,88
01.05.2002	1	U	352119,28	11391,37	352119,28
01.05.2002	2	2	-186440	-5976,27	-186440
01.05.2002	2	4	-2080	-2080	-64998,19
01.05.2002	2	U	2000	64,7	2000
01.06.2002	1	4	33900	33900	1054940,88
01.06.2002	1	U	352119,28	11391,37	352119,28
01.06.2002	2	2	-186440	-5976,27	-186440
01.06.2002	2	4	-2080	-2080	-64998,19

Рис. 2.11. Таблица RG639

Соответствующее описание расшифровки движений данного регистра в словаре базы данных представлено фрагментом:

#==TABLE по 145 : Регистр (Дв.) Банк

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=RA639	Регистр (Дв.) Банк	A	RA639	1
#-----Fields-----				

# Name	Descr	Type	Length	Precision
F=IDDOC	ID Document's	C	9	0
F=LINENO	LineNo	N	4	0
F=ACTNO	Action No	N	6	0
F=DEBKRED	Flag Debet/Kredit	N	1	0
F=SP4066	(P)Фирма	C	9	0

F=SP641	(P)БанковскийСчет	C	9	0
F=SP644	(P)СуммаВал	N	16	2
F=SP642	(P)СуммаУпр	N	16	2
F=SP643	(P)СуммаРуб	N	16	2
F=SP686	(P)КодОперации	C	9	0
F=SP4274	(P)ДвижениеДенежныхС	C	9	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=IDLINE	of IDDOC+LineN	0	IDDOC,LINENO,ACTNO	IDLINE

Из приведенного на рис. 2.11 фрагмента таблицы RG639 видно, что остаток в USD 35100 сложился из дебетового оборота 36000 (DEBKRED = 0) минус кредитовый оборот 900 (DEBKRED = 1) фрагмента таблицы RA639 (рис. 2.12).

RA639 : таблица										
IDDOC	LINENO	ACTNO	DEBKRED	SP4066	SP641	SP644	SP642	SP643	SP686	SP4274
31	0	5	0	1	4	36000	36000	1120291,2	2XZ	0
31	0	6	0	1	U	150000	5000	150000	2XY	0
32	0	65	0	2	4	300	300	9335,76	2XZ	0
32	0	66	0	2	2	100000	3333,33	100000	2XY	0
6E	0	36	0	1	U	467566,5	15000	467566,5	1PS	H
54	0	68	0	2	2	4000	128,32	4000	1MN	2
55	0	107	0	2	4	0	0	15,57	1G7	8
55	0	108	1	2	4	10000	10000	311711	3S1	8
1W	0	107	0	1	U	0	-19,89	0	1G7	2
1W	0	108	1	1	U	65400	2098,1	65400	1ML	2
3Q	0	48	0	2	2	23000	737,86	23000	1MN	9
53	0	132	0	1	U	0	-41,36	0	1G7	8
5M	0	72	0	1	U	52881,4	1696,49	52881,4	1MN	2
28	0	146	0	1	U	0	-5,25	0	1G7	0
28	0	147	1	1	U	21000	674,12	21000	1ML	0
56	0	104	0	2	2	0	-50,22	0	1G7	8
56	0	105	1	2	2	52000	1669,26	52000	1ML	8
5Z	1	68	0	2	4	7620	7620	237361,48	3S0	2
5Q	1	151	0	1	U	0	-8	0	1G7	8
5Q	1	152	1	1	U	31245,63	1002,82	31245,63	1ML	8
22	1	101	0	1	4	0	0	70,65	1G7	0
22	1	102	1	1	4	900	900	28077,93	3S1	0

Запись: 24 из 33

Рис. 2.12. Таблица RA639

Структура файла 1sstream.dbf

Для компоненты "Оперативный учет" характерно ведение последовательности документов. Последовательности ведутся в файле 1sstream.dbf. Описание последовательностей представлено в файле 1Cv7.md следующим фрагментом:

```

{"Document Streams",
{"1946","ОсновнаяПоследовательность","","Основная последовательность",
"0","0",
{"Registers","4314","4335","328","351","635","639","2964","405","438","464",
"4667","4674","4480"}},
{"Documents","1582","1611","1656","1684","1774","1790","2196","2225",
"3259","2320","2075","3274","3311","1628","3114","2988","2998","3089",
"2827","3638","5211","4913","5292","3614","4854","4847","3995","4694",
"3592","2695","2051","4389","3504","5155","2742","2457","6661","4132",
"6313","6532","6322"}},
{"4757","КнигаПокупок","","Книга покупок","0","0",
{"Registers","3549"}},
{"Documents","3592","3504","6661"}},
{"5722","КнигаПродаж","","Книга продаж","0","0",
{"Registers","4343"}},
{"Documents","4389","4694","6661"}},

```

Во фрагменте приведено описание трех последовательностей: ОсновнаяПоследовательность, КнигаПокупок, КнигаПродаж.

Соответствующее описание таблицы последовательностей в словаре базы данных представлено фрагментом:

#==TABLE по 176 : Последовательность документов

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SSTREAM	Последовательность документов	A	1SSTREAM	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=ID	Doc Stream ID	C	4	0
F=DATE	Date	D	8	0
F=TIME	Time	C	6	0
F=DOCID	Doc ID	C	9	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=ID	by doc stream	0	ID	ID

Файлы компоненты "Бухгалтерский учет"

Описание плана счетов, операций, проводок, бухгалтерских итогов, остатков, отбора счетов, корректных проводок, типовых операций, отбора проводок по субконто представлены в конце словаря базы данных.

Структура файла 1saccs.dbf

Один из самых важных файлов бухгалтерии — "Счета". Другими словами, в данном файле хранится план счетов.

#==TABLE no 191 : Счета

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SACCS	Счета	A	1SACCS	1

#-----Fields-----

# Name	Descr	Type	Length	Precision
F=ID	Account Id	C	9	0
F=PLANID	Plan Id	C	4	0
F=SCHKOD	Account code(number)	C	11	0
F=SCHIM	Account description	C	25	0
F=SCHV	Flag Currency enable	N	1	0
F=SCHKOL	Flag Amount enable	N	1	0
F=SCHSINGLE	Flag no need corresp	N	1	0
F=ISFOLDER	Flag Have child acco	N	1	0
F=ISMARK	Flag Object is Marke	C	1	0
F=LEVEL	Level of Account	N	3	0
F=MDID	Metadata Id	C	4	0
F=ACTIVE	Flag Active	N	1	0
F=VERSTAMP	Version stamp	C	6	0
F=SC0		C	4	0
F=OSC0		N	1	0
F=FSC0		N	1	0
F=SC1		C	4	0
F=OSC1		N	1	0
F=FSC1		N	1	0
F=SC2		C	4	0
F=OSC2		N	1	0
F=FSC2		N	1	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=ID1	of ID	0	ID	ID1
I=CODE	of CODE	0	PLANID,SCHKOD	CODE
I=SUBACC	of SubAcc	0	LEVEL,PLANID,SCHKOD	SUBACC

Назначение полей следующее:

- PLANID — номер плана счетов, под которым он описан в файле 1Cv7.md;
- ID — "сквозной" внутренний номер счета, независимо от принадлежности к конкретному плану счетов;
- MDID — номер счета, под которым он описан в файле 1Cv7.md;
- SCHKOD — код счета в справочнике;
- SCHIM — наименование счета в справочнике;
- SCHV — флаг валютного учета счета;
- SCHKOL — флаг количественного учета счета;
- SCHSINGLE — флаг принадлежности счета к балансовым;
- ISFOLDER — флаг папки;
- ISMARK — флаг пометки на удаление;
- LEVEL — уровень счета (для субсчетов);
- ACTIVE — флаг вида счета (активный, пассивный или активно-пассивный);
- SC0 — номер вида субконто1, которым он описан в файле 1Cv7.md;
- OSC0 — флаг учета оборотов по счету для субконто1;
- FSC0 — флаг суммового и количественного учета по субконто1;
- SC1 — номер вида субконто2, которым он описан в файле 1Cv7.md;
- OSC1 — флаг учета оборотов по счету для субконто2;
- FSC1 — флаг суммового и количественного учета по субконто2;
- SC2 — номер вида субконто3, которым он описан в файле 1Cv7.md;
- OSC2 — флаг учета оборотов по счету для субконто3;
- FSC2 — флаг суммового и количественного учета по субконто3.

Внимание!

Поскольку тип данных в полях внутренних идентификаторов объявлен "символьный", отображение в стандартных программах просмотра таблиц не со-

всем понятно (рис. 2.13). Для бухгалтера значение номера в поле MDID для счета "01" 11748, хотя в программе просмотра таблицы – 910.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
1	ID	PLANID	SCHKOD	SCHIM	SCHV	SCHKOL	SCHSINGL	ISFOLD	ISMARI	LEVEL	MDID	ACTV	VERST	SCO	OSCF	SCO	SCI	OSC1	FSC1
2	1	MH	00.	Вспомогательный	0	0	0	0	0	192C	3	0	0	0	7	0	0	7	
3	2	MH	01.	Основные средства	0	0	0	1	0	191O	1	0	8XZ	0	7	0	0	7	
4	3	MH	01. 1.	ОС в организации	0	0	0	0	0	291Q	1	0	8XZ	0	7	0	0	7	
5	4	MH	01. 2.	Выбытие ОС	0	0	0	0	0	291R	1	0	8XZ	0	7	0	0	7	
6	5	MH	02.	Амортизация ОС	0	0	0	1	0	191S	2	0	8XZ	0	7	0	0	7	
7	6	MH	02. 1.	Аморт. ОС, уч. на сч	0	0	0	0	0	291U	2	0	8XZ	0	7	0	0	7	
8	7	MH	02. 2.	Аморт. ОС, уч. на сч	0	0	0	0	0	291V	2	0	8XZ	0	7	0	0	7	
9	8	MH	03.	Доходные вложения	0	0	0	1	0	191W	1	0	8XZ	0	7	0	0	7	
10	9	MH	03. 1.	МЦ в организации	0	0	0	0	0	291Y	1	0	8XZ	0	7	0	0	7	
11	A	MH	03. 2.	МЦ, перед. во врем.	0	0	0	0	0	291Z	1	0	8XZ	0	7	EE	0	7	
12	B	MH	03. 3.	МЦ, перед. во врем.	0	0	0	0	0	2920	1	0	8XZ	0	7	EE	0	7	
13	C	MH	03. 4.	Прочие доходные вл	0	0	0	0	0	2921	1	0	8XZ	0	7	EE	0	7	
14	D	MH	03. 5.	Выбытие МЦ	0	0	0	0	0	2922	1	0	8XZ	0	7	0	0	7	
15	E	MH	04.	Нематериальные акт	0	0	0	1	0	1928	1	0	8XW	0	7	0	0	7	
16	F	MH	05.	Амортизация НМА	0	0	0	0	0	192A	2	0	8XW	0	7	0	0	7	
17	6E	MH	07.	Оборудование к уста	0	1	0	0	0	1A1B	1	0	8XX	0	7	EH	0	6	
18	G	MH	08.	Влож во внеоборотн.	0	0	0	1	0	192D	1	0	0	0	7	0	0	7	
19	H	MH	08. 1.	Приобретение зем.уч.	0	0	0	0	0	292E	1	0	8XY	0	7	0	0	7	
20	I	MH	08. 2.	Приобр. об. природо	0	0	0	0	0	292G	1	0	8XY	0	7	0	0	7	
21	J	MH	08. 3.	Строительство объек	0	0	0	0	0	292I	1	0	8XY	0	7	91H	1	7	
22	K	MH	08. 4.	Приобр. отд. объекто	0	1	0	0	0	2955	1	0	8XY	0	7	0	0	7	
23	L	MH	08. 5.	Приобретение НМА	0	0	0	0	0	2957	1	0	8XY	0	7	0	0	7	

Рис. 2.13. Таблица 1SACCS.DBF

Структура файла 1soper.dbf

Все операции, сформированные документами, фиксируются в таблице операций 1SOPER. Если документом были сформированы проводки, они фиксируются еще и в таблице 1SENTRY. Потеря записи в отдельно взятой таблице неминуемо приведет к нарушению целостности базы данных. В программе существует ряд специальных режимов по восстановлению целостности базы данных, к рассмотрению которых мы вернемся в главе 6.

#===TABLE no 192 : Операции

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SOPER	Операции	A	1SOPER	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=DOCID	Document Id	C	9	0
F=DATE	date	D	8	0
F=TIME	Time	C	6	0
F=DESCR	Oper description	C	50	0
F=SUM	Oper sum	N	14	2
F=ACTIVE		C	1	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=DOCID1	of ID	0	DOCID	DOCID1
I=DATETIME	ofDate Time	0	DATE, TIME, DOCID	DATETIME
I=SumIdx	of SUM Date Ti	0	SUM, DATE, TIME, DOCID	SumIdx
I=DescrIdx	ofSUMDateTi	0	DESCR(UPPER), DATE, TIME, DOCID	DescrIdx

На рис. 2.14 видно, что журнал операций — данная таблица с индексами.

1	2	3	4	5	6
DOCID	DATE	TIME	DESCR	SUM	ACTIVE
1					
2	1	31.05.2002	7579CD	Начисл. Зарплата за май 2002 у.	8000,00
3	2	31.05.2002	759EHS		0,00
4	3	06.06.2002	7579CD	Поступление в кассу: Договор займа	50000,00
5	4	06.06.2002	759EHS	Сдача наличных в банк: Выдача из кассы	50000,00
6	5	18.06.2002	7579CD	Поступл. в кассу: на выдачу зарплаты за май 6957=, хоз.н	7320,00
7	6	18.06.2002	759EHS	Выдача из кассы:	6957,00
8	7	18.06.2002	75BUNK	Выдача из кассы под отчет	360,00
9	8	17.05.2002	7579CD	Авенсовый отчет:	360,00
10	9	13.06.2002	7579CD	Движения по р/с	3988,00
11	A	18.06.2002	75DOTC	Платежное поручение	4000,00
12	B	18.06.2002	75FTZ4	Движения по р/с	4005,82
13	4K	30.09.2002	76OUCG	Оказание услуг	1134,00
14	D	02.07.2002	7579CD	Счет заказчику: Модус-Авто	10206,00
15	E	31.07.2002	7579CD	Оказание услуг	1701,00
16	F	31.07.2002	759EHS	Счет-фактура покупателю: Модус-Авто	1701,00
17	G	10.07.2002	9ADONC	Поступл. В кассу: На выплату зарплаты за июнь месяц	12171,00
18	H	10.07.2002	9BPOVK	Выдача из кассы: Зарплата за июнь	6957,00
19	I	10.07.2002	9CJWU8	Выдача из кассы: Зарплата за июнь	2607,00
20	J	10.07.2002	9DM2AO	Выдача из кассы: Зарплата за июнь	2607,00
21	K	05.07.2002	7579CD	Движения по р/с	10206,00

Рис. 2.14. Таблица 1SOPER.DBF

Назначение полей:

- DOCID — внутренний идентификатор документа или ручной операции;
- DATE — дата документа или ручной операции;
- TIME — время документа или ручной операции;
- DESCR — атрибут объекта метаданных Операция "Содержание" с добавленной расшифровкой. Атрибут операции формируется при выполнении predeterminedенной процедуры модуля формы документа ПриЗаписи ();
- SUM — атрибут объекта метаданных Операция "СуммаОперации". Атрибут операции формируется при выполнении predeterminedенной процедуры модуля формы документа ПриЗаписи ();
- ACTIVE — признак активности записи.

Структура файла 1sentry.dbf

Все проводки, сформированные документами, фиксируются в таблице **1SENTRY**. Индексированная таблица **1SENTRY** — это журнал проводок.

#==TABLE no 193 : Проводки

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SENTRY	Проводки	A	1SENTRY	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=DOCID	Document Id	C	9	0
F=NUMBER	Prov number	N	5	0
F=CORNO	Correspond number	N	5	0
F=DATE	date	D	8	0
F=TIME	Time	C	6	0
F=ACCDTID	AccountDt Id	C	9	0
F=ACCKTID	AccountKt Id	C	9	0
F=SUM	Prov sum	N	14	2
F=CURRID	Currency Id	C	9	0
F=CURSUM	Prov currency sum	N	14	2
F=AMOUNT	Prov amount	N	14	3
F=PROVKIND		C	1	0
F=ACTIVE		C	1	0
F=DTFLAGS		N	1	0
F=KTFLAGS		N	1	0
F=DOCLINENO	Document line number	N	4	0
F=SP547	(P)СодержаниеПроводк	C	50	0
F=SP548	(P)НомерЖурнала	C	2	0
F=SP26962	(P)ПервичныйДокумент	C	25	0
F=VDTSC0		C	4	0
F=DTSC0		C	13	0
F=ODTSC0		N	1	0
F=VDTSC1		C	4	0
F=DTSC1		C	13	0
F=ODTSC1		N	1	0
F=VDTSC2		C	4	0
F=DTSC2		C	13	0
F=ODTSC2		N	1	0

F=VKTSC0	C	4	0
F=KTSC0	C	13	0
F=OKTSC0	N	1	0
F=VKTSC1	C	4	0
F=KTSC1	C	13	0
F=OKTSC1	N	1	0
F=VKTSC2	C	4	0
F=KTSC2	C	13	0
F=OKTSC2	N	1	0

#----Indexes-----

#	Name	Descr	Uniq	Indexed fields	DBName
I=	DOCIDNUMB	of ID	0	DOCID,NUMBER,CORNO	DOCIDNUMBER
I=	DATETIME	of DateTime	0	DATE,TIME,DOCID,NUMBER,CORNO	DATETIME
I=	SumIdx	of SUM	0	SUM,DATE,TIME,DOCID,NUMBER,CORNO	SumIdx
I=	VIA548	VIA548	0	SP548(UPPER=128),DATE,TIME,DOCID,NUMBER,CORNO	VIA548

Назначение полей:

- DOCID — внутренний идентификатор документа или ручной операции;
- DATE — дата документа или ручной операции;
- TIME — время документа или ручной операции;
- NUMBER — порядковый номер проводки, начиная с 0;
- ACCDTID — идентификатор счета дебета, совпадает со значением поля ID в таблице **1SACCS**;
- ACCKTID — идентификатор счета кредита, совпадает со значением поля ID в таблице **1SACCS**;
- SUM — сумма проводки;
- CURRID — идентификатор валюты;
- CURSUM — сумма в валюте;
- AMOUNT — количество;
- PROVkind — вид проводки;
- ACTIVE — признак активности записи;
- DTFLAGS — флаги дебета (количественный учет счета, валютный и т. п.);
- KTFLAGS — флаги кредита (количественный учет счета, валютный и т. п.);

- DOCLINENO — номер строки документа;
- SP547 — атрибут СодержаниеПроводки;
- SP548 — номер журнала;
- SP26962 — данные первичного документа, графа журнала проводок "Первичный документ";
- VDTSC0 — номер вида субконто1 дебета, которым он описан в файле 1Cv7.md;
- ODTSC0 — флаг учета оборотов по счету для субконто1 дебета;
- DTSC0 — идентификатор субконто1 дебета;
- VKTSC0 — номер вида субконто1 кредита, которым он описан в файле 1Cv7.md;
- OKTSC0 — флаг учета оборотов по счету для субконто1 кредита;
- KTSC0 — идентификатор субконто1 кредита;
- VDTSC1 — номер вида субконто2 дебета, которым он описан в файле 1Cv7.md;
- ODTSC1 — флаг учета оборотов по счету для субконто2 дебета;
- DTSC1 — идентификатор субконто2 дебета;
- VKTSC1 — номер вида субконто2 кредита, которым он описан в файле 1Cv7.md;
- OKTSC1 — флаг учета оборотов по счету для субконто2 кредита;
- KTSC1 — идентификатор субконто2 кредита;
- VDTSC2 — номер вида субконто3 дебета, которым он описан в файле 1Cv7.md;
- ODTSC2 — флаг учета оборотов по счету для субконто3 дебета;
- DTSC2 — идентификатор субконто3 дебета;
- VKTSC2 — номер вида субконто3 кредита, которым он описан в файле 1Cv7.md;
- OKTSC2 — флаг учета оборотов по счету для субконто3 кредита;
- KTSC2 — идентификатор субконто3 кредита.

Структура файла 1sbkttlc.dbf

Для получения оперативных отчетов программой ведется таблица бухгалтерских итогов 1SBKTTLC (рис. 2.15). При работе с некоторыми методами

1	2	3	4	5	6	7	8	9	10
DATE	ACCDTID	ACCKTID	CURRID	KIND	OB1	OB2	OB3	DTFLAGS	KTFLAGS
01.04.2002	K	43	0	1	0.00	5200.00	0.00		2
01.04.2002	43	4L	0	1	0.00	10000.00	0.00		0
01.04.2002	3	K	0	1	0.00	5200.00	0.00		2
01.04.2002	1C	3W	0	1	0.00	360.00	0.00		0
01.04.2002	1C	3U	0	1	0.00	8000.00	14000.00		0
01.04.2002	3U	3H	0	1	0.00	1040.00	1820.00		0
01.01.2003	Z1422B	Z1422D	0	1	530.16	1111.26	1368.18		0
01.04.2002	1C	3Q	0	1	0.00	320.00	560.00		0
01.01.2003	Z1422B	Z1422C	0	1	2433.92	4710.64	5955.36		0
01.04.2002	1C	3T	0	1	0.00	16.00	28.00		0
01.04.2002	1C	Z1422B	0	1	0.00	1120.00	1960.00		0
01.04.2002	1C	Z1422C	0	1	0.00	880.00	1630.00		0
01.04.2002	1C	Z1422D	0	1	0.00	240.00	330.00		0
01.07.2003	3W	1W	0	1	0.00	5200.00	0.00		0
01.07.2003	1D	Z1422B	0	1	8071.28	0.00	0.00		0
01.04.2002	1W	4B	0	1	0.00	0.00	50000.00		0
01.04.2002	1Z	1W	0	1	0.00	0.00	50000.00		0
01.04.2003	19	3W	0	1	2290.00	2340.00	2140.00		0
01.04.2003	3I	4F	0	1	0.00	0.00	3549.50		0
01.04.2003	4F	3I	0	1	0.00	0.00	14057.00		0
01.04.2002	5A	1Z	0	1	0.00	0.00	15.82		0
01.07.2003	Z1422B	Z1422D	0	1	1906.54	0.00	0.00		0

Рис. 2.15. Таблица 1SBKTTLC.DBF

объекта метаданных "БухгалтерскиеИтоги" обращение производится к данной таблице. Нарушение в данной таблице приводит к неприятностям при получении ряда отчетов, таких как "Оборотно-сальдовая ведомость", "Оборотно-сальдовая ведомость по счету" и т. п. В программе существует специальный режим по восстановлению бухгалтерских итогов, к рассмотрению которого мы вернемся в *главе 6*.

```
#==TABLE no 194 : Итоги
```

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SBKTTLC	Итоги	A	1SBKTTLC	1

```
#-----Fields-----
```

# Name	Descr	Type	Length	Precision
F=DATE	Period	D	8	0
F=ACCDTID	AccountDt Id	C	9	0
F=ACCKTID	AccountKt Id	C	9	0
F=CURRID	Currency Id	C	9	0
F=KIND	Total kind	C	1	0
F=OB1	Total turnover	C	15	0
F=OB2	Total turnover	C	15	0
F=OB3	Total turnover	C	15	0
F=DTFLAGS		N	1	0
F=KTFLAGS		N	1	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=ACCDT1		0	DATE,KIND,ACCDTID,ACCKTID,CURRID	ACCDT1
I=ACCKT1		0	DATE,KIND,ACCKTID,ACCDTID,CURRID	ACCKT1

Назначение полей:

- DATE — период итогов, начало отчетного квартала;
- ACCDTID — идентификатор счета дебета, совпадает со значением поля ID в таблице **1SACCS**;
- ACCKTID — идентификатор счета кредита, совпадает со значением поля ID в таблице **1SACCS**;
- CURRID — валюта итогов;
- KIND — вид итогов;
- OB1 — дебетовый оборот;
- OB2 — кредитовый оборот;
- OB3 — оборот между счетами;
- DTFLAGS — флаги счета дебета;
- KTFLAGS — флаги счета кредита.

Структура файла 1sbkttl.dbf

Для получения оперативных отчетов по субконто программой ведется таблица бухгалтерских итогов **1SBKTTL**. При работе с некоторыми методами объекта метаданных "БухгалтерскиеИтоги" для получения итогов по субконто обращение производится к данной таблице.

#==TABLE no 195 : Остатки

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SBKTTL	Остатки	A	1SBKTTL	1

#-----Fields-----

# Name	Descr	Type	Length	Precision
F=DATE	Period	D	8	0
F=ACCID	AccountDt Id	C	9	0
F=CURRID	Currency Id	C	9	0
F=KIND	Total kind	C	1	0
F=OBDT1	Total turnover DT	C	15	0
F=OBKT1	Total turnover KT	C	15	0

F=OBBDT2	Total turnover DT	C	15	0
F=OBBDT2	Total turnover KT	C	15	0
F=OBBDT3	Total turnover DT	C	15	0
F=OBBDT3	Total turnover KT	C	15	0
F=SD	Saldo	C	15	0
F=FLAGS		N	1	0
F=VSC0		C	4	0
F=SC0		C	13	0
F=OSC0		N	1	0
F=VSC1		C	4	0
F=SC1		C	13	0
F=OSC1		N	1	0
F=VSC2		C	4	0
F=SC2		C	13	0
F=OSC2		N	1	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=ACC1		0	DATE,KIND,ACCID,SC0,SC1,SC2,CURRID	ACC1

Назначение полей:

- DATE — период итогов, начало отчетного квартала;
- ACCDTID — идентификатор счета дебета, совпадает со значением поля ID в таблице **1SACCS**;
- ACCKTID — идентификатор счета кредита, совпадает со значением поля ID в таблице **1SACCS**;
- CURRID — валюта итогов;
- KIND — вид итогов;
- OBBDT1 — обороты по дебету субконто1;
- OBBDT1 — обороты по кредиту субконто1;
- OBBDT2 — обороты по дебету субконто2;
- OBBDT2 — обороты по кредиту субконто2;
- OBBDT3 — обороты по дебету субконто3;
- OBBDT3 — обороты по кредиту субконто3;
- SD — сальдо;
- FLAGS — флаги;

- VSC0 — номер вида субконто1;
- SC0 — идентификатор субконто1;
- OSC0 — признак учета только оборотов субконто1;
- VSC1 — номер вида субконто2;
- SC1 — идентификатор субконто2;
- OSC1 — признак учета только оборотов субконто2;
- VSC2 — номер вида субконто3;
- SC2 — идентификатор субконто3;
- OSC2 — признак учета только оборотов субконто2.

Структура файла 1scorent.dbf

Перечень корректных проводок хранится в таблице **1SCORENT** (рис. 2.16).

#==TABLE по 196 : Корректные проводки

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SCORENT	Корректные проводки	A	1SCORENT	1

#-----Fields-----

# Name	Descr	Type	Length	Precision
F=ACCDT	Account Dt Id	C	9	0
F=ACCKT	Account Kt Id	C	9	0
F=DESCR	Description	C	50	0
F=DTCODE	Account Dt Code	C	11	0
F=KTCODE	Account Kt Code	C	11	0
F=PLANID	Account Kt Code	C	11	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=DT		0	ACCDT,ACCKT	DT
I=KT		0	ACCKT,ACCDT	KT
I=CODEDT		0	PLANID,DTCODE,KTCODE	CODEDT
I=CODEKT		0	PLANID,KTCODE,DTCODE	CODEKT

Индексы существуют по всем полям. Это позволяет осуществлять оперативный контроль вводимых проводок в соответствии с представленной таблицей.

1	2	3	4	5	6
ACCDT	ACCKT	DESCR	DTCCODE	KTCCODE	PLANID
3	H		01. 1	08. 1	MH
3	I		01. 1	08. 2	MH
3	J		01. 1	08. 3	MH
3	K		01. 1	08. 4	MH
3	49		01. 1	76. 3	MH
4	3		01. 2	01. 1	MH
5	4		02	01. 2	MH
8	1		03	00	MH
E	1		04	00	MH
E	L		04	08. 5	MH
E	49		04	76. 3	MH
F	E		05	04	MH
BE	1		07	00	MH
G	1		08	00	MH
K	2J		08. 4	60. 1	MH
L	2J		08. 5	60. 1	MH
M	1		10	00	MH
M	2J		10	60. 1	MH
N	49		10. 1	76. 3	MH
V	3W		10. 9	71. 1	MH
V	3X		10. 9	71.11	MH
W	1		14	00	MH

Рис. 2.16. Таблица 1SCORENT.DBF

Назначение полей:

- ACCDT — идентификатор счета дебета, совпадает со значением поля ID в таблице 1SACCS;
- ACCKT — идентификатор счета кредита, совпадает со значением поля ID в таблице 1SACCS;
- DESCR — содержание проводки;
- DTCCODE — код счета дебета в плане счетов;
- KTCCODE — код счета кредита в плане счетов;
- PLANID — номер счета, под которым он описан в файле 1Cv7.md.

Структура файла 1sacssel.dbf

#==TABLE no 197 : Отбор Счетов (рис. 2.17)

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SACCSEL	Отбор Счетов	A	1SACCSEL	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=ACCID	Account Id	C	9	0
F=DATE	Date	D	8	0
F=TIME	Time	C	6	0

F=DOCID	Doc Id	C	9	0
F=NUMBER	Prov number	N	5	0
F=CORNO	Correspond number	N	5	0
F=DT		C	1	0
F=KT		C	1	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=MAIN		0	ACCID,DATE,TIME,DOCID, NUMBER,CORNO	MAIN
I=PROV		0	DOCID,NUMBER,CORNO	PROV

1	2	3	4	5	6	7	8	
1	ACCID	DATE	TIME	DOCID	NUMBER	CORNO	DT	KT
2	K	06.05.2002	7579C0	N	0	0	*	
3	G	06.05.2002	7579C0	N	0	0	*	
4	43	06.05.2002	7579C0	N	0	0		*
5	42	06.05.2002	7579C0	N	0	0		*
6	K	06.05.2002	7579C0	N	1	0	*	
7	G	06.05.2002	7579C0	N	1	0	*	
8	43	06.05.2002	7579C0	N	1	0		*
9	42	06.05.2002	7579C0	N	1	0		*
10	43	06.05.2002	7579C0	N	2	0	*	
11	42	06.05.2002	7579C0	N	2	0	*	
12	4L	06.05.2002	7579C0	N	2	0		*
13	43	06.05.2002	7579C0	N	3	0	*	
14	42	06.05.2002	7579C0	N	3	0	*	
15	4L	06.05.2002	7579C0	N	3	0		*
16	3	06.05.2002	7579C0	N	4	0	*	
17	2	06.05.2002	7579C0	N	4	0	*	
18	K	06.05.2002	7579C0	N	4	0		*
19	G	06.05.2002	7579C0	N	4	0		*
20	3	06.05.2002	7579C0	N	5	0	*	
21	2	06.05.2002	7579C0	N	5	0	*	
22	K	06.05.2002	7579C0	N	5	0		*
23	G	06.05.2002	7579C0	N	5	0		*
24	1C	17.05.2002	7579C0	8	0	0	*	
25	3W	17.05.2002	7579C0	8	0	0		*
26	3V	17.05.2002	7579C0	8	0	0		*

Рис. 2.17. Таблица 1SACCSEL.DBF

Структура файла 1stoper.dbf

#===TABLE no 198 : Типовые операции (рис. 2.18)

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1STOPER	Типовые операции	A	1STOPER	1

#-----Fields-----

# Name	Descr	Type	Length	Precision
F=ID	Typ Oper ID	C	9	0

F=PARENTID	ID parent obj	C	9	0
F=CODE	Typ oper Description	C	80	0
F=ISFOLDER	Flag - Is Line – Fol	N	1	0
#----Indexes-----				
# Name	Descr	Unique	Indexed fields	DBName
I=IDD	of ID	0	ID	IDD
I=PCODE	ofPARENTand	0	PARENTID,ISFOLDER, CODE(UPPER)	PCODE
I=CODE	of CODE	0	CODE(UPPER)	CODE

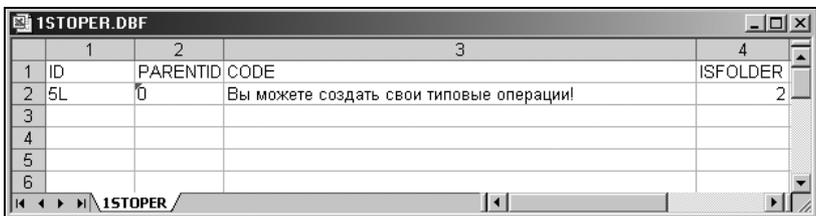


Рис. 2.18. Таблица 1STOPER.DBF

Структура файла 1ssbsel.dbf

#==TABLE no 199 : Отбор проводок по субконто (рис. 2.19)

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=1SSBSEL	Отбор проводок по субконто	A	1SSBSEL	1

#----Fields-----

# Name	Descr	Type	Length	Precision
F=SBKINDID	Sb Kind ID of select	C	4	0
F=VAL	Value	C	13	0
F=DATE	Date	D	8	0
F=TIME	Time	C	6	0
F=DOCID	Doc ID	C	9	0
F=NUMBER	Prov number	N	5	0
F=CORNO	Correspond number	N	5	0

#----Indexes-----

# Name	Descr	Uniq	Indexed fields	DBName
I=ENTRY	all entries re	0	DOCID,NUMBER,CORNO	ENTRY
I=VALUE	entrybyvalue	0	SBKINDID,VAL,DATE,TIME, DOCID,NUMBER,CORNO	VALUE

1	2	3	4	5	6	7	
SBKINDID	VAL	DATE	TIME	DOCID	NUMBER	CORNO	
2	EE	B	06.05.2002	7579C0	N	0	0
3	EE	B	06.05.2002	7579C0	N	1	0
4	EE	2	06.05.2002	7579C0	N	2	0
5	EE	B	06.05.2002	7579C0	N	3	0
6	EE	2	06.05.2002	7579C0	3	0	0
7	EE	6	18.06.2002	75FTZ4	B	1	0
8	EE	9	05.07.2002	7579C0	K	0	0
9	EE	6	25.07.2002	7579C0	1H	0	0
10	EE	6	25.07.2002	7579C0	1H	1	0
11	EE	6	25.07.2002	7579C0	1H	2	0
12	EE	6	25.07.2002	759EHS	1I	0	0
13	EE	C	30.07.2002	75BJNK	19	0	0
14	EE	C	30.07.2002	75DOTC	1A	0	0
15	EE	C	30.07.2002	75DOTC	1A	1	0
16	EE	9	31.07.2002	75DOTC	1B	0	0
17	EE	9	31.07.2002	75DOTC	1B	1	0
18	EE	9	31.07.2002	75FTZ4	1C	0	0

Рис. 2.19. Таблица 1SSBSEL.DBF

Файлы компоненты "Расчет"

Компонента "Расчет" предназначена для ведения учета объектов с большим количеством разнообразных операций, так называемых "расчетов". Поскольку виды расчетов могут задаваться пользователем, необходимы специфические журналы расчетов. Журналы расчетов хранятся в отдельных файлах подобно документам и справочникам. Имена журналов состоят из префикса sj и номера объекта метаданных в файле конфигурации 1Cv7.md.

{"CJ",

{"447","Зарплата","Журнал расчета заработной платы","Журнал зарплата",
"16","2","2450419",

{"CJParams",

{"448","Дни","Отработанные дни","Дни","N","6","2","0","0","0"},

{"449","Часы","Отработанные часы","Часы","N","7","2","0","0","0"},

{"1089","НомерСтрокиДокумента","Номер строки документа","Номер строки документа","N","4","0","0","1","0"},

{"456","СтрокаИсправления","Номер строки документа-исправления",
"Строка документа-исправления","N","3","0","0","0","0"}, {"15","2","1"},

{"FF","202"},

{"Form",

{"2703","ФормаСписка","",""},

{"2707","ПоСотруднику","Открывается из спр. Сотрудники",""}, {"2703",
"2703"},

```
{ "757", "Дополнительный", "Расчет дополнительной заработной платы",
"Журнал дополнительной зарплаты", "16", "2", "2450419",
{"CJParams",
{"752", "Дни", "Дни", "Дни", "N", "6", "2", "0", "0", "0"},
{"753", "Часы", "Часы", "Часы", "N", "7", "2", "0", "0", "0"},
{"1236", "НомерСтрокиДокумента", "Номер строки документа", "Номер строки
документа", "N", "4", "0", "0", "0", "0"}, "15", "2", "0",
{"FF", "202"},
{"Form",
{"755", "ФормаСписка", "", ""}, "755", "755"},
```

Описание таблицы журнала расчетов "Зарплата" в словаре базы данных представлено фрагментом:

```
#==TABLE по 123 : Журнал расчетов Зарплата
```

# Name	Descr	Type[A/S/U]	DBTableName	ReUsable
T=CJ447	Журнал расчетов Зарплата	A	CJ447	1

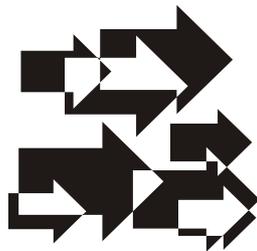
```
#-----Fields-----
```

# Name	Descr	Type	Length	Precision
F=IDDOC		C	9	0
F=IDS		C	9	0
F=IDALG		C	4	0
F=ORDER		N	3	0
F=RESULT	Result	N	15	2
F=DATEB	date1	D	8	0
F=DATEE	date2	D	8	0
F=PERIOD		C	9	0
F=RECALC		N	3	0
F=ID		C	9	0
F=DP		N	1	0
F=IDPARDOC		C	9	0
F=IDRECALC		C	9	0
F=FF202	(P)ОсновнойЭлемент	C	9	0
F=SP448	(P)Дни	N	7	2
F=SP449	(P)Часы	N	8	2
F=SP1089	(P)НомерСтрокиДокуме	N	5	0
F=SP456	(P)СтрокаИсправления	N	4	0

#----Indexes-----

# Name	Descr	Unique	Indexed fields	DBName
I=IDDOC+PER		0	IDDOC,PERIOD,IDS,ORDER	IDDOC+PERIO
I=PERIOD+ID		0	PERIOD,IDS,ORDER,DATEB	PERIOD+IDS+
I=IDS+PERIO		0	IDS,PERIOD,ORDER,DATEB	IDS+PERIOD+
I=ID		0	ID	ID
I=IDS+DATEE		0	IDS,DATEE,ID	IDS+DATEE+I
I=DATEE+ID		0	DATEE,ID	DATEE+ID
I=IDPARDOC		0	IDPARDOC	IDPARDOC
I=IDRECALC		0	IDRECALC	IDRECALC
I=FF202	FF202	0	FF202,PERIOD,IDS,ORDER, DATEB	FF202

ГЛАВА 3



Элементы диалога

Элементы диалога

Пользователь в процессе работы так или иначе имеет дело с окнами диалогов. Окно диалога представляет собой прямоугольную область экрана, которая, в самом общем случае, содержит поясняющие надписи, поля для ввода информации и управляющие элементы (кнопки), которые называются элементами диалога. С помощью диалогов программа запрашивает у пользователя ту информацию, которая необходима ей для дальнейшей работы, либо выдает какую-либо информацию пользователю для просмотра и подтверждения. Назначение диалога — предоставить пользователю удобную форму для ввода и просмотра информации. Как и шаблон бумажного документа, диалог позволяет быстро ввести необходимую информацию и запомнить ее для последующей обработки, а при необходимости вновь вернуться к ранее введенным данным для просмотра или корректировки. При создании форм для ввода информации в документы и справочники редактор диалогов использует элементы следующих типов:

- **Текст** — элемент позволяет отображать в окне диалога произвольный текст. Текстовое поле может быть статическим или информационным. В текстовых полях текст не меняется в процессе ввода информации в диалог. В информационных полях может выдаваться различная информация, изменяющаяся в процессе работы с диалогом.
- **Поле ввода** — элемент служит для ввода информации, которая затем будет запоминаться и (или) обрабатываться. Внешне реквизит представляет собой прямоугольное поле для ввода текста. В процессе работы с диалогом в поле ввода появляется курсор, что означает готовность для ввода данных. Тип данных определяется при создании элемента.

- ❑ **Таблица** — элемент организует ввод информации в таблицу. Автоматически создается, когда связанный с диалогом элемент данных имеет табличную часть.
- ❑ **Дерево** — элемент, связанный с таблицей, используется, когда представляемый таблицей список данных может иметь иерархическую структуру (например, Справочник). В таких случаях наряду с обязательным элементом "Таблица" в диалог может быть введен элемент "Дерево" для реализации возможности показа иерархического списка в двух видах: в виде таблицы и в виде дерева.
- ❑ **Список** — элемент позволяет показать пользователю список значений и предоставляет возможность выбрать одно из этих значений.
- ❑ **Поле со списком** — назначение данного элемента диалога аналогично элементу типа Список. Разница между ними заключается в их визуальном представлении: для доступа к списку значений элемента типа Поле со списком следует нажать кнопку, расположенную в правой части поля.
- ❑ **Флажок** — элемент предоставляет пользователю возможность управления типа Включено/Выключено. В зависимости от состояния элемента типа Флажок может быть выбран тот или иной путь обработки введенной пользователем информации.
- ❑ **Кнопка** — элемент позволяет создать в окне диалога стандартную Windows-кнопку. С кнопкой можно связать некоторое действие, которое будет выполняться при нажатии на нее.
- ❑ **Рисунок** — элемент позволяет разместить в диалоге рисунок. Не несет никакой функциональной нагрузки и используется только как оформительский элемент.
- ❑ **Рамка** — элемент позволяет создать вокруг группы элементов рамку с поясняющим заголовком. Никакой функциональной нагрузки данный элемент не несет, он используется только для рациональной организации других элементов диалога.

Любой элемент диалога можно разместить в диалоговом окне выбором соответствующего пункта меню **Вставить**. Из открывшегося меню необходимо выбрать тип создаваемого элемента диалога. Теперь, если поместить указатель мыши в пределах окна диалога, он примет вид перекрестия, возле которого будет расположен символ вставляемого элемента. Перекрестие необходимо поместить в одном из углов предполагаемой области размещения нового реквизита и нажать левую кнопку мыши. Далее следует перемещать указатель мыши, не отпуская левой кнопки мыши. При перемещении указателя мыши на экране будет рисоваться тонкая прямоугольная рамка. Один угол рамки будет расположен в том месте, где вы начали движение мыши, а

противоположный угол — там, где находится указатель мыши в настоящий момент. Перемещая указатель мыши, вы можете изменять размеры рамки, определяя таким образом примерные размеры и расположение создаваемого реквизита. Элемент диалога будет создан после того, как вы отпустите левую кнопку мыши. Если вас не устраивает расположение и размеры созданного элемента диалога, вы можете легко их изменить.

Реквизиты

Самый важный элемент диалога — *реквизит*. Он служит для ввода и редактирования непосредственно полей базы данных. Реквизит может принимать различные формы в зависимости от типа значения, который ему назначен. Список типов значений зависит от количества объектов метаданных, созданных в конфигурации. Рассмотрим пример объекта, в котором присутствуют реквизиты различных типов. Зададим в созданном справочнике "Квартиры" (рис. 3.1) следующие реквизиты:

- Договор типа "Строка" длиной 10;
- КоличествоКомнат типа "Число" длиной 2;
- МетражОбщий типа "Число" длиной 8 с точностью 2;
- Льгота типа "Число" длиной 6 с точностью 2;

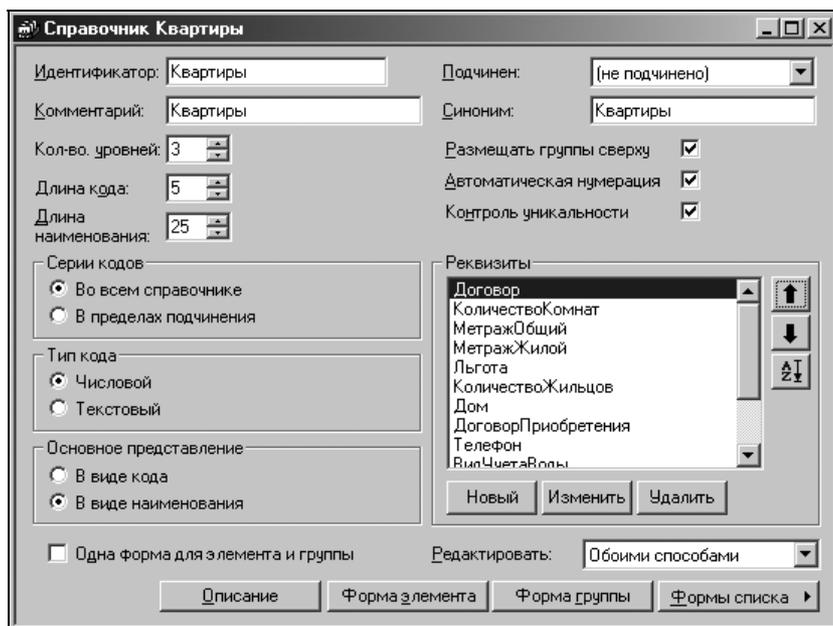


Рис. 3.1. Диалоговое окно Справочник Квартиры

- КоличествоЖильцов типа "Число" длиной 2 с точностью 0;
- Дом типа "Справочник.Дома";
- ДоговорПриобретения типа "Строка" длиной 100;
- Телефон типа "Строка" длиной 10;
- ВидУчетаВоды типа "Число" длиной 2;
- ПоказаниеХолВоды типа "Число" длиной 15 с точностью 2;
- ПоказаниеГорВоды типа "Число" длиной 15 с точностью 2;
- ОсновнойЭлемент типа "Справочник.Квартиры";
- ШтрихКод типа "Строка" длиной 10.

По нажатию кнопки **Форма элемента** программа предложит заполнить диалог элементами диалога, соответствующими заданным реквизитам. Количество реквизитов слишком велико для автоматического расположения в удобном виде, поэтому откажемся от расположения некоторых элементов путем отключения флажка в списке (рис. 3.2).

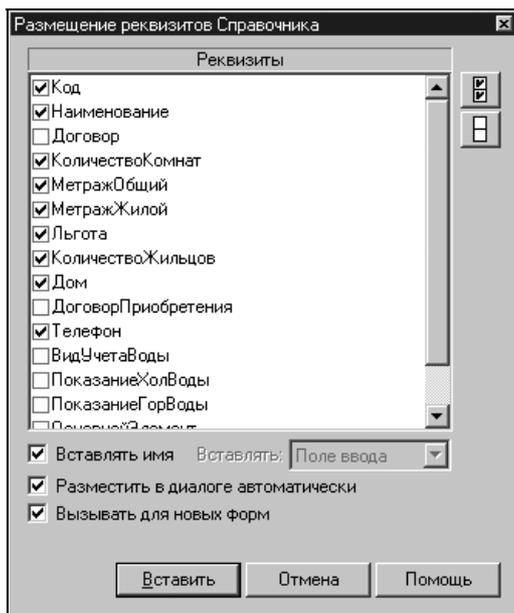


Рис. 3.2. Диалоговое окно размещения реквизитов справочника

В созданной форме элемента справочника (рис. 3.3) в правой части элемента диалога **Дом** располагается кнопка выбора, т. к. тип значения данного элемента — Справочник.Дома.

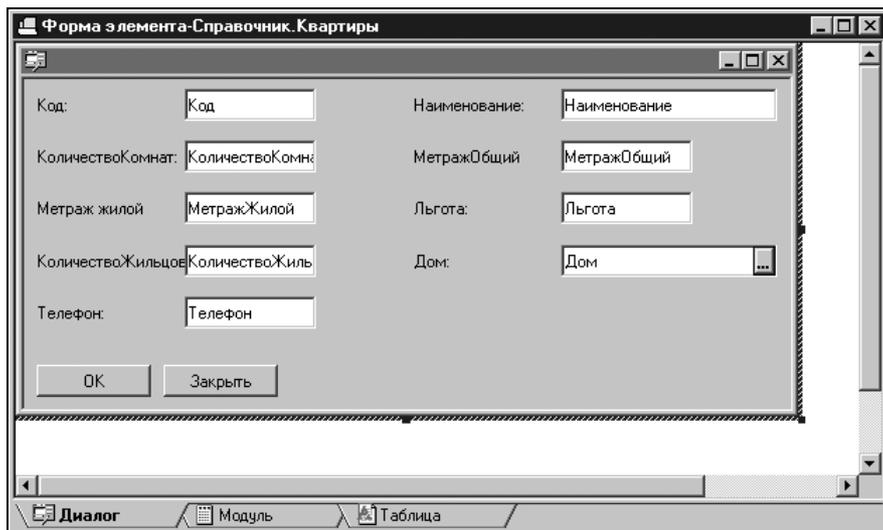


Рис. 3.3. Диалоговое окно формы элемента справочника **Квартиры**

На свободное место в форме добавим элемент диалога **ВидУчетаВоды**. Данный реквизит имеет тип "Число" длиной 1 и может принимать следующие значения:

- 0 — вид учета воды по норме;
- 1 — вид учета воды по водомеру.

Для корректного ввода значений данный реквизит должен иметь вид флажка. Для этого необходимо выбрать в окне **Размещение реквизитов Справочника** пункт **Реквизиты** меню **Вставить**. В предложенном диалоговом окне размещения элементов справочника включим флажок реквизита **ВидУчетаВоды**. Список с наименованием **Вставлять** предлагает выбор одного из вариантов элемента диалога (рис. 3.4):

- Поле ввода;
- Флажок;
- Переключатель.

Выберем из списка вариант **Флажок** и укажем мышью его месторасположение в окне.

После размещения реквизита в окне флажок присвоится автоматически по идентификатору реквизита. Включенный флажок означает, что учет воды ведется по водомеру, выключенный — по норме. Для удобства пользователя изменим заголовок, для чего двойным щелчком мыши на элементе вызовем окно **Свойства флажка** (рис. 3.5).

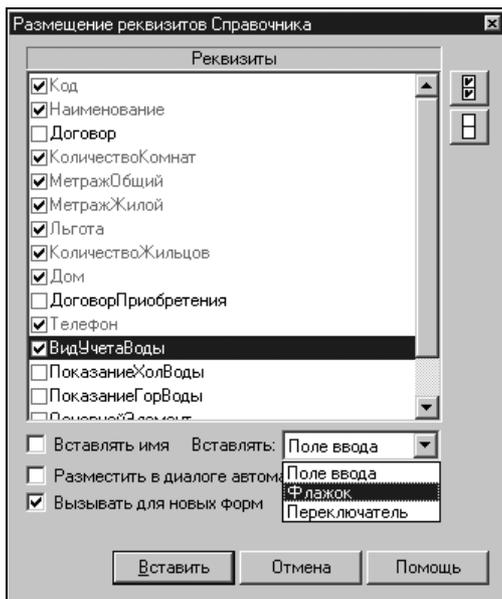


Рис. 3.4. Диалоговое окно **Размещение реквизитов Справочника**

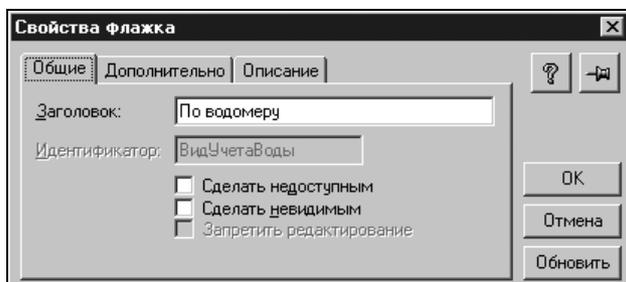


Рис. 3.5. Диалоговое окно **Свойства флажка**

Окно формы справочника с флажком представлено на рис. 3.6.

Допустим, что реквизит **ВидУчетаВоды** может принимать одно из следующих значений:

- По водомеру;
- По норме;
- Не учитывать.

Очевидно, что форма элемента диалога типа "Флажок" в данном случае не годится, здесь идеально подходит форма элемента диалога типа "Переключатель".

При размещении элемента диалога типа "Переключатель" очень важно соблюдать следующую последовательность действий:

1. Разместить простой элемент диалога типа "Переключатель".
2. Разместить еще один простой элемент диалога типа "Переключатель".
3. Разместить реквизит в форме элемента диалога типа "Переключатель".

Внимание!

Несмотря на то, что реквизит размещается в группе последним, он должен располагаться сверху, если переключатель вертикальной ориентации; слева — если переключатель горизонтальной ориентации. Нарушение данного порядка вызывает разъединение переключателя.

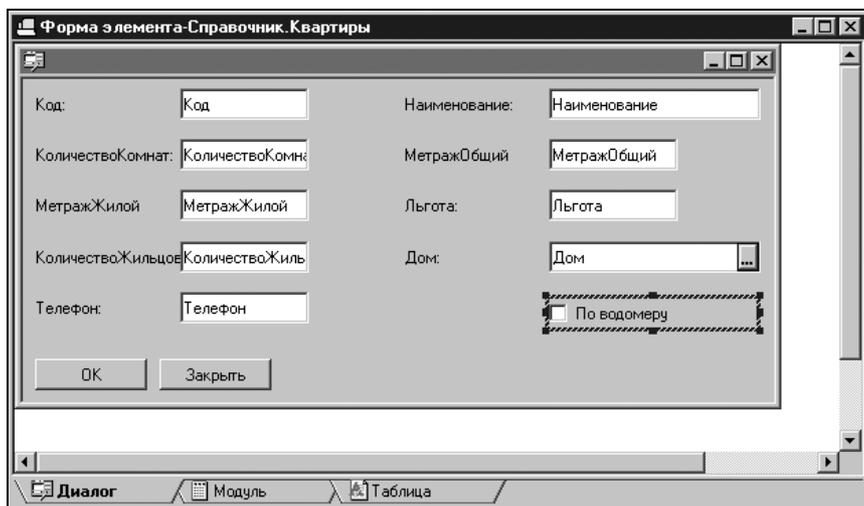


Рис. 3.6. Диалоговое окно **Форма элемента-Справочник.Квартиры** с флажком

Для размещения простого элемента диалога типа "Переключатель" необходимо в меню **Вставить** выбрать пункт **Переключатель** и разместить его в окне. Разместим в окне элемента справочника два элемента типа "Переключатель" и изменим заголовки. При выборе из списка выберем форму элемента типа "Переключатель" и разместим элемент сверху группы. Для оформления группы рамкой необходимо в меню **Вставить** выбрать пункт **Рамка** и задать заголовок (рис. 3.7).

Внимание!

В группе переключателей должен быть только один элемент с признаком **Первый в группе**. Поскольку реквизит в форме элемента типа "Переключатель" автоматически приобретает статус первого элемента, то в остальных элементах группы данный признак должен быть выключен (рис. 3.8).

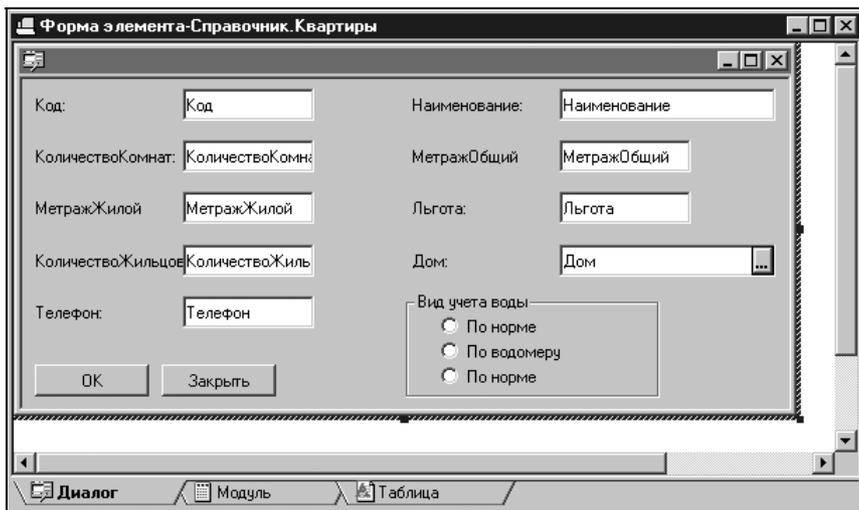


Рис. 3.7. Диалоговое окно **Форма элемента-Справочник.Квартиры** с переключателем

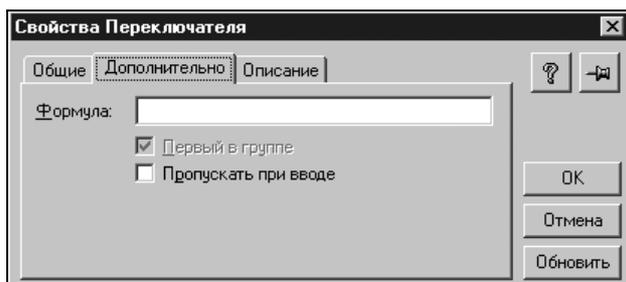


Рис. 3.8. Диалоговое окно свойств элемента диалога на вкладке **Дополнительно**

Для справочников существует реквизит **Иерархия**, который не задан в группе реквизитов, он присутствует в группе predetermined реквизитов:

- Код;
- Наименование;
- Иерархия.

Иерархия задается количеством уровней в форме справочника. Если заданное количество уровней меньше 2, следовательно, иерархия не задана, и пункт меню **Дерево групп** не активизируется. Данный реквизит имеет смысл только для формы списка справочника. Зададим форму списка справочника "Квартиры", предварительно задав количество уровней 3. Для задания формы списка справочника необходимо в диалоговом окне справочника нажать

кнопку **Формы списка**, в предложенном меню выбрать пункт **ФормаСписка** (рис. 3.9).

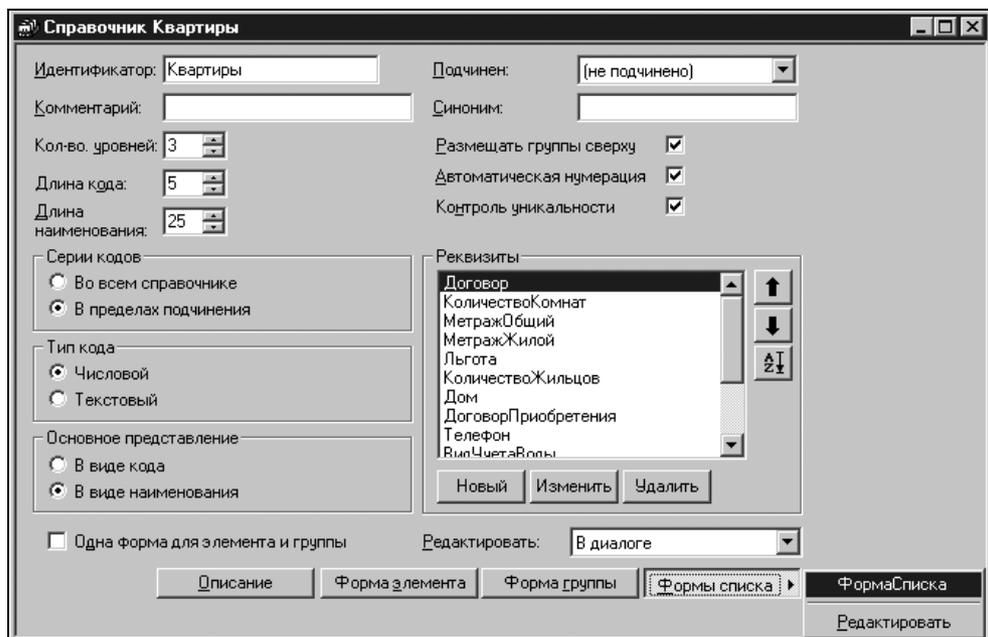


Рис. 3.9. Выбор формы списка в справочнике

В диалоговом окне размещения реквизитов справочника необходимо включить флажок **Вставить "Дерево групп"** (рис. 3.10).

Для формы списка характерно наличие только тех реквизитов, которые необходимы для поиска и идентификации элемента справочника (рис. 3.11).

Для рассмотрения элемента диалога типа Таблица значений создадим новый объект метаданных **Документ.ВводНачальногоСальдо** для ввода задолженности по квартирам. Для этого необходимо в окне конфигурации на вкладке **Метаданные** установить указатель на пункт **Документы** и выбрать в меню **Действия** пункт **Новый элемент**. Зададим в созданном документе **ВводНачальногоСальдо** (рис. 3.12) следующие реквизиты:

- НачальноеСальдо** типа Число длиной 8 с точностью 2;
- Квартира** типа Справочник.Квартиры.

Реквизиты шапки размещаются в окне такими же элементами диалога, как и в справочнике, реквизиты же табличной части являются колонками элемента диалога типа Таблица значений. При нажатии кнопки **Форма** предлагается

Документ ВводНачальногоСальдо

Идентификатор: ВводНачальногоСальдо Журнал: (Прочие)

Комментарий: Ввод начального сальдо Синоним: Ввод начального сальдо

Реквизиты шапки

Реквизиты табличной части

ТекущееСальдо
Квартира

Номер

Нумератор: << Не назначен >> Тип: Числовой Текстовый

Периодичность: По всем данного вида Длина: 5

Автоматическая нумерация Контроль уникальности

Разрешить проведение документа Бухгалтерский учет

Автоматическое удаление движений Расчет

Автоматическая нумерация строк Оперативный учет

Создавать операцию: Всегда Редактировать операцию

Ввод на основании... Описание Форма Модуль Документа

Рис. 3.12. Диалоговое окно документа ВводНачальногоСальдо

Форма-Документ.ВводНачальногоСальдо (Ввод начального сальдо)

Размещение реквизитов Документа

Реквизиты

НомерДок
 ДатаДок

Вставлять имя Вставлять: Поле ввода

Реквизиты табличной части

НомерСтроки
 ТекущееСальдо
 Квартира

Разместить в диалоге автоматически
 Вызывать для новых форм

Вставить Отмена Помощь

Диалог Модуль Таблица

Рис. 3.13. Диалоговое окно размещения реквизитов документа

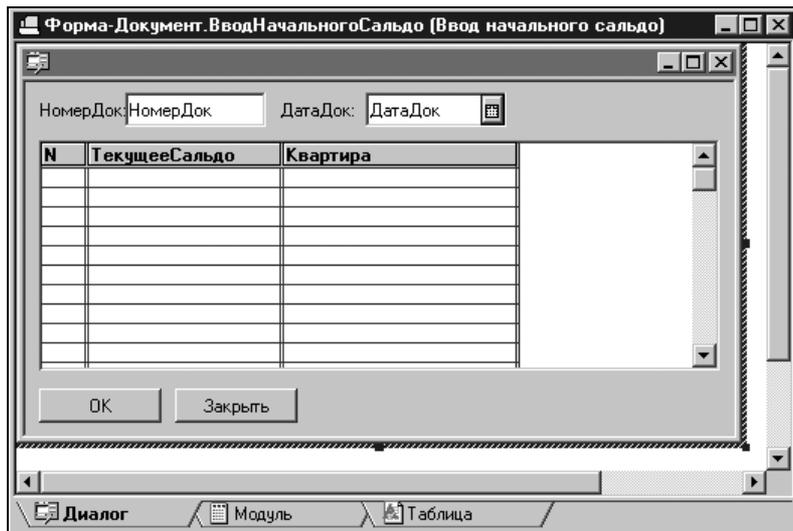


Рис. 3.14. Диалоговое окно формы документа **ВводНачальногоСальдо**

По нажатию кнопки **Вставить** табличная часть документа размещается в окне документа автоматически (рис. 3.14).

Действия с элементами диалога

Для того чтобы над элементом диалога можно было произвести какие-либо действия, необходимо его выбрать, т. е. некоторым образом отметить.

Выбор элемента

Для выбора элемента диалога необходимо указать курсором на нужный элемент диалога и нажать левую кнопку мыши. Выбранный элемент окаймляется рамкой, на которой находятся восемь небольших квадратных маркеров черного цвета: четыре из них расположены в середине каждой из сторон рамки и четыре — в ее углах. Эти маркеры служат для изменения размеров элемента диалога.

Для выбора любого другого элемента диалога достаточно щелкнуть мышью на этом элементе. При этом предыдущий выбор будет отменен, рамка с выбранного ранее элемента будет снята и помещена вокруг вновь выбранного элемента диалога.

Вы можете выбрать сразу несколько элементов диалога. Сделать это можно разными способами.

Для выбора группы элементов, расположенных друг возле друга, нужно выполнить следующие действия:

1. Поместите указатель мыши вне группы элементов диалога, которые вы хотите выбрать.
2. Нажмите и держите нажатой левую кнопку мыши. Теперь, если вы будете перемещать указатель мыши, на экране будет изображаться тонкая прямоугольная рамка. Один угол рамки будет расположен в том месте, где вы начали движение мыши, а противоположный угол — там, где находится указатель мыши в настоящий момент. Перемещая указатель мыши, вы можете изменять размеры этой рамки.
3. Поместите указатель мыши так (не отпуская при этом левой кнопки мыши), чтобы внутри рамки находились элементы диалога, которые вы хотели выделить.
4. Отпустите левую кнопку мыши.

Теперь все элементы диалога, которые оказались внутри рамки, будут выделены. Вокруг каждого из них будет нарисована пунктирная рамка — такая же, как окаймляющая одиночный выбранный элемент.

Для выбора нескольких элементов можно воспользоваться другим приемом: если выделить элемент диалога, держа при этом нажатой клавишу <Shift>, то ранее сделанное выделение не снимается, т. е. вновь выделяемый элемент будет добавлен к уже выбранным.

Для выбора группы элементов таким способом необходимо выполнить следующие действия: выделить один элемент, затем нажать клавишу <Shift> и, не отпуская ее, выделить мышью требуемые элементы, после чего отпустить клавишу <Shift>. Этот способ удобен тем, что так можно выбирать группу произвольно расположенных элементов.

Перемещение

Для перемещения элемента диалога необходимо сначала выбрать его с помощью клавиш или мыши. Затем следует поместить указатель мыши над элементом так, чтобы указатель мыши принял форму крестика со стрелками. Тогда можно нажать левую кнопку мыши, не отпуская ее, переместить элемент на нужное место и после этого отпустить кнопку. Если включена функция **Ориентировать по разметке**, то перемещение элемента будет осуществляться "скачкообразно" — левая и верхняя стороны рамки, ограничивающей элемент диалога, будут автоматически располагаться вдоль ближайших линий разметочной сетки.

Изменение размеров

Для изменения размера элемента диалога необходимо сначала выбрать его с помощью мыши или клавиатуры. Вокруг выбранного элемента диалога будет

помещена рамка, на которой расположены восемь небольших квадратных маркеров черного цвета: четыре из них расположены в середине каждой из сторон рамки и четыре — в ее углах. Квадраты, расположенные на сторонах рамки, предназначены для изменения одного размера элемента (длины или ширины). Квадраты, находящиеся в углах, служат для изменения сразу обоих размеров.

Для изменения высоты или ширины элемента диалога необходимо указать мышью на один из маркеров, расположенных на сторонах элемента диалога, пока указатель мыши не примет форму вертикальной или горизонтальной двунаправленной стрелки, и нажать левую кнопку мыши. Теперь, перемещая мышью, вы можете установить для элемента требуемую ширину или высоту. Перемещаться при этом будет та сторона реквизита, которую вы "тянете" мышью; противоположная сторона будет оставаться на месте. Когда необходимый размер будет установлен, необходимо отпустить левую кнопку мыши. Если включена функция **Ориентировать по разметке**, то изменение размеров элемента диалога будет осуществляться "скачкообразно" — та сторона рамки, ограничивающей элемент диалога, которую вы "тянете" мышью, будет автоматически располагаться вдоль ближайшей линии разметочной сетки.

Если вы хотите изменить оба размера элемента диалога — и ширину, и высоту, — надо указать мышью на маркер, расположенный в одном из углов рамки, ограничивающей элемент диалога. Указатель мыши при этом должен принять форму двунаправленной стрелки, расположенной наклонно. Если нажать левую кнопку мыши, то при перемещении мыши будут изменяться высота и ширина рамки, ограничивающей элемент диалога. Угол рамки, противоположный выбранному, при этом остается неподвижным. Установив нужный размер элемента диалога, можно отпустить кнопку мыши.

Как и при изменении одного размера элемента диалога, при включенной функции **Ориентировать по разметке** тот угол, которые вы "тянете" мышью, будет автоматически устанавливаться в ближайший узел разметочной сетки.

Автоматическая установка размеров

Для установки одинакового размера нескольких элементов вы можете использовать функцию **Ориентировать по разметке**. Однако редактор диалогов позволяет автоматически устанавливать размеры элементов диалога, используя один из элементов в качестве образца.

Для установки размера элементов диалога следует сначала выбрать эти элементы (должны быть выбраны как минимум два элемента). На рамке, ограничивающей последний из выбранных элементов, появятся маркеры, указывающие, что этот элемент будет выбран как образец для установки размеров

всех выделенных элементов. Вы можете выбрать в качестве образца любой другой элемент диалога. Для этого необходимо нажать клавишу <Shift> и, не отпуская ее, щелкнуть мышью нужный элемент.

Автоматическая установка размеров выполняется с помощью подменю **Одинаковый размер** меню **Диалог**. Вы можете установить для выбранных элементов одинаковый размер по высоте (команда **По высоте**), по ширине (команда **По ширине**) или установить одинаковый размер всех выбранных элементов (команда **Одинаковый размер**).

Выравнивание

Аккуратное размещение элементов диалога в окне позволяет впоследствии удобно и эффективно с ними работать. Редактор диалогов предоставляет разнообразные возможности для выравнивания элементов в окне диалога. Вы можете использовать разметку для задания точного положения или одинакового размера элементов диалога в процессе их создания или размещения в окне диалога. После того как все необходимые элементы диалога уже размещены в окне, для автоматического выравнивания их положения или размеров можно пользоваться функциями меню **Диалог**. Они позволяют размещать элементы диалога, задав в качестве ориентира положение любого из них, помещать реквизит в центр окна диалога, задавать элементам диалога одинаковые размеры, а также автоматически устанавливать равные расстояния между ними.

Выравнивание по образцу

Вы можете значительно улучшить внешний вид созданного вами диалога, если его элементы, расположенные друг под другом, будут выровнены по вертикали, а элементы, расположенные в ряд, — по горизонтали. Функция **Выровнять** меню **Диалог** позволяет указать, какие стороны выбранных элементов диалога будут выровнены.

Для выравнивания элементов их сначала следует выбрать: должно быть выбрано как минимум 2 элемента диалога. В качестве образца используется элемент, вокруг которого размещены квадратные маркеры (среди группы выбранных элементов такой элемент — единственный). Вы можете выбрать в качестве образца любой другой элемент диалога. Для этого необходимо нажать клавишу <Shift> и, не отпуская ее, щелкнуть мышью нужный элемент. На пунктирной рамке вокруг этого элемента появятся маркеры. Затем следует в главном меню открыть группу функций **Диалог** и выбрать подменю **Выровнять**. Команды, включенные в это подменю, позволяют указать, по какой стороне элемента-образца будут выровнены остальные выделенные элементы:

- **Влево** — выровнять левые стороны всех выделенных элементов по левой стороне элемента-образца;
- **Вправо** — выровнять правые стороны всех выделенных элементов по правой стороне элемента-образца;
- **По низу** — выровнять нижние стороны всех выделенных элементов по нижней стороне элемента-образца;
- **По верху** — выровнять верхние стороны всех выделенных элементов по верхней стороне элемента-образца.

Центрирование элемента

Иногда возникает необходимость поместить какой-либо из элементов в середину окна диалога. Можно сделать это используя разметку, а можно воспользоваться подменю **Центр** из меню **Диалог**. Команды этого подменю позволяют автоматически переместить выбранный элемент диалога (или несколько элементов) в середину окна диалога:

- **По вертикали** — выбранный элемент будет передвинут по горизонтали и размещен так, чтобы его центр находился на одной вертикальной линии с центром окна диалога;
- **По горизонтали** — выбранный элемент будет передвинут по вертикали и размещен так, чтобы его центр находился на одной горизонтальной линии с центром окна диалога.

Удаление элементов диалога

Для удаления элемента диалога его необходимо выбрать с помощью мыши или клавиатуры и нажать клавишу .

Отмена сделанных изменений

Для отмены ошибочных действий следует воспользоваться функцией **Отменить последнее действие** меню **Действия**. При выборе этой функции последняя выполненная операция — создание элемента диалога, его перемещение, копирование, изменение размеров и т. п. — будет отменена. Для быстроты удобнее использовать горячие клавиши — комбинацию <Ctrl>+<Z> или ее аналог <Alt>+<Backspace>.

Однако существует более простой способ отменить выполнение нежелательных действий. Предположим, вы перемещаете какой-либо элемент диалога мышью и "по дороге" вдруг передумали это делать. Тогда, не отпуская левую кнопку мыши, нажмите клавишу <Esc> — тонкая рамка, отмечающая поло-

жение элемента диалога в процессе перемещения, пропадет. Теперь можно отпустить левую кнопку мыши — никаких изменений сделано не будет.

Этот прием применим и для других операций, выполняемых мышью: создание элемента диалога, его копирование и изменение размеров.

Определение порядка обхода элементов диалога

Редактор диалогов позволяет установить порядок обхода элементов по нажатию клавиши <Tab> при работе пользователя с диалогом.

По умолчанию обход элементов диалога осуществляется слева направо сверху вниз. Для изменения порядка обхода следует выбрать функцию **Порядок обхода** из меню **Диалог**. После выбора этой функции у каждого элемента появится квадратная рамка с цифрой, соответствующей порядковому номеру элемента при обходе. Вы можете задать порядок обхода, выбирая мышью элементы диалога в нужной последовательности. Нумерация элементов при этом будет изменяться, показывая выбранную последовательность обхода. Для выхода из функции определения порядка обхода достаточно щелкнуть мышью в любом свободном месте окна редактора диалогов.

Восстановить стандартный порядок обхода элементов можно, выбрав функцию **Автоматический порядок обхода** из меню **Диалог**.

Работа с таблицей

Элемент типа "Таблица" используется для доступа к группе реквизитов, входящих в табличную часть элемента данных — Журнала, Справочника или Документа.

По умолчанию таблица автоматически создается и помещается в форму диалога. Составляющие таблицу реквизиты будут представлены в виде колонок с соответствующими заголовками. Дополнительно в таблицу помещается служебная графа, она занимает крайнюю левую позицию, и не может быть удалена. В зависимости от контекста использования элемента "Таблица" в этой графе может выводиться дополнительная информация.

Работа с таблицей при редактировании диалога имеет ряд особенностей. Как и всякий другой элемент, вы можете перемещать таблицу по окну диалога и изменять ее размеры. Однако таблицу нельзя скопировать, а также нельзя удалить таблицу справочника или журнала.

В пределах элемента "Таблица" можно работать с составляющими таблицу колонками, кроме служебной колонки. Например, вы можете менять порядок расположения колонок табличной части. Для этого следует выполнить следующие действия:

1. Указать мышью на заголовок и нажать левую кнопку мыши.
2. Не отпуская кнопку мыши, переместить указатель мыши в то место таблицы, куда предполагается переместить выбранную колонку.
3. Отпустить левую кнопку мыши.

Колонка, которая была на указанном месте раньше, будет сдвинута в сторону. Таким образом, перетаскивая колонки мышью, можно расставить их в нужном порядке.

Если не определен иной порядок, колонки расставляются в соответствии с порядком соответствующих им реквизитов в списке — чем выше реквизит, тем левее предназначенная для него колонка табличной части.

Если элемент "Таблица" используется для создания экранного представления журнала, можно расширять состав таблицы, вставляя в нее колонки, соответствующие дополнительным графам Журнала. Для этого используется пункт **Графы** меню **Вставить** главного меню программы.

Можно удалить колонку табличной части. Для этого необходимо выбрать ее, щелкнув мышью на заголовке колонки, и нажать клавишу . Для вставки колонки в табличную часть диалога следует использовать функцию **Реквизиты** из группы функций **Вставить** главного меню программы.

Для изменения ширины колонок табличной части необходимо передвигать мышью разделитель между колонками.

Внимание!

Следует помнить, что если включен признак **Автонастройка ширины колонок** в свойствах таблицы, ширина колонок при работе с диалогом будет определяться автоматически.

Большинство характеристик той или иной колонки таблицы определяется свойствами связанного с ней реквизита Документа, Справочника или Журнала. Однако можно установить свойства колонке как элементу диалога. Для вызова на экран диалога, для определения свойств необходимо выбрать колонку, затем нажатием правой кнопки мыши вызвать на экран контекстное меню, из которого выбрать функцию **Свойства**. На экран будет выдан диалог для установки свойств элемента диалога типа "Реквизит".

Определение свойств элементов диалога

Каждый элемент диалога имеет определенный набор свойств, от которого зависит поведение элемента при работе пользователя с готовым диалогом и порядок обработки введенной в диалог информации.

Установка свойств элементов диалога производится в палитре свойств **Свойства**, которую можно вызвать двойным щелчком мыши на элементе диалога.

Диалог **Свойства** организован в виде картотеки. Все свойства, задаваемые для элемента диалога, объединены в несколько групп, количество которых зависит от типа элемента. Для доступа к управляющим элементам конкретной группы необходимо щелкнуть мышью на соответствующей вкладке в окне диалога **Свойства**.

Внимание!

Для реквизитов объектов метаданных типа "Справочник" и "Документ" свойства **Заголовок**, **Идентификатор** и **Тип** задаются в диалоговом окне **Свойства реквизита**, а не в окне **Свойства элемента диалога**. В окне свойств элемента диалога данные свойства недоступны. Эта особенность связана с определением информационных полей базы данных.

В зависимости от типа элемента диалога состав управляющих элементов той или иной вкладки может меняться, кроме того, отдельные элементы вкладок могут становиться недоступными для использования. Недоступные для редактирования элементы диалога выводятся серым цветом. В справочнике "Квартиры" в окне **Свойства Переключателя** на вкладке **Дополнительные** порядковый номер строки переключателя предопределен, т. к. данная строка переключателя является реквизитом (см. рис. 3.8).

Кнопки диалога свойств

В окне диалога свойств присутствуют следующие кнопки:

- ОК** — позволяет применить установленные в диалоге свойства к объекту, и закрывает окно;
- Отмена** — позволяет закрыть окно без использования последних установок свойств объекта;
- Обновить** — позволяет применить установленные в диалоге свойства к объекту, но не закрывает само окно.

Вкладка **Общие**

Вкладка служит для определения общих свойств элемента диалога. Состав управляющих элементов вкладки и их доступность зависит от объекта метаданных и типа редактируемого элемента диалога.

Реквизит **Заголовок** задает текст, который будет выводиться в заголовке окна диалога при работе с диалогом. Если заголовок не указан, будет выводиться название объекта метаданных, связанных с этим диалогом.

Реквизит **Идентификатор** позволяет обращаться к элементу из модуля по идентификатору.

Флажок **Сделать недоступным** устанавливает доступ к элементу при использовании. Если флажок установлен, элемент диалога будет недоступен при использовании диалога. Недоступные элементы будут выводиться "серым" цветом. Управление этим свойством доступно из встроенного языка системы 1С:Предприятие, что позволяет динамически устанавливать для элемента диалога атрибут **Сделать недоступным**.

Реквизит **Сделать невидимым** устанавливает режим отображения элемента в окне. Если флажок установлен, элемент не будет отображаться при использовании диалога. Управление этим свойством доступно из встроенного языка системы 1С:Предприятие, что позволяет динамически управлять видимостью элемента диалога.

Реквизит **Запретить редактирование** устанавливает режим редактирования элемента. Этот флажок становится доступным для элементов диалога типа **поле ввода**, имеющих тип значения "Число", "Строка", "Дата" и "Счет". Если флажок установлен, запрещено редактирование содержимого поля ввода с клавиатуры. Однако в такое поле можно вводить значение, используя выбор: например, ввести бухгалтерский счет, выбрав его из плана счетов.

Реквизит **Режим сохранения настройки** устанавливает режим сохранения настройки параметров отчета. Этот флажок появляется во вкладке, когда редактируется форма объекта метаданных типа "Отчет" ("Обработка") или внешнего отчета (обработки). Если этот флажок установлен, то при работе в режиме запуска "1С:Предприятие" для этого отчета будут доступны функции сохранения и восстановления настройки — текущих значений реквизитов диалога отчета.

После автоматического размещения реквизитов в окне диалога наименования реквизитов в элементах диалога типа "Текст" совпадают с идентификаторами самих реквизитов, которые не содержат пробелов. Для пользователя такие заголовки неестественны. Изменим заголовок в тексте, который комментирует реквизит **Метраж жилой** (рис. 3.15).

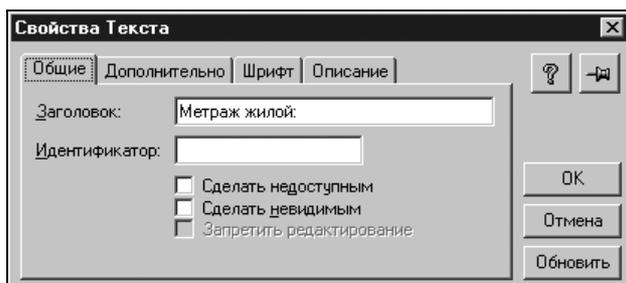


Рис. 3.15. Диалоговое окно свойств элемента диалога на вкладке **Общие**

Особое внимание следует обратить на элемент диалога типа "Таблица". Табличная часть документа не является элементом типа "Таблица" и имеет другие свойства. Рассмотрим форму документа **ВводНачальногоСальдо**. В диалоговом окне свойств табличной части документа вкладка **Общие** содержит свойства **Авто настройка ширины колонок** и **Автоматический ввод новой строки** (рис. 3.16).

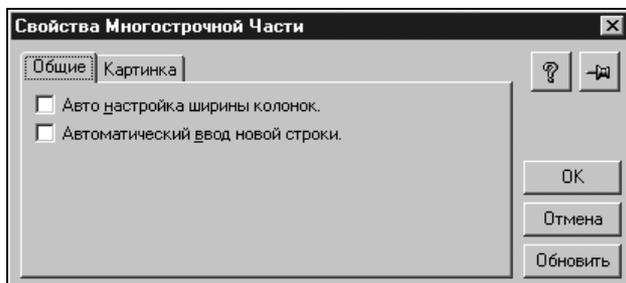


Рис. 3.16. Диалоговое окно свойств табличной части документа на вкладке **Общие**

Если атрибут **Авто настройка ширины колонок** включен, то ширина колонок табличной части диалога, независимо от сделанных установок при редактировании диалога, будет устанавливаться одинаковой. Если этот элемент выключен, то будут использоваться установки ширины колонок, сделанные в процессе редактирования диалога. При изменении ширины окна диалога будут сохраняться и относительные размеры колонок.

Если атрибут **Автоматический ввод новой строки** включен, то новая строка вводится в таблицу автоматически после окончания редактирования последнего элемента предыдущей строки. В противном случае — для ввода новой строки в таблицу следует нажать клавишу <Ins>.

Вкладка **Тип**

Вкладка **Тип** в поле **Тип** задает тип информации, которая может быть введена в данный элемент:

- Число** — в элемент можно ввести натуральное число (положительное или отрицательное). При вводе используются цифры от 0 до 9 и знак минус (-);
- Строка** — в элемент можно ввести произвольную последовательность символов;
- Дата** — в элемент можно ввести дату в формате ДД.ММ.ГГ.

Перечисленные типы значений имеют общее применение и используются достаточно широко и в других программных продуктах. Кроме них, в системе 1С:Предприятие можно использовать и специальные типы данных.

- **Справочник** — в качестве значения реквизита используется элемент данных из какого-либо определенного в системе объекта метаданных типа "Справочник". В этом случае ввод реквизита осуществляется следующим образом: сначала необходимо из списка справочников, выбрать нужный, и затем из этого справочника выбрать конкретный элемент данных. Кроме указания в качестве типа значения реквизита справочника неопределенного вида (**Справочник**), можно также указать наименование конкретного вида справочника (например, **Справочник.Дома**). Тогда при вводе значения реквизита на экран будет выдан указанный справочник (например, **Дома**), и из него надо будет выбрать наименование конкретного элемента для использования в качестве значения реквизита;
- **Документ** — в качестве значения реквизита используется ссылка на объект метаданных типа "Документ". При вводе в реквизит значения этого типа на экран можно вызвать список видов документов, выбрать из него требуемый вид документа и затем выбрать конкретный документ. Так же, как и для типа значения "Справочник", в качестве типа значения реквизита можно выбрать наименование конкретного вида документа (например, **Документ.ВводНачальногоСальдо**). Тогда при вводе значения в реквизит на экран будет выдан список документов указанного вида, из которого надо будет выбрать конкретный документ (например, документ № 00001 от 31.11.04);
- **Перечисление** — в качестве значения реквизита используется элемент перечисления. При указании этого типа значения из выдаваемого на экран списка необходимо выбрать наименование конкретного перечисления. При вводе реквизита этого типа на экран будет выдано перечисление, из которого требуется выбрать нужный элемент.

Реквизит **Разделять триады** устанавливает признак разделения триад при отображении значения объекта.

Реквизит **Длина** задает максимальное количество позиций, которое отводится для ввода значения реквизита. Длина указывается только для значений типа "Строка" и "Число". Для значения типа "Дата" его длина всегда равна 8 (ДД.ММ.ГГ), а для значений типа "Справочник" и "Документ" указания длины значения не требуется.

Реквизит **Точность** устанавливает максимально возможное количество знаков после десятичной точки для числового реквизита.

Вкладка Дополнительно

Управляющие элементы вкладки служат для установки различных дополнительных свойств элемента диалога. Состав управляющих элементов, которые могут помещаться на этой вкладке, зависит от типа реквизита и от типа зна-

чения реквизита. Кроме того, отдельные управляющие элементы могут закрываться от использования, в этом случае они выводятся серым цветом.

Реквизит **Формула** задает имя функции, которая содержит последовательность операторов на встроенном языке системы 1С:Предприятие, которые будут выполняться при изменении значения элемента. В формуле возможно задать простую последовательность операторов, не вынося ее в отдельную функцию. Данный реквизит присутствует во всех элементах диалога, кроме многострочной части документа. Рассмотрим в справочнике "Квартиры" свойства кнопки **ОК** на вкладке **Дополнительно** (рис. 3.17).

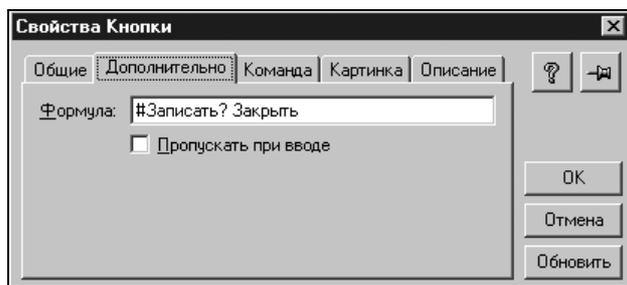


Рис. 3.17. Диалоговое окно **Свойства Кнопки** на вкладке **Дополнительно**

Символ # определяет использование последовательности прямых команд без ссылки на функцию модуля формы. В данном случае строка

#Записать? Закреть

дает при нажатии кнопки **ОК** команду записи с запросом на подтверждение и закрытие окна элемента справочника.

Реквизит **Связан с** используется для реквизитов типа "Справочник", если используемый в качестве типа значения справочник является подчиненным. Он может содержать идентификатор реквизита диалога, который имеет в качестве типа значения справочник-хозяин. При вводе значения в редактируемый элемент диалога выбор элемента из подчиненного справочника будет производиться без запроса элемента справочника-хозяина. Список элементов справочника, выданный на экран для выбора, будет содержать только элементы, подчиненные элементу справочника-хозяина. Для иллюстрации этого очень важного реквизита создадим справочник "Жильцы", подчиненный справочнику "Квартиры", и документ "Заявка" с использованием элемента-хозяина и подчиненного элемента.

Зададим следующие реквизиты справочника "Жильцы" (рис. 3.18):

- ФИО — строка длиной 50 символов;
- Паспорт — строка длиной 200 символов;

Рис. 3.18. Диалоговое окно Справочник Жильцы

- ДатаРождения — дата;
- СтатусПрописки — строка длиной 50 символов;
- СемейныйСтатус — строка длиной 10 символов;
- ДатаПрописки — дата;
- ДатаВыбытия — дата;
- Контрагент — справочник "Контрагенты".

Зададим следующие реквизиты шапки документа "Заявка" (рис. 3.19):

- Квартира — справочник "Квартиры";
- ДатаИсполнения — дата;
- КонтрольОплаты — число длиной 1;
- ДатаОплаты — дата;
- Дополнение — строка символов длиной 30;
- Заявитель — справочник "Жильцы".

Зададим следующие реквизиты табличной части документа "Заявка" (рис. 3.19):

- Услуга — справочник "Номенклатура";
- Количество — число длиной 8, точность 0;

- ЕдиницаИзмерения — строка символов длиной 10;
- Цена — число длиной 15, точность 2;
- Сумма — число длиной 8, точность 2;
- Всего — число длиной 8, точность 2.

Рис. 3.19. Диалоговое окно **Документ Заявка**

Заполним форму документа заданными реквизитами (рис. 3.20).

В диалоговом окне свойств реквизита шапки **Заявитель** в поле **Связан с** зададим идентификатор реквизита типа справочника-хозяина "Квартира" (рис. 3.21). Теперь для выбора значения реквизита **Заявитель** будет представлен список только жильцов квартиры, заданной в реквизите **Квартира**.

Реквизит **Форма** используется только для реквизитов типа "Справочник". Позволяет выбрать форму, которая будет использоваться при выдаче справочника на экран для выбора из него значения. Различные формы выдачи справочника создаются при разработке справочника в процессе настройки конфигурации задачи.

Реквизит **Быстрый выбор** используется только для реквизитов типа "Справочник". Если опция включена, то для выбора значения реквизита справоч-

ник будет выдаваться в виде выпадающего списка, а не в отдельном окне. Независимо от количества уровней в выпадающем списке все элементы справочника будут располагаться на одном уровне.

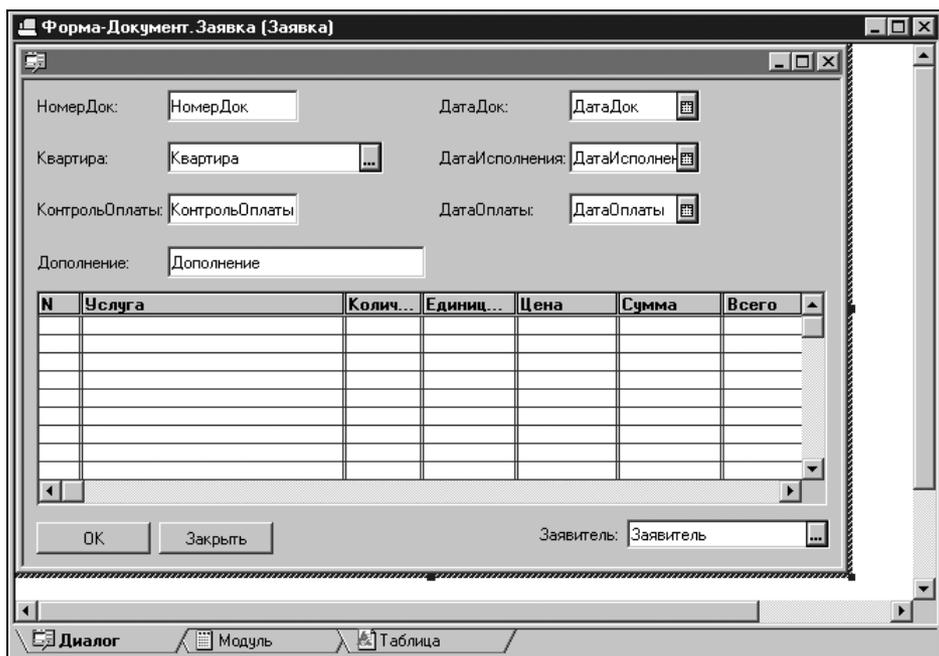


Рис. 3.20. Диалоговое окно формы документа Заявка

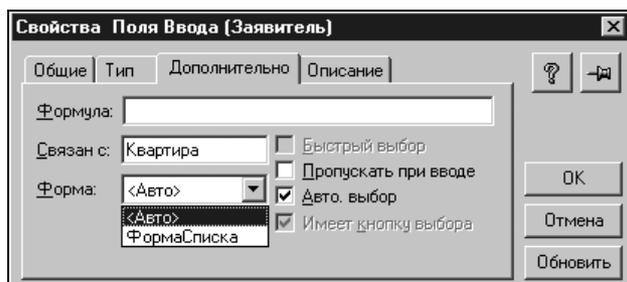


Рис. 3.21. Диалоговое окно Свойства Поля Ввода на вкладке Дополнительно

Реквизит **Пропускать при вводе** задает режим ввода в данный элемент при обходе элементов формы. Если опция включена, то редактируемый элемент диалога будет пропускаться, если переход по элементам диалога выполняется нажатием клавиши <Enter>.

Реквизит **Авто. выбор** используется для реквизитов типа "Документ", "Справочник" и "Перечисление". Если эта опция включена, то при помещении курсора в такой реквизит будет автоматически открываться соответствующий список для выбора значения реквизита без нажатия кнопки выбора.

Реквизит **Имеет кнопку выбора** используется для реквизитов типа "Число" и "Дата". Если опция включена, то при работе с диалогом в поле реквизита появится кнопка, по которой можно вызвать на экран соответственно калькулятор или календарь для ввода значения реквизита.

При создании элемента диалога типа "Переключатель" в справочнике "Квартиры" в окне свойств переключателя использовалось свойство **Первый в группе** (см. рис. 3.8). Для организации выбора с помощью элементов типа "Переключатель" в диалоге должна быть создана группа таких элементов. Один из элементов должен быть назначен начальным элементом группы, это выполняется установкой флажка **Первый в группе**.

На вкладке **Дополнительно** элемента диалога типа "Текст" присутствуют характерные только для текста свойства (рис. 3.22).

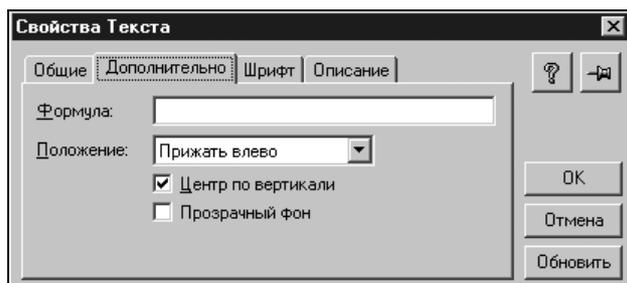


Рис. 3.22. Диалоговое окно **Свойства Текста** на вкладке **Дополнительно**

Реквизит **Положение** управляет выравниванием текста в пределах рамки, окаймляющей элемент диалога (рамка, которая рисуется вокруг выбранного элемента). Можно выбрать один из трех видов выравнивания:

- Прижать влево** — выравнивание текста по левой границе рамки элемента;
- Прижать вправо** — выравнивание текста по правой границе рамки элемента;
- Центр по горизонтали** — размещение текста вдоль вертикальной линии, проходящей через центр рамки элемента;
- Центр по вертикали** — размещает текст вдоль горизонтальной линии, проходящей через центр рамки элемента.

Вкладка **Шрифт**

Вкладка **Шрифт** активизируется только для элементов диалога типа "Текст" и шаблона окна (рис. 3.23).

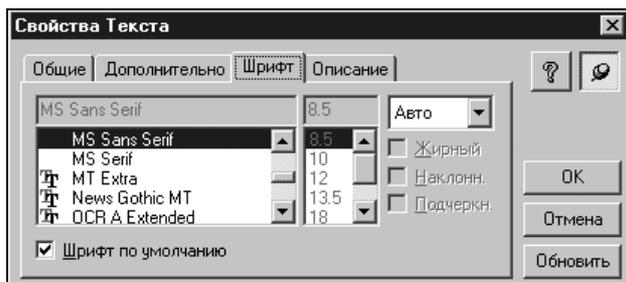


Рис. 3.23. Диалоговое окно свойств элемента диалога для реквизита типа "Текст" на вкладке **Шрифт**

Вкладка **Шрифт** содержит стандартный набор реквизитов для выбора шрифта: списки типов и размеров шрифта, **Жирный**, **Наклонный**, **Подчеркнутый**. Если установлен признак **Шрифт по умолчанию**, то будет использоваться шрифт, установленный в режиме **Параметры**.

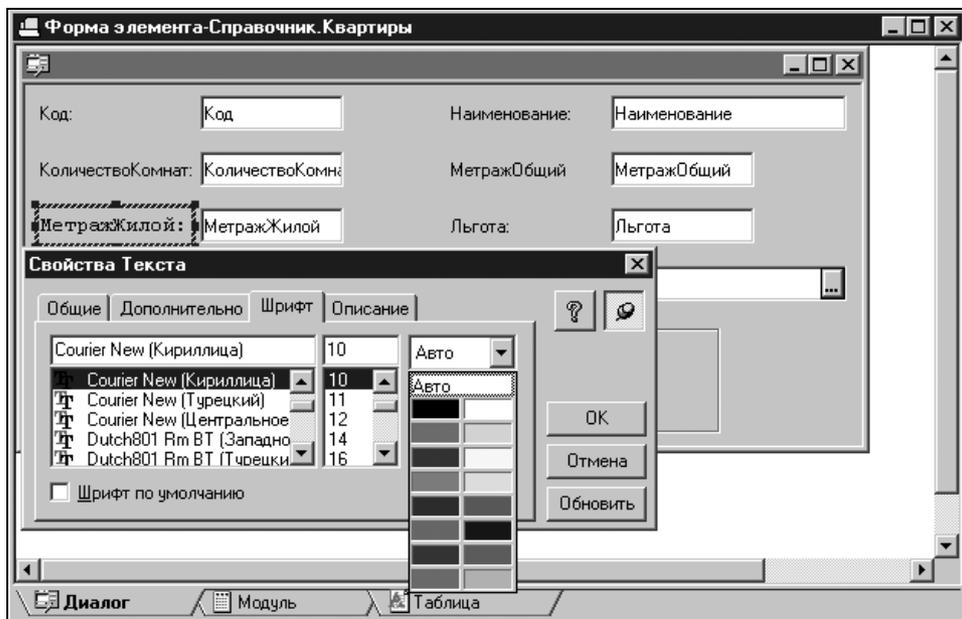


Рис. 3.24. Диалоговое окно свойств элемента диалога для реквизита типа "Текст" на вкладке **Шрифт**

Внимание!

Если признак **Шрифт по умолчанию** включен, редактирование всех свойств шрифта, кроме установки цвета, недоступно. Для задания свойств шрифта в окне свойств необходимо данный признак выключить.

Управляющие элементы вкладки **Шрифт** предназначены для установки гарнитуры, размера и стиля шрифта, который будет использоваться для вывода текста в диалоге (рис. 3.24).

Вкладка *Описание*

Вкладка **Описание** содержит многострочное поле ввода текста описания элемента формы. Реквизит **Подсказка** содержит краткое описание назначения данного элемента диалога. Этот текст будет выдаваться, если при использовании диалога подержать указатель мыши над данным элементом 1—2 секунды.

Реквизит **Использовать описание** устанавливает режим использования текста описания в качестве текста подсказки.

Управляющие элементы этой вкладки позволяют ввести описание, поясняющее смысл и порядок использования данного элемента формы. Текст описания вводится в многострочном поле ввода.

Диалог

Кроме перечисленных элементов, которые образуют сам диалог, существует еще один, который является объединяющим по отношению к остальным. Это шаблон окна — прямоугольная область, на которой располагаются все остальные элементы диалога. Диалог для объекта метаданных создается автоматически после того, как объект метаданных определен в Конфигураторе. В окне диалога могут размещаться реквизиты и управляющие элементы, задаваемые по умолчанию. В дальнейшем при создании диалога на него помещаются реквизиты для ввода информации, поясняющий текст и кнопки. Таблица помещается в окне диалога автоматически, если связанный с диалогом элемент данных имеет табличную часть.

Шаблон окна диалога называется прямоугольная область, в пределах которой должны размещаться все элементы, образующие диалог. Шаблон окна создается для нового диалога, и в дальнейшем только редактируется. По своим свойствам шаблон окна похож на прочие элементы диалога, однако имеется ряд отличий. Прежде всего, шаблон окна нельзя удалить. Также его нельзя перемещать и копировать, т. к. он имеет фиксированное положение в окне редактора диалогов.

Процесс редактирования диалога заключается в размещении в пределах окна диалога элементов описанных ранее типов, задании свойств для них и размещении элементов в окне диалога вручную или автоматически. Для редактирования окна диалога используется мышь или клавиатура.

Вкладка **Общие** окна свойств содержит только один управляющий элемент. В поле **Заголовок** можно ввести текст, который будет выводиться в заголовке окна диалога при его использовании. Если заголовок не указан, будет выводиться название типа данных, связанных с этим диалогом.

Управляющие элементы вкладки **Шрифт** предназначены для установки гарнитуры, размера и стиля шрифта, который будет использоваться для вывода текста в диалоге.

Имеется возможность просмотра разработанного диалога при его использовании. Для перехода к проверке следует выбрать функцию **Проверить** в меню **Диалог**.

На экран будет вызвано окно проверки, имитирующее работу созданного диалога. Для выхода из режима проверки следует закрыть это окно.

Слои

Для облегчения редактирования диалогов, содержащих много элементов, в редакторе диалогов существует механизм слоев. Наиболее распространенный вариант использования механизма слоев — организация картотеки. Для каждой вкладки создается отдельный слой.

В процессе редактирования диалога может быть создано необходимое количество слоев. Элементы диалога при создании помещаются на один из существующих слоев, назначенный "активным слоем". По умолчанию слой называется "основным". Любой слой может быть "спрятан" (сделан невидимым), тогда расположенные на этом слое управляющие элементы не будут мешать при редактировании управляющих элементов диалога, расположенных на других слоях.

Для управления слоями служит диалог **Слои**, который вызывается выбором пункта **Слои** из меню **Диалог** главного меню Конфигуратора.

Добавление слоя

При создании окна справочника "Квартиры" не все реквизиты справочника были размещены в окне. Разместим оставшиеся реквизиты на дополнительных слоях:

- Основной;
- Кнопки;
- ПоказанияСчетчиков.

Создадим новый слой. Для этого необходимо в меню **Диалог** выбрать пункт **Слой** или на панели инструментов **Редактор диалогов** нажать кнопку . Для диалога **Слой** не задано контекстное меню, поэтому необходимо пользоваться кнопками на панели инструментов окна (рис. 3.25).

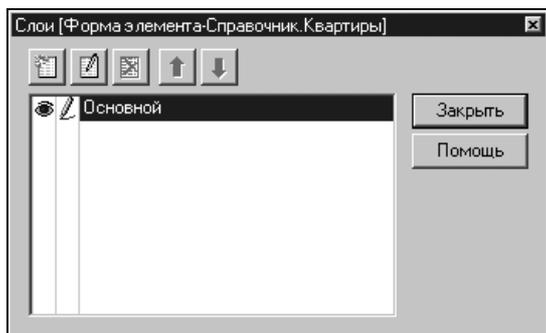


Рис. 3.25. Диалоговое окно **Слой** справочника "Квартиры"

Окно со списком слоев содержит три колонки. В правой колонке выдаются идентификаторы существующих слоев. В средней колонке значком  обозначается активный слой. Этот значок можно поставить двойным щелчком мыши в этой колонке напротив наименования нужного слоя. Он может быть проставлен только у одного из слоев. При постановке значка у какого-либо слоя он автоматически снимается с предыдущего активного слоя. В крайней левой колонке списка значком обозначены видимые слои. Эти значки можно ставить или снимать двойным щелчком мыши в этой колонке напротив наименования нужного слоя. В отличие от признака активного слоя, видимость слоев можно включать произвольно. Однако видимость нельзя снять у активного слоя; а при назначении активным невидимого слоя он автоматически становится видимым.

При нажатии кнопки  предлагается задать идентификатор и опции нового слоя. Назовем новый слой **Кнопки** и оставим признаки включенными (рис. 3.26).

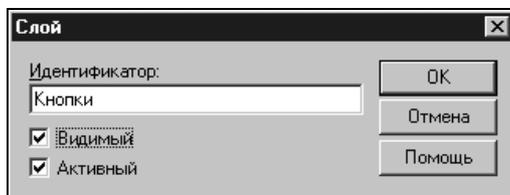


Рис. 3.26. Диалоговое окно **Слой**

При нажатии кнопки **ОК** будет создан новый слой **Кнопки**. Новый слой пустой. Перенесем кнопки с основного слоя на слой **Кнопки**. Для этого двойным щелчком мыши на строке **Основной** вернемся к основному слою. Не закрывая диалога **Слои**, выделим кнопки **ОК** и **Заккрыть**, а также реквизит **Вид-УчетаВоды**, в меню **Действия** выберем пункт **Запомнить** и удалим выделенные кнопки с помощью нажатия клавиши <Delete>. Перейдем к слою **Кнопки**, в меню **Действия** выберем пункт **Вставить**. Новый слой при управлении из модуля формы будет на экране постоянно (рис. 3.27).

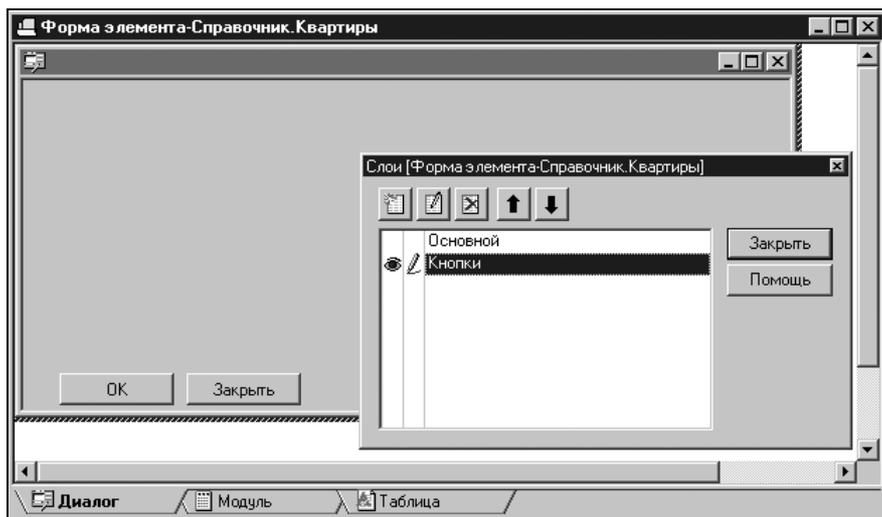


Рис. 3.27. Диалоговое окно слоя **Кнопки** справочника "Квартиры"

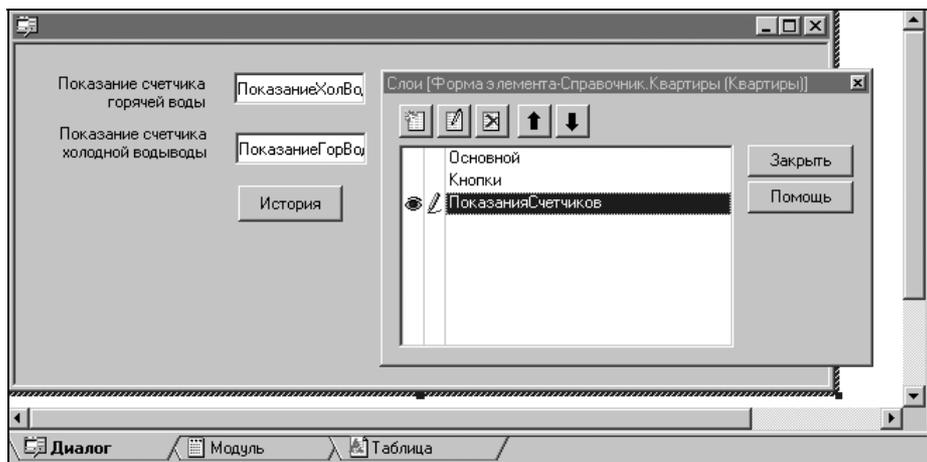


Рис. 3.28. Диалоговое окно слоя **ПоказанияСчетчиков** справочника "Квартиры"

Аналогичным образом создадим еще один слой с идентификатором **ПоказанияСчетчиков**. Разместим в нем реквизиты **ПоказанияХолВоды** и **ПоказанияГорВоды** с помощью пункта **Реквизиты** меню **Вставить** и создадим кнопку **История** (рис. 3.28).

Удаление слоя

При нажатии кнопки  на панели инструментов диалога **Слой** после ответа **Да** на запрос на подтверждение удаления слоя (рис. 3.29) предлагается задать режим удаления слоя в диалоге **Удаление слоя**.

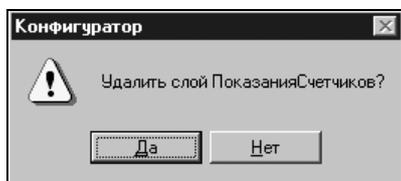


Рис. 3.29. Запрос на подтверждение удаления слоя

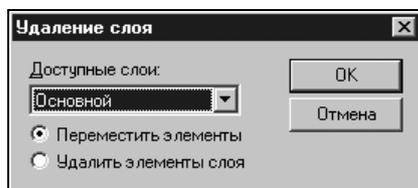


Рис. 3.30. Диалоговое окно Удаление слоя

Диалог **Удаление слоя** позволяет указать, как следует поступить с элементами удаляемого слоя.

Признак **Удалить элементы слоя** устанавливает режим удаления управляющих элементов диалога, расположенных на этом слое, вместе со слоем (рис. 3.30).

Если включен признак **Переместить элементы**, то в списке **Доступные слои** необходимо выбрать наименование одного из слоев. В этом случае управляющие элементы удаляемого слоя будут перемещены на указанный слой.

Программирование управления слоями

Конструирование слоев без программирования управления ими — механизм бесполезный. При создании нового элемента справочника или документа в окне отобразятся все слои формы элемента справочника или документа, что приведет к наложению элементов диалога разных слоев. Для управления видимостью слоев существует механизм закладок, который создается с помощью нескольких методов, задаваемых в модуле формы:

- **Форма.ИспользоватьЗакладки;**
- **Форма.Закладки.ДобавитьЗначение;**
- **Форма.Закладки.УдалитьЗначение(1);**
- **Форма.ИспользоватьСлой().**

С помощью механизма predeterminedенных процедур ПриОткрытии и ПриСмене-Закладки зададим форму элемента справочника "Квартиры" в виде картотеки. При включении признака учета воды по водомеру слой "Кнопки" должен присутствовать на обеих вкладках (рис. 3.31).

Рис. 3.31. Форма элемента справочника "Квартиры" в виде картотеки

В predeterminedенной процедуре ПриОткрытии () определяются закладки, если признак учета воды по водомеру включен:

ПроцедураПриОткрытии ()

Если ВидУчетаВоды = 1 **тогда**

Форма.ИспользоватьЗакладки (1) ;

Форма.Закладки.ДобавитьЗначение ("Общий", "Общие") ;

Форма.Закладки.ДобавитьЗначение ("ПоказанияСчетчиков", "Показания счетчиков") ;

КонецЕсли;

// При открытии используем комбинацию двух слоев: "Основной" и "Кнопки"

Форма.ИспользоватьСлой ("Основной, Кнопки", 2) ;

Если Форма.ТолькоПросмотр () = 1 **Тогда**

Форма.КнопкаИзменить.Доступность (0) ;

Форма.КнопкаДобавить.Доступность (0) ;

Форма.КнопкаУдалить.Доступность (0) ;

КонецЕсли;

КонецПроцедуры

В предопределенной процедуре `ПриВыбореЗакладки()` определяются слои, которые активизируются при выборе той или иной закладки. Параметр `НомерЗакладки` передает номер выбранной закладки:

Процедура `ПриВыбореЗакладки(НомерЗакладки)`

Если `НомерЗакладки = 2` **тогда**

`Форма.ИспользоватьСлой("ПоказанияСчетчиков, Кнопки", 2);`

Иначе

`Форма.ИспользоватьСлой("Основной, Кнопки", 2);`

КонецЕсли;

КонецПроцедуры

Поскольку закладка **Показания счетчиков** нужна только в случае вида учета воды по водомеру, при выключении данного признака закладка должна убираться с экрана. В процедуре `ПоВодомеру()` активизируется или деактивируется закладка **Показания счетчиков**.

Процедура `ПоВодомеру()`

Если `ВидУчетаВоды = 1` **тогда**

`Форма.ИспользоватьЗакладки(1);`

`Форма.Закладки.ДобавитьЗначение("Общий", "Общие");`

`Форма.Закладки.ДобавитьЗначение("ПоказанияСчетчиков", "Показания
счетчиков");`

Иначе

`Форма.Закладки.УдалитьЗначение(1);`

`Форма.Закладки.УдалитьЗначение(1);`

`Форма.ИспользоватьЗакладки(0);`

КонецЕсли;

`Форма.ИспользоватьСлой("Основной, Кнопки", 2);`

КонецПроцедуры

Осталось назначить данную процедуру свойству **Формула** реквизита **ВидУчетаВоды** (рис. 3.32).

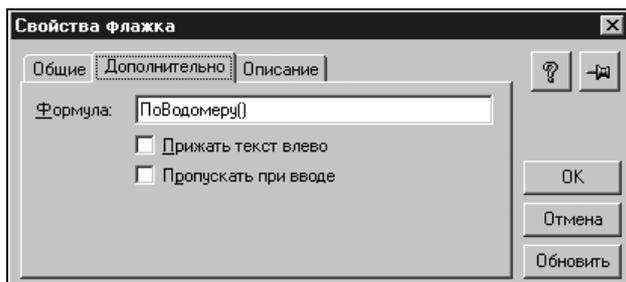


Рис. 3.32. Диалоговое окно свойств реквизита **ВидУчетаВоды** на вкладке **Дополнительно**

При включении признака учета воды по водомеру выполняется процедура `ПоВодомеру()`, которая делает активной закладку **Показания счетчиков**. При выключении данного признака эта же процедура скроет закладку.

Синтаксис и конструкции встроенного языка

Несмотря на кажущуюся простоту составления процедур на встроенном языке, существует определенный порядок составления модулей и использования операторов.

В форму списка созданного справочника "Квартиры" добавим кнопки **Лицевой счет** с процедурой обработки `ПечатьЛицевогоСчета(1)` и **Состав семьи** с процедурой обработки `ПечатьЛицевогоСчета(0)` (рис. 3.33).

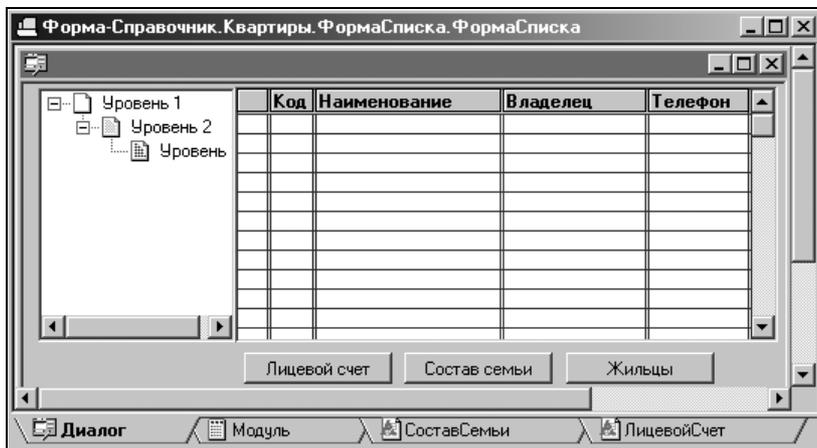


Рис. 3.33. Диалоговое окно формы списка справочника "Квартиры"

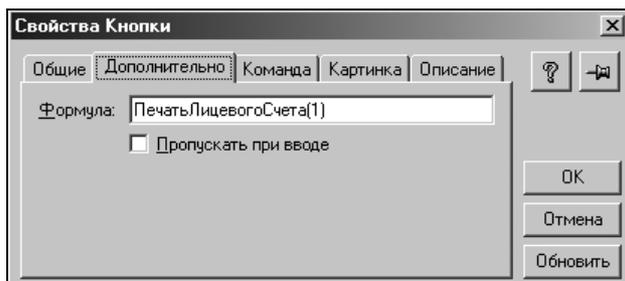


Рис. 3.34. Диалоговое окно **Свойства Кнопки**

Для начала необходимо определить место и правила создания соответствующей процедуры обработки нажатия кнопки. Имя процедуры обработки события в данном случае — нажатие кнопки, задается на вкладке **Дополнительно** элемента интерфейса в поле **Формула** (рис. 3.34).

Процедура

На вкладке **Модуль** формы списка справочника создадим процедуру, руководствуясь следующими правилами:

- Операторы внутри тела модуля должны заканчиваться символом ";" за исключением структурных:
 - Процедура;
 - КонечПроцедуры;
 - Функция;
 - КонечФункции;
 - Если ... тогда;
 - Иначе;
 - ИначеЕсли ... тогда;
 - Пока ... цикл;
 - Для ... цикл;
 - Попытка;
 - Исключение.
- Наличие пробелов между словами в строках не регламентировано.
- Комментарии в тексте выделяются любым количеством символов "/", но не меньше двух.
- Объявления глобальных переменных должны находиться в начале тела модуля до определения первой процедуры или функции.
- Объявления локальных переменных должны находиться внутри процедуры до первого оператора.
- Инициализация глобальных переменных должна находиться в конце тела модуля после определения всех функций и процедур.
- Первой строкой процедуры должно быть ключевое слово **Процедура**, после которого через любое количество пробелов следует имя процедуры. Ключевое слово **Процедура** начинает секцию исходного текста, выполнение которого можно инициировать из любой точки программного модуля (в данном случае по нажатию кнопки), просто указав имя процедуры со списком параметров (если параметры не передаются, то круглые скобки в объявлении имени обязательны).

Внимание!

Все ключевые слова выделяются в редакторе текста Конфигуратора красным цветом. Если ключевое слово, которое указано курсивом, не стало красного цвета, это значит, что вкралась синтаксическая ошибка.

- После списка параметров может следовать ключевое слово **Экспорт**, которое указывает на то, что данная процедура является доступной из других программных модулей (это имеет смысл только в глобальном программном модуле).
- Если в процедуре существует одна или несколько точек возврата в строку программы, из которой было обращение к процедуре, используется оператор **Возврат**, который завершает выполнение процедуры и осуществляет этот возврат. Использование данного оператора в процедуре не обязательно.
- Последняя строка процедуры определяет конец программной секции процедуры ключевым словом **КонецПроцедуры**.

Список параметров

Элементы списка параметров разделяются запятыми. Значения формальных параметров должны соответствовать значениям передаваемых при вызове процедуры фактических параметров. В этом списке определяются имена каждого из параметров так, как они используются в тексте процедуры. Список формальных параметров может быть пуст.

Список параметров может содержать:

- *Ключевое слово **Знач.*** — необязательное ключевое слово, которое указывает на то, что следующий за ним параметр передается по значению, т. е. изменение значения формального параметра при выполнении процедуры никак не повлияет на фактический параметр, переданный при вызове процедуры.
- *Имя формального параметра* — после него может следовать знак "=" и значение параметра по умолчанию. Параметры с установленными значениями по умолчанию можно располагать в любом месте списка формальных параметров. Если параметр при вызове процедуры опущен, то он принимает либо установленное по умолчанию значение (если оно есть), либо принимает "пустое" значение (значение неопределенного типа). При задании параметров необходимо выполнить следующие условия:

Если параметру не задано значение по умолчанию, и он является последним в списке передаваемых параметров, то при вызове процедуры его нельзя опускать. Если параметру не задано значение по умолчанию, то при вызове процедуры его можно опускать в списке передаваемых фактических параметров, но разделительную запятую надо ставить.

Если параметру задано значение по умолчанию, и он является последним в списке, то при вызове процедуры его можно опускать в списке переда-

ваемых фактических параметров и не ставить запятую перед опущенным параметром.

Внимание!

Параметр процедуры передается по ссылке, т. е. изменение внутри процедуры значения формального параметра приведет к изменению значения соответствующего фактического параметра.

Предварительное объявление процедуры

Если в программном модуле задается вызов процедуры, определение которой находится после определения вызывающей процедуры, выдается сообщение об ошибке. В тексте программного модуля допускается предварительное описание процедур и функций без их определения. На то, что это описание предварительное, указывает наличие ключевого слова **Далее**, которое замещает тело процедуры, и ключевое слово **КонецПроцедуры**. Предварительное описание процедуры/функции может содержаться в любом месте текста модуля, где допускается фактическое определение процедуры или функции, а сам заголовок процедуры/функции должен в точности соответствовать заголовку в фактическом определении, включая наличие, если необходимо, ключевого слова **Экспорт** и имен формальных параметров.

Процедура ПечатьЛицевогоСчета (Режим) формирования отчетов "Выписка из лицевого счета" и "Справка о составе семьи" будет выглядеть так:

```
Процедура ПечатьЛицевогоСчета (Режим)
    Таб=СоздатьОбъект ("Таблица");
    Если Режим = 0 тогда
        Таб.ИсходнаяТаблица ("СоставСемьи");
    Иначе
        Таб.ИсходнаяТаблица ("ЛицевойСчет");
    КонецЕсли;
    Таб.ВывестиСекцию ("Шапка");
    Жильцы = СоздатьОбъект ("Справочник.Жильцы");
    Жильцы.ИспользоватьВладельца (ТекущийЭлемент ());
    Жильцы.ВыбратьЭлементы ();
    Пока Жильцы.ПолучитьЭлемент ()=1 цикл
        Таб.ВывестиСекцию ("Строка");
    КонецЦикла;
    Таб.Опции (0, 0, 0, 0, "ОпцииСправки");
    Таб.ТолькоПросмотр (1);
    Если Режим = 0 тогда
        Таб.Показать ("Справка о составе семьи", "");
```

Иначе

Таб.Показать ("Выписка из лицевого счета", "");

КонецЕсли;

КонецПроцедуры

В данном фрагменте список параметров состоит из одного параметра `Режим`, значение которого задается в свойствах кнопок.

Функция

Определение функции в программном модуле совпадает с определением процедуры с одной существенной разницей: выполнение функции заканчивается обязательным оператором `Возврат`. Функции отличаются от процедур только тем, что возвращают `ВозвращаемоеЗначение`.

`ВозвращаемоеЗначение` — выражение, значение которого содержит результат обращения к функции.

Первой строкой функции должно быть ключевое слово `функция`. Последняя строка процедуры определяет конец программной секции процедуры ключевым словом `Конецфункции`.

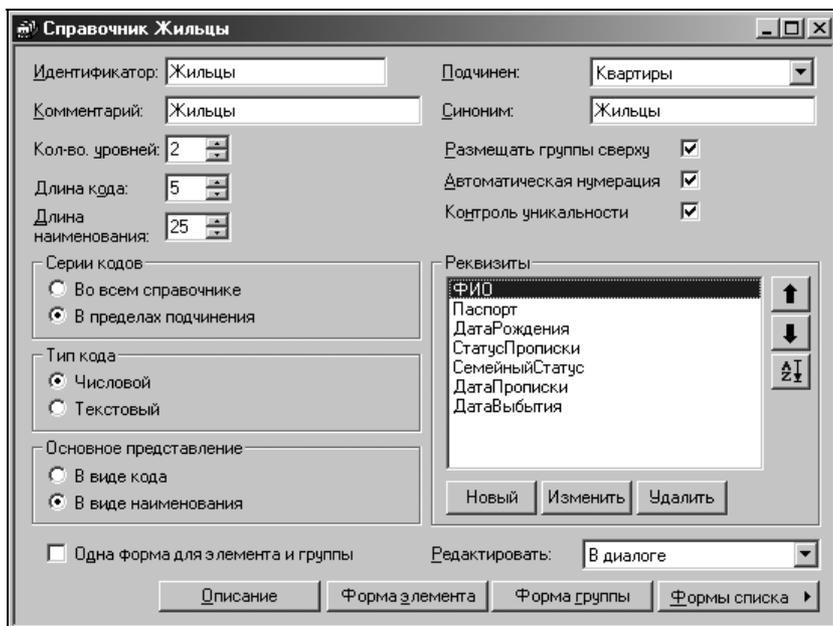


Рис. 3.35. Диалоговое окно Справочник Жильцы

Определим функцию формирования колонки **Владелец** в справочнике "Квартиры" (рис. 3.35). Данного реквизита нет непосредственно в справочнике, но в подчиненном справочнике "Жильцы" у элемента есть реквизит **СтатусПрописки**, который определяет владельца (рис. 3.36).

Реквизит **СтатусПрописки** имеет тип **Перечисление.Статусы**.

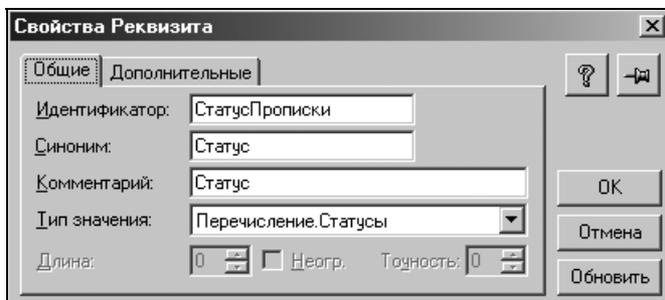


Рис. 3.36. Диалоговое окно свойств реквизита **СтатусПрописки**

Одно из значений перечисления **Статусы** определяет владельца квартиры (рис. 3.37).

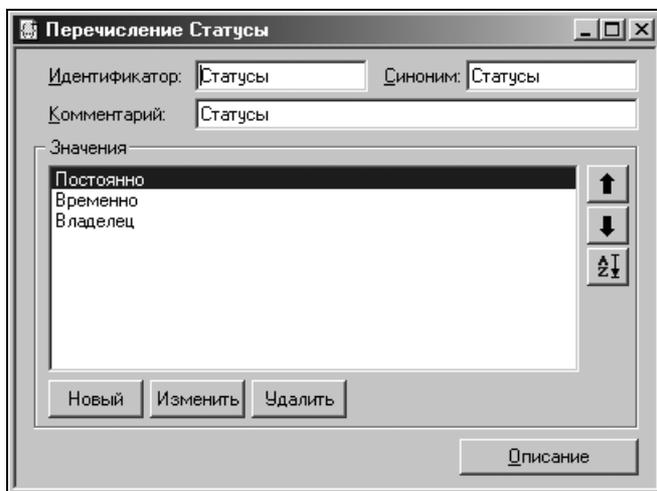


Рис. 3.37. Диалоговое окно **Перечисление Статусы**

На вкладке **Дополнительно** свойств текста **Владелец** зададим функцию `определениеВладельца()` (рис. 3.38). Выражением `ВозвращаемоеЗначение` должна быть строка текста с ФИО владельца квартиры.

Определение данной функции будет выглядеть так:

Функция ОпределениеВладельца ()

Жильцы = СоздатьОбъект ("Справочник.Жильцы") ;

Жильцы.ИспользоватьВладельца (ТекущийЭлемент ()) ;

Жильцы.ВыбратьЭлементы () ;

Пока Жильцы.ПолучитьЭлемент () = 1 **цикл**

Если Жильцы.СтатусПрописки = Перечисление.Статусы.Владелец **тогда**

Возврат Жильцы.ФИО

КонецЕсли ;

КонецЦикла ;

КонецФункции

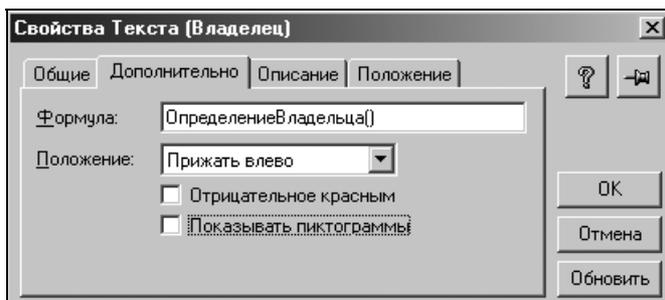


Рис. 3.38. Диалоговое окно свойств текста Владелец

Оператор условного выполнения

Оператор задания алгоритма выполнения имеет разнообразные формы от самого простого выполнения варианта условия до сложной многовариантной конструкции. Самый простой вариант оператора условного выполнения:

Если <Логическое_выражение> **Тогда**

<операторы>

КонецЕсли ;

<Логическое выражение> — любая комбинация операндов логического типа в сочетании с логическими операциями И, ИЛИ и НЕ. В тексте функции ОпределениеВладельца () присутствует простой вариант оператора условного выполнения:

Если Жильцы.СтатусПрописки = Перечисление.Статусы.Владелец **тогда**

Возврат Жильцы.ФИО

КонецЕсли ;

Оператор условного выполнения, в котором задается последовательность не только при выполнении определенного условия, но и последовательность операторов в случае невыполнения этого условия имеет следующий вид:

```
Если <Логическое_выражение> Тогда
    <Последовательность операторов по "истина">
Иначе
    <Последовательность операторов по "ложь">
КонецЕсли;
```

Ключевое слово **Иначе** обозначает начало последовательности операторов в случае невыполнения заданного условия. В приведенном тексте процедуры ПечатьЛицевогоСчета (Режим) присутствует данная конструкция:

```
Если Режим = 0 тогда
    Таб.ИсходнаяТаблица ("СоставСемьи");
Иначе
    Таб.ИсходнаяТаблица ("ЛицевойСчет");
КонецЕсли;
```

Более сложный оператор условного выполнения с заданием многих последовательностей позволяет вести анализ вариантов условия. Данный оператор является объединением последовательности простых операторов условного выполнения:

```
Если <Логическое_выражение_1> Тогда
    < Последовательность операторов по "истина"1>
ИначеЕсли <Логическое_выражение_2> Тогда
    < Последовательность операторов по "истина" 2]
...
[ИначеЕсли <Логическое_выражение_к> Тогда
    < Последовательность операторов по "истина" к]>
[Иначе
    < Последовательность операторов по "ложь" по всем>]
КонецЕсли;
```

Объединим три кнопки печати формирования отчетов **Лицевой счет**, **Состав семьи** и **Жильцы** в одну кнопку с выбором (рис. 3.39).

Кнопка с иконкой  задает выбор из списка печатных форм процедурой `ПопКнопкеВыборПечатнойФормы()` и сохраняет название выбранной формы в названии кнопки **Печать**. Объявим переменную `НазваниеТекущейФормы` глобальной и определим процедуры выполнения по нажатию кнопок  и **Печать**.

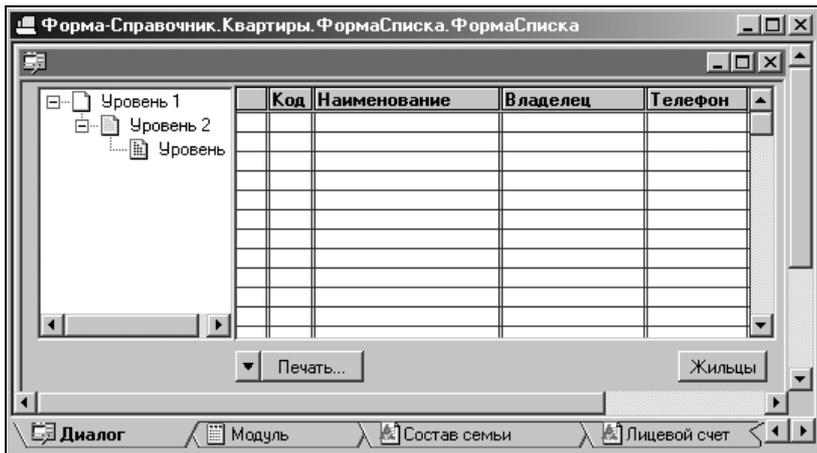


Рис. 3.39. Диалоговое окно формы списка справочника "Квартиры" с кнопкой выбора печатной формы

```
// Объявление глобальной переменной для модуля формы
```

```
Перем НазваниеТекущейФормы;
```

```
// Определение процедуры по нажатию на кнопку Печать
```

```
Процедура ПоКнопкеПечать ()
```

```
    Если ПустоеЗначение (НазваниеТекущейФормы) = 1 Тогда  
        НазваниеТекущейФормы = "Состав семьи";
```

```
    КонецЕсли;
```

```
    Форма.КнопкаПечать.Заголовок (НазваниеТекущейФормы) ;
```

```
    ПечатьЛицевогоСчета (НазваниеТекущейФормы) ;
```

```
КонецПроцедуры // ПоКнопкеПечать
```

```
//*****
```

```
// Определение процедуры выбора отчета
```

```
Процедура ПоКнопкеВыборПечатнойФормы ()
```

```
    СписокПечФорм = СоздатьОбъект ("СписокЗначений") ;
```

```
    СписокПечФорм.ДобавитьЗначение ("Состав семьи") ;
```

```
    СписокПечФорм.ДобавитьЗначение ("Лицевой счет") ;
```

```
    СписокПечФорм.ДобавитьЗначение ("Жильцы") ;
```

```
    Выбор = СписокПечФорм.ВыбратьЗначение (НазваниеТекущейФормы, , 5, 0) ;
```

```
    Если Выбор = 1 Тогда // Если выбор сделан
```

```
        ПоКнопкеПечать () ;
```

```
    ИначеЕсли Выбор = -1 тогда // Если истек тайм-аут
```

```
        Сообщить ("Выбор формы по умолчанию") ;
```

```
        ПоКнопкеПечать () ;
```

Иначе

Форма.КнопкаПечать.Заголовок("Печать..."); // По клавише Esc

КонецЕсли;

КонецПроцедуры // ПоКнопкеВыборПечатнойФормы

В процедуре присутствует рассматриваемая конструкция при анализе действия пользователя:

- был сделан выбор формы;
- истек тайм-аут;
- нажата клавиша <Esc>.

Подобная конструкция присутствует и в процедуре ОткрытьЖильцы(), которая выполняется при нажатии на кнопку **Жильцы**:

// Определение процедуры вызова подчиненного справочника "Жильцы"

Процедура ОткрытьЖильцы()

Если ТекущийЭлемент().Выбран() = 0 **Тогда**

Предупреждение("Нельзя открыть список жильцов, не указав квартиру.
|Введите Квартиру.");

ИначеЕсли ТекущийЭлемент().ЭтоГруппа() = 1 **Тогда**

Предупреждение("Нельзя открыть список жильцов для группы Квартир.
|Выберите конкретную Квартиру.");

Иначе

ОткрытьФорму("Справочник.Жильцы");

КонецЕсли;

КонецПроцедуры // ОткрытьЖильцы

В данном случае ведется анализ положения указателя в справочнике "Квартиры":

- справочник пустой;
- указатель установлен на группе;
- указатель установлен на конкретной квартире.

Оператор цикла

Многokrатно выполняемая последовательность операторов объединяется в конструкцию цикла. Цикл с предусловием выполняется до тех пор, пока выполняется заданное условие. Например:

Пока <Логическое_выражение> **Цикл**

<Последовательность операторов>

КонецЦикла;

"Логическое выражение" задает условие выполнения.

В процедуре ПечатьЛицевогоСчета (Имя) присутствует данная конструкция:

```
Пока Жильцы.ПолучитьЭлемент ()=1 Цикл
```

```
Таб.ВывестиСекцию ("Строка") ;
```

```
КонецЦикла ;
```

В приведенном фрагменте из подчиненного справочника "Жильцы" выбираются элементы и выводятся в отчет до тех пор, пока функция ПолучитьЭлемент () не вернет 0, означающий, что достигнут конец справочника.

Цикл с шагом повторяется заданное счетчиком количество раз:

```
Для <Имя_переменной> = <Выражение1> По <Выражение2> Цикл
    <Последовательность операторов>
```

```
КонецЦикла ;
```

Здесь:

- Имя_переменной — идентификатор переменной (счетчик цикла);
- <Выражение1> — начальное значение счетчика цикла;
- <Выражение2> — конечное значение счетчика цикла;
- <операторы> — последовательность исполняемых операторов.

Обычно оператор используется при работе со списками значений и с таблицами. В этих объектах метаданных, как правило, на определенный момент можно получить количество строк в списке или таблице.

Оператор Попытка...Исключение

В процессе выполнения программы возникают ситуации, которые невозможно обработать обычным анализом с помощью операторов условного выполнения. Для управления такими ошибочными (исключительными) ситуациями предназначен оператор Попытка...Исключение. В качестве ошибочных (исключительных) ситуаций рассматриваются ошибки времени выполнения модуля. Для диагностики ошибки обычно используется системная функция ОписаниеОшибки (), которая возвращает описание ошибки времени выполнения модуля, такое же, как и в окне сообщений.

Синтаксис оператора:

Попытка

```
<Последовательность операторов>
```

Исключение

```
<Обработка исключительной ситуации>
```

КонецПопытки ;

Функция ОписаниеОшибки() предназначена для использования в группе <Обработка исключительной ситуации>.

Самый простой пример использования данной конструкции является обработка возможной ошибки выполнения при загрузке данных из файла, например, из конфигурации "Зарплата и кадры":

Попытка

Правила.Загрузить (ФормИмяФайлаПравил);

Исключение

Предупреждение (ОписаниеОшибки());

Возврат (0);

КонецПопытки;

В качестве более сложного примера с использованием функций НачатьТранзакцию() и ЗафиксироватьТранзакцию рассмотрим фрагмент из обработки ЗаписьПериодическихРеквизитов. Эта обработка вызывается в предопределенной процедуре формы элемента справочника **ПриЗаписи()**. В нашем справочнике "Квартиры" реквизит **Льгота** периодический, поэтому вставим в предопределенную процедуру ПриЗаписи() вызов общей функции для всех справочников глБухЗаписьПериодическихРеквизитов, которая находится в **Глобальном модуле**. В качестве параметра передадим контекст формы:

// Предопределенная процедура.

Процедура ПриЗаписи()

СтатусВозврата (глБухЗаписьПериодическихРеквизитов (Контекст));

КонецПроцедуры // ПриЗаписи()

Запись периодических реквизитов требует комбинации действий по удалению и записи сразу нескольких записей в базу данных. Во время выполнения этих действий может возникнуть исключительная ситуация.

Процедура ЗаписатьЗначения()

Периодический = СоздатьОбъект ("Периодический");

Попытка

НачатьТранзакцию();

// При первой записи элемента системой принудительно записываются первые значения периодических реквизитов – их надо удалить.

ПерваяЗапись=ПустоеЗначение (КонтекстФормы.ТекущийЭлемент());

КонтекстФормы.Записать();

ТекущийЭлемент = КонтекстФормы.ТекущийЭлемент();

Если ПерваяЗапись = 1 **Тогда**

Для Номер = 1

По Метаданные.Справочник(КонтекстФормы.Вид()) .Реквизит() **Цикл**

мдРеквизит = Метаданные.Справочник(КонтекстФормы.Вид()) .Реквизит(Номер);

Если мдРеквизит.Периодический = 0 **Тогда**

Продолжить;

КонецЕсли;

Если (ТекущийЭлемент.ЭтоГруппа() = 0) И (мдРеквизит.Использование = "ДляГруппы") **Тогда**

Продолжить;

КонецЕсли;

Если (ТекущийЭлемент.Этогруппа() = 1) И (мдРеквизит.Использование = "ДляЭлемента") **Тогда**

Продолжить;

КонецЕсли;

Периодический.ИспользоватьОбъект(мдРеквизит.Идентификатор, ТекущийЭлемент);

Периодический.ВыбратьЗначения();

Пока Периодический.ПолучитьЗначение() = 1 **Цикл**

Периодический.Удалить();

КонецЦикла;

КонецЦикла;

КонецЕсли;

Значения.ВыбратьСтроки();

Пока Значения.ПолучитьСтроку() = 1 **Цикл**

Если Значения.Пометка <> 2 **Тогда** // Пометка выключена

Продолжить;

КонецЕсли;

Если Периодический.ИспользоватьОбъект(Значения.Идентификатор, ТекущийЭлемент) = 1 **Тогда**

ТипВид = ТипЗначенияСтр(Значения.НовоеЗначение);

Длина = 0;

Точность = 0;

Если (ТипВид = "Число") **Тогда**

Длина = СтрДлина(Значения.НовоеЗначение);

Точность = Длина-Мин(Длина, Найти(Строка(Значения.НовоеЗначение)+".", "."));

ИначеЕсли (ТипВид = "Строка") **Тогда**

Длина = СтрДлина(Значения.НовоеЗначение);

ИначеЕсли (ТипВид = "Справочник") **ИЛИ** (ТипВид = "Документ")

ИЛИ (ТипВид = "Перечисление") **ИЛИ** (ТипВид = "Счет") **ИЛИ**

(ТипВид = "Календарь") **Тогда**

ТипВид = ТипВид+"."+Значения.НовоеЗначение.Вид();

КонецЕсли;

Периодический.НазначитьТип(ТипВид, Длина, Точность);

```

Периодический.Значение = Значения.НовоеЗначение;
Если (Значения.СтараяДатаЗнач = '00.00.0000') и (ПерваяДата
<> '00.00.0000') Тогда

```

// Первые значения периодических реквизитов в некоторых случаях следует записывать на заведомо давнюю дату.

```

    Периодический.ДатаЗнач = ПерваяДата;

```

```

Иначе

```

```

    Периодический.ДатаЗнач = НоваяДата;

```

```

КонецЕсли;

```

```

    Периодический.Записать ();

```

```

КонецЕсли;

```

```

КонецЦикла;

```

```

ЗафиксироватьТранзакцию ();

```

```

// Признак успешной записи.

```

```

Форма.Параметр = 1;

```

```

Форма.Закрыть ();

```

```

Исключение

```

```

    Сообщить (ОписаниеОшибки (), "!");

```

```

КонецПопытки;

```

```

КонецПроцедуры // ЗаписатьЗначения ()

```

Как видно из фрагмента, в случае возникновения исключительной операции транзакция не фиксируется.

Оператор *ВызватьИсключение*

Данный оператор применяется в языковой конструкции `ПопыткаИсключение... КонецПопытки`. Обработка исключительной ситуации может быть вложенной. При этом управление будет передано на самый последний по вложению обработчик, начинающийся с ключевого слова **Исключение**. Последовательность операторов обработки исключительной ситуации может содержать оператор `ВызватьИсключение`. Выполнение данного оператора прекращает выполнение последовательности обработки исключительной ситуации и производит поиск "внешнего" обработчика. Если таковой есть, то управление передается на его первый оператор. Если нет, то выполнение модуля прекращается с выдачей сообщения о первоначально возникшей ошибке. Оператор `ВызватьИсключение` может встречаться только внутри операторных скобок `Исключение ... КонецПопытки`.

Оператор *Перейти*

Использование оператора безусловной передачи управления на исполняемый оператор программного блока считается признаком "плохого тона" в объектно-ориентированном программировании.

Синтаксис:

Перейти <Метка>;

Однако оператор в редких случаях еще используется. Например, в отчете ПодготовкаСведенийДляПФР:

```

Процедура ЗаписатьИПоказатьТекст (ВыходнойТекст, ИмяФайла, ЗаголовокОкна)
    ФайлЗаписан = 1;
Если Найти ("А, В", Лев (ИмяФайла, 1)) > 0 Тогда
    ПутьВывода = Лев (ИмяФайла, 2);
    ТекстВопроса = "Поставьте чистую дискету для файла пачки
    "+Сред (ЗаголовокОкна, Найти (ЗаголовокОкна, "№"));
    ~Начало:Хорошо=0;
Если Вопрос (ТекстВопроса, "ОК+Отмена")="ОК" Тогда
    Если ФС.СуществуетФайл (ПутьВывода+"\NUL")=0 Тогда
        ТекстВопроса = "Поставьте чистую дискету для файла пачки
        "+Сред (ЗаголовокОкна, Найти (ЗаголовокОкна, "№"));
        Перейти ~Начало;
    КонецЕсли;
Если ФС.СуществуетФайл (ИмяФайла)=1 Тогда
        ФС.УдалитьФайл (ИмяФайла)
    КонецЕсли;
Если ПустаяСтрока (ФС.НайтиПервыйФайл (ПутьВывода + "\" +
    Прав (СокрЛП (глПолучитьНалог ("ПФР_страх").РегНомер), 6)
    +"??.*")) = 0 Тогда
        ТекстВопроса = "На дискете присутствует файл, относящийся
        к другой пачке!"+"РазделительСтрок"+"Поставьте чистую дискету
        для файла пачки
        "+Сред (ЗаголовокОкна, Найти (ЗаголовокОкна, "№"));
        Перейти ~Начало;
    КонецЕсли;
Иначе
        ФайлЗаписан = 0;
    КонецЕсли;
КонецЕсли;
Если ФайлЗаписан = 1 Тогда
        ВыходнойТекст.Записать (ИмяФайла);
    Если ФС.СуществуетФайл (ИмяФайла) = 0 Тогда
        Предупреждение ("Не возможен вывод файла данных для
        передачи в ПФР в каталог "+
        Лев (ИмяФайла, Найти (ИмяФайла, "\" ) - 1) + "!");
        ФайлЗаписан = 0;
    КонецЕсли;
КонецЕсли;

```

```

Если ФайлЗаписан = 1 Тогда
    ТекстДляПоказа = СоздатьОбъект ("Текст");
    ТекстДляПоказа.КодоваяСтраница (1);
    ТекстДляПоказа.Открыть (ИмяФайла);
    ТекстДляПоказа.Показать (ЗаголовокОкна);

Иначе
    ВыходнойТекст.Показать (ЗаголовокОкна, Прав (ИмяФайла, 12));

КонецЕсли;

```

```

КонецПроцедуры // ЗаписатьИПоказатьТекст ()

```

Оператор *Продолжить*

Оператор передачи управления в начало цикла используется, когда необходимо прервать выполнение шага цикла и перейти к началу следующего шага. Рассмотрим процедуру формирования лицевого счета ПечатьЛицевогоСчета (Имя). Добавим анализ помеченных на удаление элементов:

```

Пока Жильцы.ПолучитьЭлемент () = 1 цикл
    Если Жильцы.ПометкаУдаления () = 1 тогда
        Продолжить;
КонецЕсли;
    Таб.ВывестиСекцию ("Строка");
КонецЦикла;

```

Если элемент справочника помечен на удаление, строка данного элемента в отчет выведена не будет.

Оператор *Прервать*

Оператор принудительного завершения выполнения цикла используется, когда необходимо прервать выполнение всего цикла.

Оператор *Возврат*

Синтаксис оператора:

```

Возврат [<Выражение>]

```

Оператор завершения процедуры или функции используется в случае, когда по данной ветке процедуры или функции достигнут логический конец. По данному оператору происходит возврат в точку вызова процедуры или функции. <Выражение> после ключевого слова **Возврат** представляет собой значение, возвращаемое функцией.

Внимание!

<Выражение> имеет смысл только для функции.

Рассмотрим функцию `ОпределениеВладельца()`, которая была приведена в разделе "Функция".

Оператором

Возврат Жильцы.ФИО

в графу **Владелец** возвращается текущий элемент справочника "Жильцы".

Директивы

Директива переключения загрузки программного модуля на загрузку из текстового файла `ЗагрузитьИзФайла` используется в тех случаях, когда обновление конфигурации непосредственно через Конфигуратор затруднительно. В таких случаях достаточно изменить содержимое текстового файла.

Синтаксис оператора:

`#ЗагрузитьИзФайла <ИмяФайла>`

где `<ИмяФайла>` — имя файла, содержащего исходный текст программного модуля (записывается без кавычек и скобок).

Внимание!

Конструкция `#ЗагрузитьИзФайла` должна записываться в первой строке программного модуля с первой позиции.

Создадим текстовый файл в корневом каталоге диска C: с именем `Загрузка.txt`. Скопируем содержимое тела модуля формы списка справочника "Квартиры":

```
// Определение процедуры по нажатию на кнопку Печать
```

```
Процедура ПоКнопкеПечать ()
```

```
    Если ПустоеЗначение (НазваниеТекущейФормы) = 1 Тогда
        НазваниеТекущейФормы = "Состав семьи";
```

```
    КонецЕсли;
```

```
    Форма.КнопкаПечать.Заголовок (НазваниеТекущейФормы) ;
```

```
    ПечатьЛицевогоСчета (НазваниеТекущейФормы) ;
```

```
КонецПроцедуры //ПоКнопкеПечать
```

```
//*****
```

```
// Определение процедуры выбора отчета
```

```
Процедура ПоКнопкеВыборПечатнойФормы ()
```

```
    СписокПечФорм = СоздатьОбъект ("СписокЗначений") ;
```

```

СписокПечФорм.ДобавитьЗначение ("Состав семьи");
СписокПечФорм.ДобавитьЗначение ("Лицевой счет");
СписокПечФорм.ДобавитьЗначение ("Жильцы");
Выбор = СписокПечФорм.ВыбратьЗначение (НазваниеТекущейФормы, ,, 5, 0);
Если Выбор = 1 Тогда // Если выбор сделан
    ПоКнопкеПечать ();
ИначеЕсли Выбор = -1 тогда
    Сообщить ("Выбор формы по умолчанию"); // Если истек тайм-аут
    ПоКнопкеПечать ();
Иначе
    Форма.КнопкаПечать.Заголовок ("Печать..."); // По клавише Esc
КонецЕсли;
КонецПроцедуры //ПоКнопкеВыборПечатнойФормы

```

```

//*****
// Определение прцедуры формирования отчета
Процедура ПечатьЛицевогоСчета (Имя)
    Таб=СоздатьОбъект ("Таблица");
    Таб.ИсходнаяТаблица (Имя);
    Таб.ВывестиСекцию ("Шапка");
    Жильцы = СоздатьОбъект ("Справочник.Жильцы");
    Жильцы.ИспользоватьВладельца (ТекущийЭлемент ());
    Жильцы.ВыбратьЭлементы ();
Пока Жильцы.ПолучитьЭлемент ()=1 цикл
    Если Жильцы.ПометкаУдаления () = 1 тогда
        Продолжить;
    КонецЕсли;
    Таб.ВывестиСекцию ("Строка");
КонецЦикла;
    Таб.Опции (0, 0, 0, 0, "ОпцииСправки");
    Таб.ТолькоПросмотр (1);
    Таб.Показать (Имя, "");
КонецПроцедуры //ПечатьЛицевогоСчета

```

```

//*****
// Определение процедуры вызова окна подчиненного справочника "Жильцы"
Процедура ОткрытьЖильцы ()
Если ТекущийЭлемент ().Выбран () = 0 Тогда
    Предупреждение ("Нельзя открыть список жильцов, не указав квартиру.
        |Введите Квартиру.");
ИначеЕсли ТекущийЭлемент ().ЭтоГруппа () = 1 Тогда
    Предупреждение ("Нельзя открыть список жильцов для группы Квартир.
        |Выберите конкретную Квартиру.");

```

Иначе

ОткрытьФорму ("Справочник.Жильцы");

КонецЕсли;

КонецПроцедуры // ОткрытьЖильцы

В модуле формы списка справочника "Квартиры" зададим директиву загрузки текста из файла:

```
#ЗагрузитьИзФайла С:\Загрузка.txt
```

В случае отсутствия заданного файла пользователю будет выдано сообщение:

Не могу открыть файл: <имя файла>

Алгоритм исполнения модулей встроенного языка

В Конфигураторе в задании большинства объектов метаданных в описании форм присутствует вкладка **Модуль**, а для документов еще и вкладка **Модуль Документа**. В каком же порядке исполняется набор процедур в модулях?

При нажатии пользователем кнопки, вводе данных в поле ввода, задании признака исполняется процедура или функция, определенные в модуле формы. Кроме выполнения процедур и функций, назначенных для исполнения элементам интерфейса, существуют и так называемые "регламентные" процедуры.

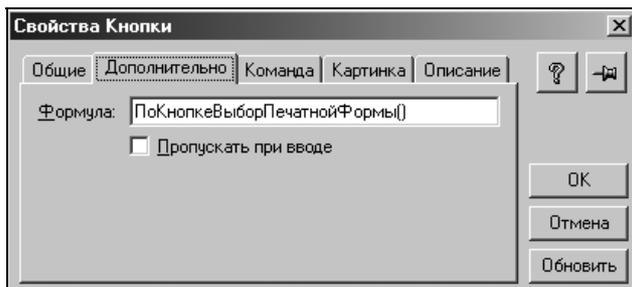
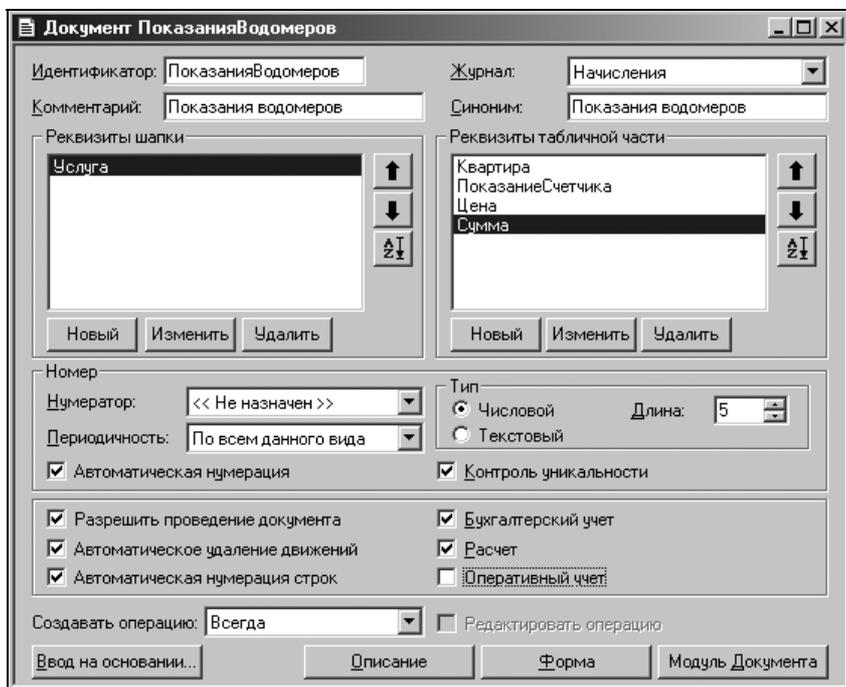
Процедуры, назначенные для исполнения объектам метаданных, называются *предопределенными процедурами*.

Процедуры и функции элементов формы

Процедуры и функции, исполняемые при действиях пользователя с элементами формы, задаются в свойствах элемента на вкладке **Дополнительно** в поле **Формула**. Имя процедуры или функции должно совпадать с именем определенной в модуле формы процедуры. В предыдущем разделе был создан модуль формы списка справочника "Квартиры". Процедура `ПоКнопкеВыборПечатнойФормы()` была назначена на элемент интерфейса типа **Кнопка** с иконкой  (рис. 3.40).

Следует обратить внимание на то, что на вкладке **Дополнительно** в качестве формулы чаще всего задаются процедуры или функции.

Характерной процедурой, обрабатывающей при вводе в несколько полей, является функция пересчета в документах. Создадим документ ввода показаний водомеров (рис. 3.41).

Рис. 3.40. Диалоговое окно свойств кнопки на вкладке **Дополнительно**Рис. 3.41. Диалоговое окно **Документ ПоказанияВодомеров**

Реквизит шапки **Услуга** имеет тип **Справочник.Услуги** (рис. 3.42).

Реквизит **Цена** справочника **Услуги** — периодический для ведения истории изменения тарифов на услуги.

Назначим в форме документа **Показания водомеров** реквизитам **Квартира**, **ПоказаниеСчетчика** и **Цена** процедуру **Пересчет ()** (рис. 3.43).

The screenshot shows a dialog window titled 'Справочник Услуги'. It contains several fields and options for configuring a service record. The 'Идентификатор' field is set to 'Услуги', and 'Подчинен' is '(не подчинено)'. The 'Кол-во. уровней' is 1, 'Длина кода' is 5, and 'Длина наименования' is 25. There are radio buttons for 'Серии кодов' (selected 'В пределах подчинения') and 'Тип кода' (selected 'Числовой'). The 'Основное представление' is set to 'В виде наименования'. A list of 'Реквизиты' contains 'Цена' and 'Расчет'. At the bottom, there are buttons for 'Описание', 'Форма элемента', 'Форма группы', and 'Формы списка'.

Идентификатор: Услуги Подчинен: (не подчинено)

Комментарий: Синоним:

Кол-во. уровней: 1 Размещать группы сверху

Длина кода: 5 Автоматическая нумерация

Длина наименования: 25 Контроль уникальности

Серии кодов:
 Во всем справочнике
 В пределах подчинения

Тип кода:
 Числовой
 Текстовый

Основное представление:
 В виде кода
 В виде наименования

Реквизиты:
Цена
Расчет

Новый Изменить Удалить

Одна форма для элемента и группы Редактировать: В списке

Описание Форма элемента Форма группы Формы списка ▶

Рис. 3.42. Диалоговое окно задания справочника Услуги

The screenshot shows a dialog window titled 'Форма-Документ. Показания Водометров'. It contains fields for 'НомерДок' (set to 'НомерДок'), 'ДатаДок' (set to 'ДатаДок'), and 'Расчет' (set to 'Услуга'). Below these is a table with columns 'N', 'Квартира', 'Показание (куб. м3)', 'Цена', and 'Сумма'. The table is currently empty. At the bottom, there are 'OK' and 'Закрыть' buttons, and a status bar with 'Диалог', 'Модуль', and 'Таблица' tabs.

НомерДок: НомерДок ДатаДок: ДатаДок

Расчет: Услуга

N	Квартира	Показание (куб. м3)	Цена	Сумма

OK Закрыть

Диалог Модуль Таблица

Рис. 3.43. Диалоговое окно формы документа Показания Водометров

При выборе квартиры из справочника, вводе показания счетчика, изменении цены будет выполняться процедура, которая производит пересчет всех реквизитов табличной части.

Процедура Пересчет ()

Цена = Услуга.Цена.Получить (ДатаДок) ;

// Получим предыдущее показание счетчика

ПоказаниеПр= Квартира.ПоказаниеХолВоды.Получить (НачМесяца (ДатаДок)) ;

Если Квартира.Льгота.Получить (ДатаДок) =1 **тогда**

Цена = Цена*Льгота.Получить (ДатаДок) ;

КонецЕсли ;

Сумма = Цена*(ПоказаниеСчетчика - ПоказаниеПр) ;

КонецПроцедуры

Предопределенные процедуры

В модуле формы можно определить процедуры, которые выполняются при задании пользователем определенных действий, таких как открытие документа, элемента справочника, создание нового документа или элемента справочника, запись документа или справочника; при вводе документа на основании другого и т. п. Предопределенные процедуры имеют специальные имена и набор параметров, причем они отличаются для модуля формы и формы списка.

Статус возврата

Во всех предопределенных процедурах имеется возможность отменить выполнение действия, для обработки которого процедура предназначена. Для этого существует специальная функция СтатусВозврата (Статус).

Если необходимо отменить действие, которое обрабатывает предопределенная процедура, то значение параметра Статус равно 0, в противном случае значение параметра Статус — 1.

Начальное значение статуса возврата предопределенной процедуры равно 1 (Выполнить Действие), устанавливается системой при вызове предопределенной процедуры.

Предопределенные процедуры справочников

Форма элемента справочника может быть только одна. Форм списка справочника может быть сколько угодно. Предопределенные процедуры задаются для формы элемента справочника в модуле формы элемента справочника, для форм списка — в модулях форм списка.

Предопределенные процедуры формы элемента справочника

Рассмотрим предопределенные процедуры для формы элемента справочника.

❖ Процедура `ВводНового` (`ПризнакКопирования`, `ОбъектКопирования`) выполняется при интерактивном вводе нового элемента справочника. Система `1С:Предприятие` передает процедуре два параметра:

- `ПризнакКопирования` — имеет значение 1, если новый элемент справочника создается путем копирования;
- `ОбъектКопирования` — в том случае, если параметр `ПризнакКопирования` имеет значение 1.

Как правило, данная процедура применяется для задания реквизитам значеный "по умолчанию". Например, в нашем справочнике "Квартиры" можно задать в модуле формы элемента количество жильцов по умолчанию 1 и льгота — 1 следующим образом:

Процедура `ВводНового` (`Копирование`)

Если `Копирование = 0` **тогда**

`КоличествоЖильцов = 1;`

`Льгота = 1;`

КонецЕсли;

КонецПроцедуры // `ВводНового` ()

❖ Процедура `ПриЗаписи` (`СписокПериодическихРеквизитов`) выполняется при интерактивной записи элемента справочника.

Параметр `СписокПериодРекв` представляет собой строку со списком изменяемых периодических реквизитов справочника. В данный параметр система `1С:Предприятие` передает перечень периодических реквизитов, которые были интерактивно выбраны пользователем для обновления в окне диалога выбора. В теле процедуры значение данного параметра может быть изменено, что позволяет в данной процедуре непосредственно управлять списком записываемых значений периодических реквизитов. В *разд. "Оператор Попытка... Исключение"* данной главы был приведен фрагмент из обработки `ЗаписьПериодическихРеквизитов`, в которой пользователю предлагается диалоговое окно для задания даты периодических реквизитов с информацией о каждом из них. Вызов обработки задается из процедуры `ПриЗаписи`:

Процедура `ПриЗаписи` ()

`СтатусВозврата` (`глБухЗаписьПериодическихРеквизитов` (`Контекст`)) ;

КонецПроцедуры // `ПриЗаписи` ()

❖ Процедура `ПриОткрытии()` выполняется при интерактивном открытии формы элемента справочника.

Внимание!

В Синтакс-Помощнике процедуры `ПриОткрытии()` и `ПриЗакрытии()` отнесены в раздел предопределенных процедур форм списка справочника, на самом деле эти процедуры выполняются при интерактивном открытии и закрытии как формы элемента справочника, так и формы списка.

Определим процедуру `ПриОткрытии()` в модуле формы элемента справочника "Квартиры" для управления количеством закладок в форме. Введем новый реквизит в справочник **ВидУчетаВоды** (рис. 3.44) и реквизиты показаний водомеров **ПоказаниеГорВоды** и **ПоказаниеХолВоды**.

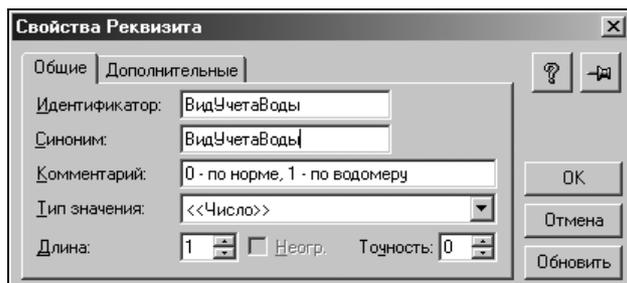


Рис. 3.44. Диалоговое окно свойств реквизита **ВидУчетаВоды**

Создадим два новых слоя для распределения реквизитов с учетом вида учета воды.

Слой **Основной** содержит общие реквизиты (рис. 3.45).

Слой **Кнопки** содержит кнопки и признак учета воды по водомеру (рис. 3.46).

Слой **ПоказанияСчетчиков** содержит реквизиты ввода показаний и кнопку **История** (рис. 3.47).

В предопределенной процедуре `ПриОткрытии()` ведется анализ реквизита **ВидУчета воды** для активизации новой закладки:

Процедура `ПриОткрытии()`

Если `ВидУчетаВоды = 1` **тогда**

 форма.ИспользоватьЗакладки(1);

 форма.Закладки.ДобавитьЗначение("Общий", "Общие");

 форма.Закладки.ДобавитьЗначение("ПоказанияСчетчиков", "Показания счетчиков");

КонецЕсли;

Форма.ИспользоватьСлой ("Основной, Кнопки", 2);

КонецПроцедуры // ПриОткрытии

Форма элемента-Справочник. Квартиры (Квартиры)

Код:	<input type="text" value="Код"/>	Наименование:	<input type="text" value="Наименование"/>
Количество комнат:	<input type="text" value="КоличествоКомнат"/>	Дом:	<input type="text" value="Дом"/> ...
Метраж жилой:	<input type="text" value="МетражЖилой"/>	Метраж общий:	<input type="text" value="МетражОбщий"/>
Количество жильцов:	<input type="text" value="КоличествоЖиль"/>	Льгота:	<input type="text" value="Льгота"/>
Договор приобретения:	<input type="text" value="ДоговорПриобретения"/>	Телефон:	<input type="text" value="Телефон"/>

Диалог Модуль Таблица

Рис. 3.45. Диалоговое окно справочника "Квартиры", слой Основной

Форма элемента-Справочник. Квартиры (Квартиры)

Вода по водомеру

Диалог Модуль Таблица

Рис. 3.46. Диалоговое окно справочника "Квартиры", слой Кнопки

Целесообразно рассмотреть predetermined procedure of the form context `ПриВыбореЗакладки`, которую мы тоже определим в модуле формы элемента справочника:

Процедура ПриВыбореЗакладки (НомерЗакладки)

Если НомерЗакладки = 2 **тогда**

Форма.ИспользоватьСлой ("ПоказанияСчетчиков, Кнопки", 2);

Иначе

Форма.ИспользоватьСлой ("Основной, Кнопки", 2);

КонецЕсли;

КонецПроцедуры // ПриВыбореЗакладки

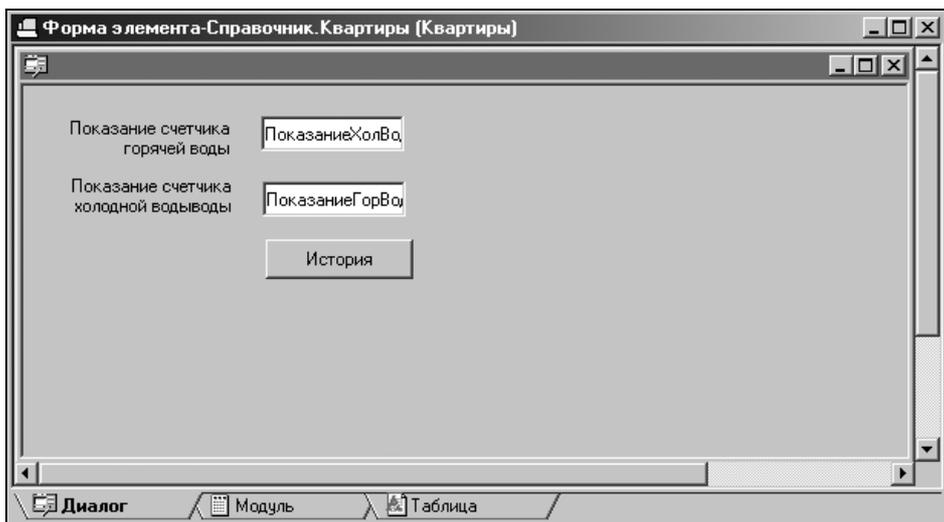


Рис. 3.47. Диалоговое окно справочника "Квартиры", слой ПоказанияСчетчиков

Наконец, процедура ПриИзмененииВида () активизирует закладку **Показания-Водомера** и назначается элементу интерфейса **Вода по водомеру**.

Процедура ПриИзмененииВида ()

Если ВидУчетаВоды = 1 **тогда**

Форма.ИспользоватьЗакладки (1);

Форма.Закладки.ДобавитьЗначение ("Общий", "Общие");

Форма.Закладки.ДобавитьЗначение ("ПоказанияСчетчиков", "Показания счетчиков");

Иначе

Форма.Закладки.УдалитьЗначение (1);

Форма.Закладки.УдалитьЗначение (1);

Форма.ИспользоватьЗакладки (0);

КонецЕсли;

Форма.ИспользоватьСлой ("Основной, Кнопки", 2);

КонецПроцедуры // ПриИзмененииВида

❖ Процедура `ПриЗакрытии()` выполняется при интерактивном закрытии формы элемента справочника. Предопределенной процедурой `ПриЗаписи()` отслеживается момент записи элемента справочника, но не момент окончания редактирования элемента справочника в форме. Данная процедура полезна для заполнения некоторых подчиненных справочников при вводе нового элемента.

Предопределенные процедуры формы списка справочника

Рассмотрим предопределенные процедуры форм списка справочников, которые задаются в модулях форм списка справочника соответственно:

- `ПриОткрытии()`;
- `ПриЗакрытии()`;
- `ПриЗаписи()`;
- `ПриВводеСтроки()`;
- `ПриРедактированииНовойСтроки()`;
- `ПриНачалеРедактированияСтроки()`;
- `ПриВыбореСтроки()`;
- `ПриПереносеЭлементаВДругуюГруппу()`;
- `ПриВыбореРодителя()`;
- `ПриВыбореВладельца()`;
- `ПриСменеИерархии()`;
- `ПриУстановкеОтбора()`.

❖ Процедура `ПриОткрытии()` — выполняется при интерактивном открытии формы списка справочника. В процедуре `ПриОткрытии()` формы списка можно проанализировать настройки пользователя или атрибут формы **Параметр** для установки отбора или изменения интерфейса формы.

❖ Процедура `ПриЗакрытии()` выполняется при интерактивном закрытии формы списка справочника. Данная процедура используется для сохранения специфических настроек формы, например, показ остатков на складе:

Процедура `ПриЗакрытии()`

СохранитьЗначение

("ФормаПодбораТоваровОтгруженных_ФлагОстаткиВКолонке",
ФлагОстаткиВКолонке);

КонецПроцедуры // `ПриЗакрытии()`

❖ Процедура `ПриЗаписи(СписокПериодическихРеквизитов)` относится к записи элемента справочника, когда редактирование элемента справочника производится в форме списка справочника, а не в отдельной форме.

❖ Процедура `ПриВводеСтроки()` выполняется при интерактивном вводе новой строки списка справочника.

❖ Процедура `ПриРедактированииНовойСтроки()` выполняется при интерактивном редактировании новой строки списка справочника. Процедура `ПриВводеСтроки()` относится к вводу нового элемента справочника в случае, когда редактирование элемента справочника производится в форме списка справочника, а не в отдельной форме. Аналогична процедуре `ВводНового()` для формы элемента справочника.

❖ Процедура `ПриНачалеРедактированияСтроки()` выполняется при начале интерактивного редактирования существующей строки списка. Процедура относится к открытию существующего элемента справочника, когда редактирование элемента справочника производится в форме списка справочника, а не в отдельной форме. Аналогична процедуре `ПриОткрытии()` для формы элемента справочника.

❖ Процедура `ПриВыбореСтроки()` выполняется при выборе строки списка (двойной щелчок мыши или клавиша <Enter>).

Рассмотрим созданный нами документ "ПоказанияВодомеров". Выбор квартиры в табличной части документа производится из справочника "Квартиры". Создадим новую форму списка для выбора квартиры и в модуле этой формы списка — предопределенную процедуру `ПриВыбореСтроки()` с использованием функции `СтатусВозрата()`:

Процедура `ПриВыбореСтроки()`

```
Если Вопрос("Вы уверены, что выбрали правильно квартиру " +  
Наименование, 1, 60) = 2 тогда  
СтатусВозрата(0);
```

Возврат

КонецЕсли;

КонецПроцедуры // `ПриВыбореСтроки()`

После выбора пользователем квартиры из списка на экране появится запрос на подтверждение выбора (рис. 3.48).

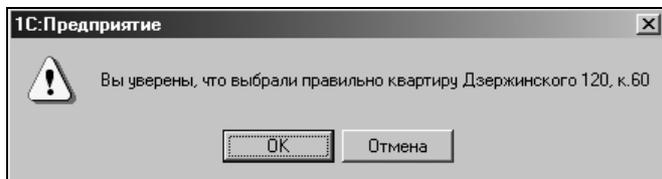


Рис. 3.48. Диалоговое окно вопроса на подтверждение выбора строки справочника "Квартиры"

Если пользователь подтверждает свой выбор, в документ возвращается выбранная строка справочника. Если пользователь нажмет кнопку **Отмена**, процедура `ПриВыбореСтроки()` вернет его в справочник для выбора другой строки.

❖ Процедура `ПриПереносеЭлементаВДругуюГруппу(Элемент,Группа)` выполняется при интерактивном переносе элемента справочника в другую группу.

Параметры процедуры:

- `Элемент` — элемент справочника;
- `Группа` — группа, в которую переносится элемент справочника.

Зададим в справочнике "Квартиры" 3 уровня иерархии. На первом уровне введем группы улиц, на втором — дома, на третьем — квартиры (рис. 3.49).

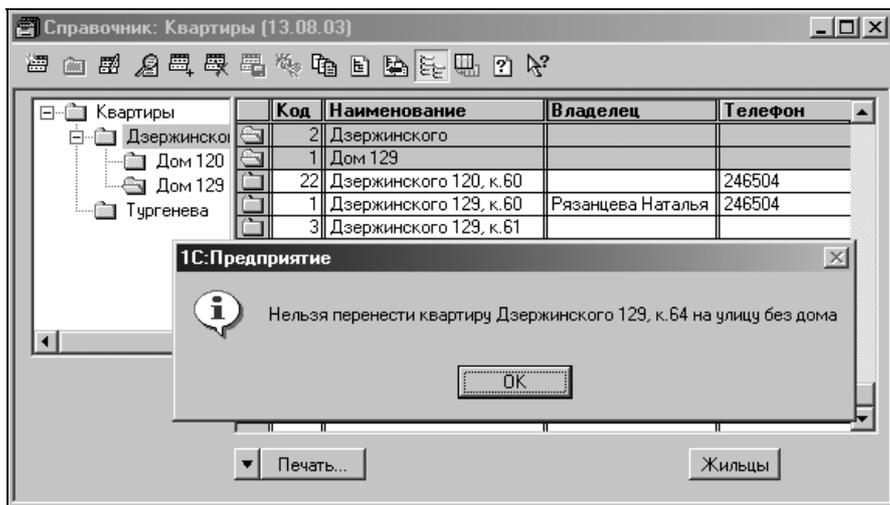


Рис. 3.49. Диалоговое окно справочника "Квартиры"

Создадим предопределенную процедуру `ПриПереносеЭлементаВДругуюГруппу()`:

Процедура `ПриПереносеЭлементаВДругуюГруппу(Элемент,Группа)`

Если `ПустаяСтрока(Группа.Родитель)=1` **тогда**

Предупреждение("Нельзя перенести квартиру " + `Элемент` + " на улицу без дома",20);

`СтатусВозврата(0)`;

Возврат

КонецЕсли;

КонецПроцедуры // `ПриПереносеЭлементаВДругуюГруппу`

При ошибочном выборе пользователя вместо дома на втором уровне сразу улицы на первом уровне будет выдано предупреждение, и действие в соответствии со статусом возврата будет отменено.

❖ **Процедура ПриВыбореРодителя (Элемент)** выполняется при интерактивной смене родительской группы справочника (выбор следующего или предыдущего уровня).

Параметр Элемент — значение элемента справочника, который интерактивно устанавливается в качестве родителя.

❖ **Процедура ПриВыбореВладельца (Элемент)** выполняется при интерактивном выборе владельца подчиненного справочника (при интерактивной смене владельца, т. е. смене позиции в справочнике-владельце, которая приводит к смене отображаемых в подчиненном справочнике элементов).

Параметр Элемент — значение элемента справочника, который интерактивно устанавливается в качестве владельца подчиненного справочника.

❖ **Процедура ПриСменеИерархии (Способ)** выполняется при интерактивной смене режима отображения иерархии справочника (пункт меню **Иерархический список**).

Параметр Способ — значение устанавливаемого (тот, который пользователь хочет установить) способа просмотра справочника:

1 — иерархический список;

0 — все элементы сразу.

❖ **Процедура ПриУстановкеОтбора (ТипОтбора, ЗначениеОтбора)** выполняется при интерактивной установке отбора любым способом (отбор, быстрый отбор, отбор по значению, история отбора) и при отключении отбора.

Параметры:

ТипОтбора — строковое значение — тип устанавливаемого отбора (имя реквизита справочника, по которому устанавливается отбор).

ЗначениеОтбора — устанавливаемое значение отбора.

Предопределенные процедуры документов

С точки зрения информации содержимое документа можно разделить на две части:

1. Информация первичного документа, т. е. совокупность реквизитов, которые задает пользователь.
2. Бухгалтерские операции, движения регистров, расчеты в журнале расчетов, которые создаются на основании введенных реквизитов в результате проведения документа.

Предопределенные процедуры, которые выполняются при действиях пользователя с первой частью информации документа, задаются в модуле формы документа. Предопределенные процедуры, которые выполняются при проведении документа, задаются в модуле документа.

Предопределенные процедуры формы документов

Для обработки интерактивных действий пользователя с документами существуют следующие предопределенные процедуры:

- ВводНового ();
- ПриОткрытии ();
- ПриЗакрытии ();
- ПриЗаписи ();
- ПриВводеСтроки ();
- ВводНаОснoвании ();
- ПриРедактированииНовойСтроки ();
- ПриНачалеРедактированияСтроки ();
- ПриОкончанииРедактированияСтроки ();
- ПриИзмененииПорядкаСтрок ();
- ПриПереносеЭлементаВДругуюГруппу ();
- ПриУдаленииСтроки () .

Как видно из приведенного перечня предопределенных процедур, пять первых аналогичны соответствующим предопределенным процедурам справочников.

Рассмотрим предопределенные процедуры, характерные только для документов.

❖Процедура `ВводНаОснoвании (ДокументОснoвание)` выполняется при интерактивном вводе нового документа на основании другого документа.

Параметр `ДокументОснoвание` — значение типа "Документ", на основании которого вводится новый документ.

Для иллюстрации создадим необходимые для работы бухгалтерии документы:

- ПоступлениеОС;
- УслугиСтороннихОрганизаций;
- справочники:
 - Контрагенты;
 - Договоры;

- ОсновныеСредства;
- ОбъектыСтроительства;
- Валюты;
- ВариантыРасчетаНалогов;
- СтавкиНДС;
- Банки;
- РасчетныеСчета;
- Подразделения;
- Сотрудники;
- КлассификаторЕН;
- ПланСчетов;

□ перечисления:

- ПричиныВыбытияОС;
- СостоянияОС;
- СпособыНачисленияАмортизацииОС.

Оформим поступление основного средства "Комплект офисной мебели" с дополнительной услугой сторонней организации "Доставка и сборка мебели", которую введем на основании документа поступления. В документе **Услуги-СтороннихОрганизаций** в диалоговом окне **Ввод на основании** установим признак документа — основания документу **Поступление ОС** (рис. 3.50).

Процедурой `ВводНаОсновании()` заполняются поля нового документа, необходимые для формирования проводок:

Процедура `ВводНаОсновании(ДокОсн)`

`ДокументПоступления = ДокОсн;`

`ЗачитыватьАванс = ДокОсн.ЗачитыватьАванс;`

`ВариантРасчетаНалогов = ДокОсн.ВариантРасчетаНалогов;`

`ТипИсполнителя = 1;`

`НДСвключатьВСтоимость = ДокОсн.НДСвключатьВСтоимость;`

`Комментарий = "Введен на основании: "+ГлПредставлениеДокумента(ДокОсн);`

КонецПроцедуры // `ВводНаОсновании()`

В данном случае в модуле проведения документа реквизиты **ДокументПоступления** и **ТипИсполнителя**, заполненные процедурой `ВводНаОсновании()`, содержат информацию для формирования проводок по счету 08.4 "Приобретение отдельных объектов ОС" и счету 60.1 "Расчеты с поставщиками в рублях".

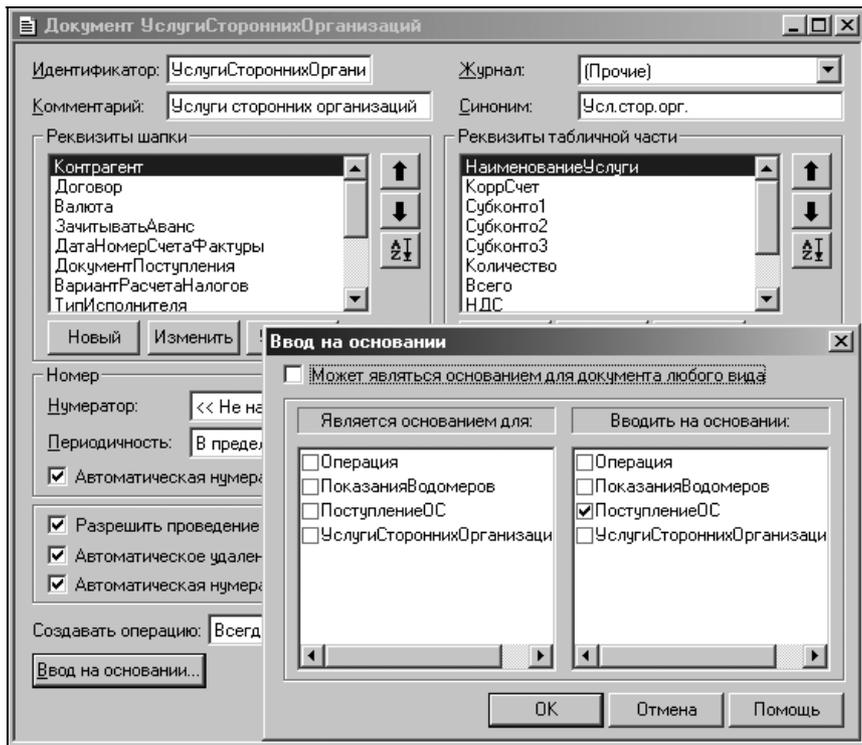


Рис. 3.50. Диалоговое окно УслугиСтороннихОрганизаций

❖ Процедура `ПриВводеСтроки()` выполняется при интерактивном вводе строки табличной части документа. Процедура может быть полезной, если по каким-либо признакам необходимо заблокировать ввод новой строки.

❖ Процедура `ПриНачалеРедактированияСтроки()` выполняется при интерактивном редактировании строки табличной части документа. Процедуру удобно использовать при однотипных операциях в разных реквизитах строки.

❖ Процедура `ПриОкончанииРедактированияСтроки(НовСтр)` выполняется при окончании редактирования строки табличной части документа.

Параметр `НовСтр` — флаг новой строки:

1 — если произошло окончание ввода новой строки;

0 — если произошло окончание редактирования существующей строки.

❖ Процедура `ПриИзмененииПорядкаСтрок(Действие)` выполняется при изменении порядка строк табличной части документа.

Параметр Действие:

- 1 — перемещение строки вверх;
- 1 — перемещение строки вниз;
- 0 — перенумерация строк.

❖ Процедура ПриУдаленииСтроки () выполняется при интерактивном удалении строки табличной части документа.

Интерактивные действия пользователя, которые необходимо ограничить по каким-либо признакам, удобно обрабатывать в приведенных процедурах. Рассмотрим документ **ПоказанияВодомеров**. Документы начисления по квартирам вводятся ежемесячно, отчетный период — месяц. После закрытия периода документ редактировать нельзя. Обработать данный запрет можно следующей предопределенной процедурой:

Процедура ПриУдаленииСтроки ()

Если ПроведенВПрошлом=1 **Тогда**

Предупреждение ("Документ проведен в прошлом периоде. Его нельзя редактировать.", 10);

СтатусВозврата (0);

КонецЕсли;

КонецПроцедуры // ПриУдаленииСтроки

Предопределенные процедуры модуля документа

Для обработки проведения документа существуют следующие предопределенные процедуры:

- ОбработкаПроведения ();
- ОбработкаУдаленияПроведения ();
- АрхивироватьДокумент () .

❖ Процедура ОбработкаПроведения (Знач) процедура обработки проведения документа в любом режиме.

Параметр Знач — идентификатор переменной, которая получает значение из процедуры Провести () .

Создадим процедуру ОбработкаПроведения () для нашего документа **ПоказанияВодомеров**. Все расчеты начислений оплаты за коммунальные услуги по квартирам будем вести в журнале расчетов **Начисления** (рис. 3.51).

Определим вид расчета оплаты коммунальной услуги **ХолоднаяВодаПоВодомеру** (рис. 3.52).

Предоставим пользователю возможность задавать шаблоны проводок по ним, т. е. создадим справочник **ВидыРасчетов** (рис. 3.53).

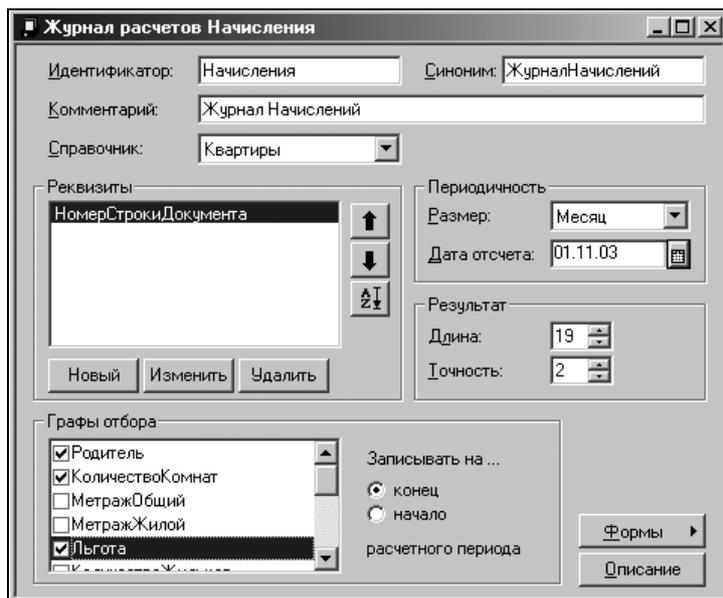


Рис. 3.51. Диалоговое окно журнала Начисления

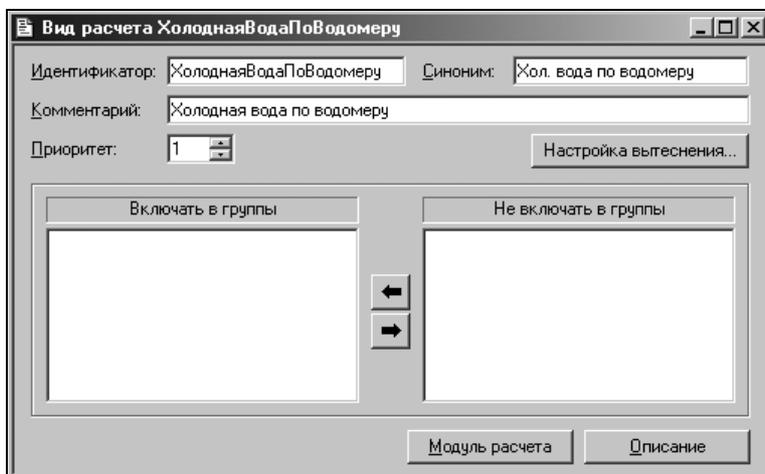


Рис. 3.52. Диалоговое окно вида расчета Холодная вода по водомеру

Предоставим пользователю возможность также задавать произвольные виды расчетов, т. е. создадим обработку **Ввод расчета** и введем для ее запуска кнопку **Помощник ввода расчета** (рис. 3.54).

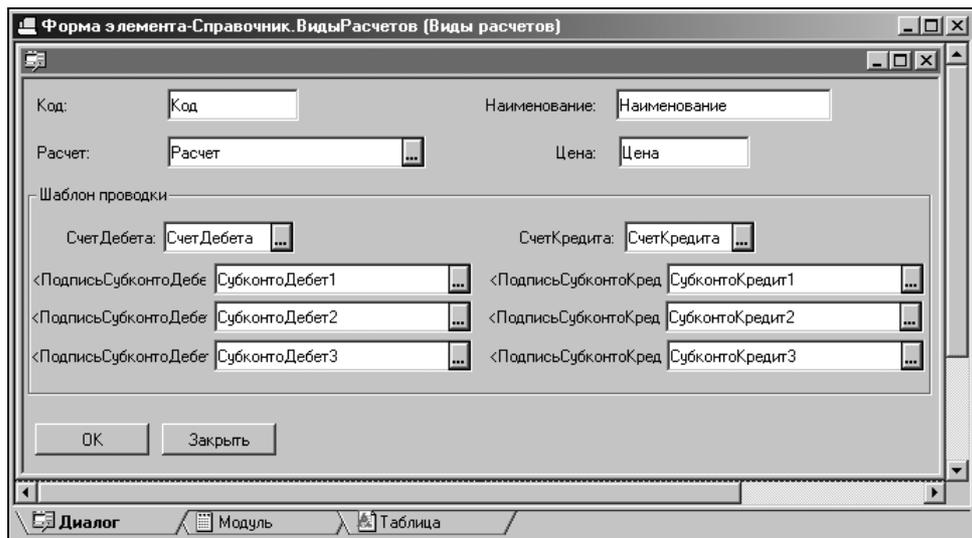


Рис. 3.53. Диалоговое окно формы элемента справочника **ВидыРасчетов**

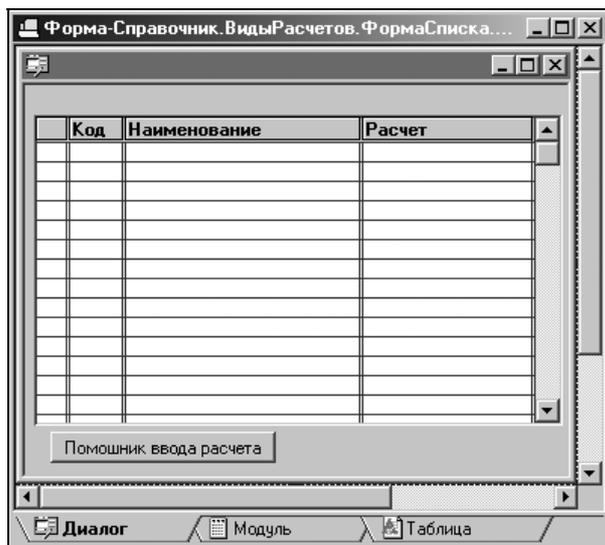


Рис. 3.54. Диалоговое окно формы списка справочника **ВидыРасчетов**

С учетом всех необходимых расчетов определим процедуру обработки проведения:

Процедура ОбработкаПроведения ()

ЖурналРасчетовНачислений = СоздатьОбъект ("ЖурналРасчетов.Начисления");

```

ЖурналРасчетовНач = СоздатьОбъект ("ЖурналРасчетов.Начисления");
ДатаН = НачМесяца (ДатаДок);
ДатаК = КонМесяца (ДатаДок);
Пока ПолучитьСтроку ()=1 Цикл
// Создадим записи в журнале расчетов
Если ЖурналРасчетовНачислений.ВыбратьПериодПоОбъекту (Квартира,
    ЖурналРасчетовНачислений.КонецТекущегоПериода ()) = 1 Тогда
    ЖурналРасчетовНачислений.УстановитьРеквизит (
        "НомерСтрокиДокумента", НомерСтроки);
    ЖурналРасчетовНачислений.ВвестиРасчет (Квартира, Услуга.Расчет, ДатаН,
        ДатаК);

Иначе
    ЖурналРасчетовНачислений.Новая ();
    ЖурналРасчетовНачислений.УстановитьРеквизит ("Объект", Квартира);
    ЖурналРасчетовНачислений.УстановитьРеквизит ("ВидРасч",
        Услуга.Расчет);
    ЖурналРасчетовНачислений.УстановитьРеквизит ("Документ",
        ТекущийДокумент ());
    ЖурналРасчетовНачислений.УстановитьРеквизит (
        "РодительскийДокумент", ТекущийДокумент ());
    ЖурналРасчетовНачислений.УстановитьРеквизит ("ДатаНачала", ДатаН);
    ЖурналРасчетовНачислений.УстановитьРеквизит ("ДатаОкончания",
        ДатаК);
    ЖурналРасчетовНачислений.УстановитьРеквизит (
        "НомерСтрокиДокумента", НомерСтроки);
    ЖурналРасчетовНачислений.Записать ();

КонецЕсли;

КонецЦикла;

Жильцы = СоздатьОбъект ("Справочник.Жильцы");
ТаблицаНачислений = СоздатьОбъект ("ТаблицаЗначений");
ТаблицаНачислений.НоваяКолонка ("Контрагент", "Справочник.Контрагенты");
ТаблицаНачислений.НоваяКолонка ("Сумма");
ЖурналРасчетовНач.ВыбратьЗаписиПоДокументу (ТекущийДокумент ());
Пока ЖурналРасчетовНач.ПолучитьЗапись ()=1 цикл
    ЖурналРасчетовНач.Рассчитать ();
Если ЖурналРасчетовНач.Результат>0 тогда
    Жильцы.ИспользоватьВладельца (ЖурналРасчетовНач.Объект);
    Жильцы.ВыбратьЭлементы ();
Пока Жильцы.ПолучитьЭлемент ()=1 цикл
Если Жильцы.СтатусПрописки=Перечисление.Статусы.Владелец тогда
    ТаблицаНачислений.НоваяСтрока ();
    ТаблицаНачислений.Контрагент = Жильцы.Контрагент;
    ТаблицаНачислений.Сумма = ЖурналРасчетовНач.Результат;
    Прервать;
КонецЕсли;

```

КонецЦикла ;

КонецЕсли ;

КонецЦикла ;

ТаблицаНачислений.Свернуть ("Контрагент", "Сумма") ;

ТаблицаНачислений.ВыбратьСтроки () ;

// Создадим проводки

Пока ТаблицаНачислений.ПолучитьСтроку () = 1 **Цикл**

Операция.НоваяПроводка () ;

Операция.Кредит.Счет = СчетПоКоду ("90.1.1") ;

Операция.Кредит.ВидыНоменклатуры =

Константа.ОсновнойВидНоменклатуры;

Операция.Дебет.Счет = СчетПоКоду ("62.1") ;

Операция.Дебет.Контрагенты = ТаблицаНачислений.Контрагент ;

Операция.Сумма = ТаблицаНачислений.Сумма ;

Операция.СодержаниеПроводки = "Начисл. за ком. услуги" ;

КонецЦикла ;

Операция.Записать () ;

КонецПроцедуры

❖Процедура `ОбработкаУдаленияПроведения()` выполняется при обработке удаления проведения документа или удаления проведенных документов в любом режиме. Если некоторые реквизиты получены расчетным путем с использованием периодических реквизитов документов или справочников, результат расчета может остаться некорректным даже после простого удаления проведения. Для выполнения поправок и существует предопределенная процедура `ОбработкаУдаленияПроведения()`. Уберем расчет суммы оплаты воды по водомеру в нашем документе **ПоказанияВодомеров**:

Процедура `ОбработкаУдаленияПроведения()`

Если `ВыбратьСтроки()` = 1 **Тогда**

Пока `ПолучитьСтроку()` = 1 **Цикл**

`Сумма` = 0 ;

КонецЦикла ;

КонецЕсли ;

КонецПроцедуры // `ОбработкаУдаленияПроведения()`

❖Процедура `АрхивироватьДокумент()` выполняется для обработки архивирования документа.

Данная процедура может располагаться только в модуле документа, вызываемом в Конфигураторе: **Документ | Редактировать | Модуль документа**. Вызов процедуры происходит в процессе смены расчетного периода журнала расчетов.

Предопределенные процедуры журнала документов

Предопределенные процедуры журналов документов задаются в их модулях формы. Для обработки интерактивных действий пользователя с журналом документов существуют следующие предопределенные процедуры:

- ПриУстановкеОтбора ();
- ПриВыбореСтроки ();
- ПриУстановкеИнтервала ().

❖ Процедура ПриУстановкеОтбора (ИмяРеквизОтбора, Значение) выполняется при интерактивной установке отбора документов в журнале.

Параметры процедуры:

- ИмяРеквизОтбора — строка с названием общего реквизита документа, по которому производится отбор;
- Значение — значение реквизита отбора.

❖ Процедура ПриВыбореСтроки () выполняется при выборе строки списка (двойной щелчок мыши или клавиша <Enter>). Целесообразно использовать данную процедуру при дальнейшем использовании выбранного документа журнала:

```
Процедура ПриВыбореСтроки ()
    Форма.Параметр = ТекущийДокумент;
КонецПроцедуры // ПриВыбореСтроки ()
```

❖ Процедура ПриУстановкеИнтервала (ДатаНач, ДатаКон) выполняется при установке интервала журнала.

Параметры процедуры:

- ДатаНач — дата начала интервала журнала;
- ДатаКон — дата конца интервала журнала.

В нашей конфигурации в журнале **Операция** в модуле формы списка создадим следующую процедуру:

```
Процедура ПриУстановкеИнтервала (ДатаНач, ДатаКон)
    Если (ДатаНач < ДатаИзВарианта (НачалоСтандартногоИнтервала (), 0)) или
        (ДатаКон > ДатаИзВарианта (КонецСтандартногоИнтервала (), 1)) Тогда
        Предупреждение ("Указанный интервал выходит за границы,
        | установленные в параметрах системы для журналов документов,
        | операций и проводок. Он не сохранится после закрытия журнала.
        | Установка интервала для всех журналов выполняется из меню Сервис в
        | настройке параметров системы (Параметры) на закладке Журналы.");
    КонецЕсли;
КонецПроцедуры
```

В процедуре используется функция определения дат начала и конца интервала из настройки пользователя ДатаИзВарианта () :

```
Функция ДатаИзВарианта (Вариант, КонецИнтервала)
    ДатаГраницы = "";
Если ТипЗначенияСтр (Вариант) = "Дата" Тогда
    ДатаГраницы = Вариант;
Иначе
    Если Вариант = "Месяц" Тогда
        Если КонецИнтервала = 0 Тогда // начало интервала
            ДатаГраницы = НачМесяца (РабочаяДата ( ) );
        Иначе
            ДатаГраницы = КонМесяца (РабочаяДата ( ) );
        КонецЕсли;
    ИначеЕсли Вариант = "Квартал" Тогда
        Если КонецИнтервала = 0 Тогда // начало интервала
            ДатаГраницы = НачКвартала (РабочаяДата ( ) );
        Иначе
            ДатаГраницы = КонКвартала (РабочаяДата ( ) );
        КонецЕсли;
    ИначеЕсли Вариант = "Год" Тогда
        Если КонецИнтервала = 0 Тогда // начало интервала
            ДатаГраницы = НачГода (РабочаяДата ( ) );
        Иначе
            ДатаГраницы = КонГода (РабочаяДата ( ) );
        КонецЕсли;
    Иначе
        ДатаГраницы = РабочаяДата ( ) ;
    КонецЕсли;
КонецЕсли;
Возврат ДатаГраницы;
Конецфункции //ДатаИзВарианта
```

Внимание!

Приведенная процедура ПриУстановкеИнтервала () и функция ДатаИзВарианта () разработаны в типовой конфигурации "1С:Бухгалтерский учет".

Предопределенные процедуры журнала расчетов

Предопределенные процедуры журналов расчетов задаются в модуле формы журнала расчетов. Для обработки интерактивных действий пользователя с журналом документов существуют следующие предопределенные процедуры:

- ПриИсправленииРезультата ();
- ПриОтменеИсправления ();
- ПриРасчете ();
- ПриУстановкеОтбора ();
- ПриУстановкеГраницыПросмотра ();
- ПриУстановкеПредставления ();
- ПриВыбореВладельца () .

❖Процедура ПриИсправленииРезультата (Запись) выполняется при редактировании результата расчета записи журнала расчетов.

Параметр процедуры Запись — ссылка на запись журнала расчетов, результат расчета которой исправляется.

Если результаты расчета по какой-либо причине необходимо отредактировать (например, убрать копейки), двойным щелчком мыши вызывается диалоговое окно **Исправление результата**. После ввода нужной суммы выполняется процедура ПриИсправленииРезультата (). Создадим процедуру в нашем журнале расчетов **Начисления** с запросом на подтверждение:

Процедура ПриИсправленииРезультата (Запись)

Если Запись.Исправлена = 0 **тогда**

Если Вопрос ("Сохранить изменения результата расчета?", 1, 60) = 2 **тогда**
СтатусВозврата (0);

Возврат;

КонецЕсли;

КонецЕсли;

КонецПроцедуры // ПриИсправленииРезультата ()

❖Процедура ПриОтменеИсправления (Запись) выполняется при отмене исправления результата расчета в журнале расчета.

Параметр процедуры Запись — ссылка на запись журнала расчетов, исправление результата которой отменяется.

Если необходимо отменить ручное исправление и рассчитать запись заново, нужно двойным щелчком мыши вызвать диалоговое окно **Исправление результата**. После ввода нужной суммы выполняется процедура ПриИсправленииРезультата (). Создадим процедуру в нашем журнале расчетов **Начисления** с запросом на подтверждение:

Процедура ПриОтменеИсправления (Запись)

Если Вопрос ("Исправление будет потеряно! Отменить исправление?", 1, 60) = 2 **тогда**

СтатусВозврата (0);

Возврат;

КонецЕсли;

КонецПроцедуры // ПриОтменеИсправления()

❖ Процедура ПриРасчете (ОбъектРасчета) выполняется при расчете одной записи, записей объекта или записей документа в журнале расчетов.

Параметр процедуры ОбъектРасчета — запись журнала расчетов или элемент справочника, являющийся объектом расчета, или документ, записи которого рассчитываются. Что в данный момент передается системой при вызове процедуры, зависит от выполняемого действия: расчет одной записи, расчет объекта или расчет документа.

Для иллюстрации возможностей процедуры при расчете по всему объекту введем новый документ **ВводНачальногоСальдо** (рис. 3.55).

Рис. 3.55. Диалоговое окно задания документа ВводНачальногоСальдо

Введем константу **ПределСуммыДолга**, чтобы предоставить пользователю возможность задавать нижнюю границу долга за коммунальные услуги по квартире. Процедура ПриРасчете () при выборе пункта меню **Рассчитать объект** будет вести анализ задолженности квартиры за коммунальные услуги:

Процедура ПриРасчете (Объект)

Если ТипЗначенияСтр(Объект) = "Справочник" **Тогда** // расчет по объекту
 ЖурналРасчетов = СоздатьОбъект("ЖурналРасчетов.Начисления");
 ЖурналРасчетов.ВыбратьЗаписиПоОбъекту(Объект, ДатаНачала, ДатаОкончания);
 СуммаНачисления=0;

СуммаДолга = 0;

Пока ЖурналРасчетов.ПолучитьЗапись()=1 **Цикл**

Если ЖурналРасчетов.ВидРасч = ВидРасчета.НачальноеСальдо **тогда**

СуммаДолга = ЖурналРасчетов.Результат;

Иначе

СуммаНачисления = СуммаНачисления+ЖурналРасчетов.Результат;

КонецЕсли;

КонецЦикла;

Долг = -(СуммаНачисления+СуммаДолга);

Если Долг>Константа.ПределСуммыДолга.Получить(ДатаОкончания) **тогда**

Если Вопрос(Строка(Долг)+" превышает порог долга! Печатать сводную
 квитанцию?", 1, 60)=1 **тогда**

ПечатьСводнойКвитанции();

КонецЕсли;

КонецЕсли;

КонецЕсли;

КонецПроцедуры

❖Процедура ПриУстановкеОтбора(ИмяОтбора, ЗначениеОтбора) выполняется при установке отбора в форме журнала расчетов.

Параметры процедур:

- ИмяОтбора — строковое значение. Наименование графы отбора, которая выбрана пользователем при интерактивной установке отбора.
- ЗначениеОтбора — значение отбора, которое выбрано пользователем при интерактивной установке отбора.

Создадим процедуру ПриУстановкеОтбора() для организации отображения полной строки сложного отбора с вложенным условием, например, все записи должников по трехкомнатным квартирам заданного дома. Стандартная реализация такого отбора отображает в заголовке формы только последнее условие отбора, что не совсем удобно для пользователя.

Введем в начале модуля переменную Отбор:

ПеремОтбор;

Инициализируем значение данной переменной в конце модуля формы списка:

Отбор ="";

Наша процедура выполняется до того, как меняется заголовок формы, поэтому в форме списка зададим элемент диалога типа **Текст** с идентификатором **ТекстОтбора** (рис. 3.56).

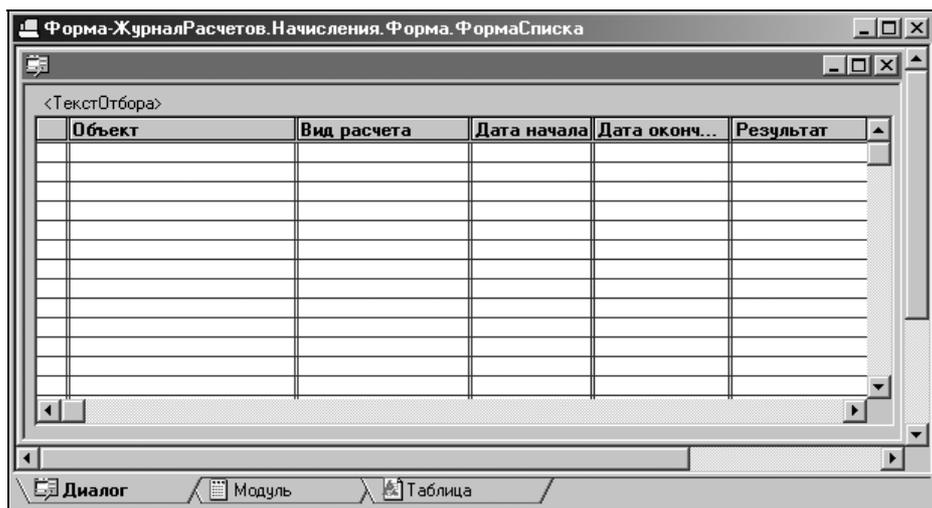


Рис. 3.56. Диалоговое окно формы журнала Начисления

Наша процедура формирует полную строку установленного отбора:

Процедура ПриУстановкеОтбора (ИмяОтбора, ЗначениеОтбора)

Если ПустаяСтрока (ИмяОтбора) = 1 **тогда**

Отбор = "";

ИначеЕсли ПустаяСтрока (Отбор) = 1 **тогда**

Отбор = "Установлен отбор "+ИмяОтбора+" = "+ЗначениеОтбора;

Иначе

Отбор=Отбор+" и "+ИмяОтбора+" = "+ЗначениеОтбора;

КонецЕсли;

форма.ТекстОтбора.Заголовок (Отбор) ;

КонецПроцедуры

❖ **Процедура** ПриУстановкеГраницыПросмотра (Период) выполняется при установке границы просмотра журнала расчетов.

Параметр **Период** — значение типа **Период** журнала расчетов. **Период**, устанавливаемый пользователем как граница просмотра записей.

Аналогично процедуре ПриУстановкеОтбора () можно отслеживать отображение границы просмотра записей процедурой ПриУстановкеГраницыПросмотра (), хотя это не так актуально.

❖ Процедура ПриУстановкеПредставления (Режим) выполняется при установке режима представления записей журнала расчетов.

Параметр: Режим — числовое значение — режим представления записей журнала расчетов, устанавливаемый пользователем. Параметр может принимать следующие значения:

- 1 — выбран режим представления всех записей журнала расчетов за текущий период;
- 2 — выбран режим представления записей текущего объекта за текущий период;
- 3 — выбран режим представления записей текущего документа за текущий период.

Аналогично процедуре ПриУстановкеОтбора () можно отслеживать отображение текущей суммы долга при выполнении процедуры ПриУстановкеПредставления () с параметром 2:

Процедура ПриУстановкеПредставления (Режим)

Если Режим = 2 **тогда**

ЖурналРасчетов = СоздатьОбъект ("ЖурналРасчетов.Начисления");

ЖурналРасчетов.ВыбратьЗаписиПоОбъекту (Объект, ДатаНачала, ДатаОкончания);

СуммаНачисления=0;

СуммаДолга = 0;

Пока ЖурналРасчетов.ПолучитьЗапись ()=1 **Цикл**

Если ЖурналРасчетов.ВидРасч = ВидРасчета.НачальноеСальдо **тогда**

СуммаДолга = ЖурналРасчетов.Результат;

Иначе

СуммаНачисления = СуммаНачисления+ЖурналРасчетов.Результат;

КонецЕсли;

КонецЦикла;

Долг = - (СуммаНачисления+СуммаДолга);

Если Долг>Константа.ПределСуммыДолга **тогда**

Форма.ТекстОтбора.Заголовок ("Квартира - должник! Сумма долга - "+Долг);

Иначе

Форма.ТекстОтбора.Заголовок ("");

КонецЕсли;

Иначе

Форма.ТекстОтбора.Заголовок ("");

КонецЕсли;

КонецПроцедуры

❖ Процедура ПриВыбореВладельца (Элемент) выбора элемента справочника, по которому будут выведены расчеты, выполняется при интерактивной смене

владельца, т. е. смене позиции в "главном" справочнике, для которого создан журнал расчетов.

Параметр `Элемент` — значение устанавливаемого владельца, т. е. элемент справочника, по которому будут выведены расчеты.

Предопределенные процедуры контекста формы

Формы, которые открываются не интерактивно пользователем, а вызовом из программных модулей, также имеют ряд предопределенных процедур, которые описываются в модуле формы и выполняются независимо от принадлежности к объекту метаданных:

- ПриОткрытии ();
- ПриПовторномОткрытии ();
- ПриЗакрытии ();
- ПриВыбореЗакладки ();
- ОбработкаПодбора ();
- ПриНачалеВыбораЗначения ();
- ОбработкаВыбораЗначения ();
- ПриВыбореСтроки () .

❖ Процедура `ПриПовторномОткрытии ()` выполняется при повторном открытии формы в случае, если открывают уже открытую форму, т. е. форма просто активизируется.

❖ Процедура `ПриВыбореЗакладки (НомерЗакладки, ЗначениеЗакладки)` обработки подбора значения выполняется в момент интерактивного выбора пользователем закладки в форме.

Параметры:

- `НомерЗакладки` — числовое значение — номер выбранной закладки формы;
- `ЗначениеЗакладки` — значение выбранной закладки формы.

Процедуру `ПриВыбореЗакладки ()` мы уже создали для формы элемента справочника "Квартиры" (см. разд. "Предопределенные процедуры формы элемента справочника" данной главы):

Процедура `ПриВыбореЗакладки (НомерЗакладки)`

Если `НомерЗакладки = 2` **тогда**

`Форма.ИспользоватьСлой ("ПоказанияСчетчиков, Кнопки", 2);`

Иначе

`Форма.ИспользоватьСлой ("Основной, Кнопки", 2);`

КонецЕсли;

КонецПроцедуры // `ПриВыбореЗакладки`

❖ **Процедура** `ОбработкаПодбора (Элемент, КонтФормы)` обработки подбора значения выполняется при нажатия кнопки **Выбрать** в **Форме Подбора** значения.

Параметры:

- `Элемент` — элемент справочника подбора или документ, передаваемый для обработки;
- `КонтФормы` — контекст той формы, из которой шел подбор.

В процедуре `ОбработкаПодбора ()` программист может обработать список выбранных элементов на предмет повтора или задания дополнительного критерия выборки из списка.

❖ **Процедура** `ПриНачалеВыбораЗначения (ИдентЭлемДиалога, ФлагСтандОбр)` выполняется после выбора значения в форме выбора (выбор может быть инициирован в немодальном режиме интерактивно, с помощью элемента диалога с "педалькой").

Параметры:

- `ИдентЭлемДиалога` — идентификатор элемента диалога, которым инициализирован выбор значения;
- `ФлагСтандОбр` — флаг, изначально, при вызове процедуры, равен 1, если в теле процедуры значение этого параметра поменять на 0, то стандартный процесс выбора значения не будет происходить.

❖ **Процедура** `ОбработкаВыбораЗначения (ВыбЗнач, ИдентЭлемДиалога, ФлагСтандОбр)` выполняется после выбора значения в форме выбора (выбор может быть инициирован в немодальном режиме интерактивно, с помощью элемента диалога с "педалькой").

Параметры:

- `ВыбЗнач` — выбранный элемент справочника, документ или иной объект, передаваемый для обработки;
- `ИдентЭлемДиалога` — идентификатор элемента диалога, которым инициализирован выбор значения;
- `ФлагСтандОбр` — флаг, установка которого в теле процедуры в 0 (ноль) приведет к отмене стандартного присвоения значения.

Предопределенные процедуры форм отчетов и обработок имеют свою специфику, хоть и имеют одинаковые названия с аналогичными процедурами других объектов метаданных:

- `ВводНового ()`;
- `ПриОткрытии ()`;
- `ПриВыбореЯчейкиТаблицы ()`.

❖ Процедура `ВводНового()` выполняется при открытии формы отчета (обработка) и при восстановлении сохраненной настройки отчета (обработки).

❖ Процедура обработки ячейки таблицы в режиме ввода данных `ПриВыбореЯчейкиТаблицы(Адрес, Значение)` выполняется по двойному щелчку мыши или по нажатию клавиши `<Enter>` в табличном документе на выбранной ячейке.

Параметры:

□ `Адрес` — здесь в процедуру системой передается строковое значение имени области таблицы, если выбранная ячейка помечена в таблице, как отдельная область, или адрес ячейки в формате `"R1C1:R2C2"`;

□ `Значение` — здесь в процедуру системой передается значение данной ячейки, а если диаграмма — значение выбранного элемента.

❖ Процедура `ПриОткрытии(ФлагЧтенияНастройки)` выполняется при открытии формы отчета (обработки).

Параметр `ФлагЧтенияНастройки` — числовое значение — признак считывания сохраненной настройки отчета (обработки). Может принимать значения:

□ 1 — при открытии формы была восстановлена последняя сохраненная настройка отчета (обработки);

□ 0 — при открытии формы настройка не восстановлена.

Создадим механизм настройки колонок для нашего справочника "Квартиры", для реализации которого необходим не интерактивный вызов формы **НастройкаКолонок**.

Для этого добавим кнопку **Колонки** в форме списка справочника "Квартиры", зададим процедуру `УстановитьВидимостьКолонок()`, которая будет вызываться нажатием кнопки (рис. 3.57).

Процедура `УстановитьВидимостьКолонок(ПриОткрытии=0)`

`Перем СписокКолонок, ТекущаяКолонка, ИдКолонки, Колонка;`

`ПолныйСписокКолонок = СоздатьОбъект("СписокЗначений");`

`ПолныйСписокКолонок.ДобавитьЗначение("Код", "Код");`

`ПолныйСписокКолонок.ДобавитьЗначение("Наименование", "Наименование");`

`ПолныйСписокКолонок.ДобавитьЗначение("КоличествоКомнат", "КоличествоКомнат");`

`ПолныйСписокКолонок.ДобавитьЗначение("МетражОбщий", "МетражОбщий");`

`ПолныйСписокКолонок.ДобавитьЗначение("МетражЖилой", "МетражЖилой");`

`ПолныйСписокКолонок.ДобавитьЗначение("Льгота", "Льгота");`

`ПолныйСписокКолонок.ДобавитьЗначение("КоличествоЖильцов", "КоличествоЖильцов");`

`ПолныйСписокКолонок.ДобавитьЗначение("Дом", "Дом");`

```
ПолныйСписокКолонок.ДобавитьЗначение ("ДоговорПриобретения",
"ДоговорПриобретения");
```

```
ПолныйСписокКолонок.ДобавитьЗначение ("Телефон", "Телефон");
```

```
// При первом запуске, пока пользователь не настроил видимость колонок,
// будут видимы только колонки "Код", "Наименование"
```

```
ПолныйСписокКолонок.Пометка (1, 1);
```

```
ПолныйСписокКолонок.Пометка (2, 1);
```

Если ПриОткрытии=0 **Тогда**

```
СписокПараметров= СоздатьОбъект ("СписокЗначений");
```

```
СписокПараметров.ДобавитьЗначение (Контекст);
```

```
СписокПараметров.ДобавитьЗначение ("Справочник_Квартиры_ФормаСписка_
ДляВыбора");
```

```
СписокПараметров.ДобавитьЗначение (ПолныйСписокКолонок);
```

```
ОткрытьФормуМодально ("Обработка.НастройкаКолонок", СписокПараметров);
```

Иначе

```
СписокКолонок = ВосстановитьЗначение
("СписокКолонок_Справочник_Квартиры_ФормаСписка_ДляВыбора");
```

```
ТекущаяКолонка = ВосстановитьЗначение
("ТекущаяКолонока_Справочник_Квартиры_ФормаСписка_ДляВыбоа");
```

Если ПустоеЗначение (СписокКолонок) = 1 **Тогда**

```
ПолныйСписокКолонок.Выгрузить (СписокКолонок);
```

```
СохранитьЗначение
```

```
("СписокКолонок_Справочник_Квартиры_ФормаСпискаДляВыбора",
СписокКолонок);
```

ИначеЕсли ТипЗначенияСтр (СписокКолонок) = "СписокЗначений" **Тогда**

Если СписокКолонок.РазмерСписка () <>

ПолныйСписокКолонок.РазмерСписка () **Тогда**

```
ПолныйСписокКолонок.Выгрузить (СписокКолонок);
```

```
СохранитьЗначение
```

```
("СписокКолонок_Справочник_Квартиры_ФормаСписка_ДляВыбора",
СписокКолонок);
```

КонецЕсли;

КонецЕсли;

Для Сч = 1 По СписокКолонок.РазмерСписка () Цикл

```
ИдКолонки = СписокКолонок.ПолучитьЗначение (Сч);
```

Попытка

```
Колонка = Форма.ПолучитьАтрибут (ИдКолонки);
```

Исключение

```
Сообщить (ИдКолонки);
```

КонецПопытки;

Если СписокКолонок.Пометка (Сч) = 1 **Тогда**

```
Колонка.Видимость (1);
```

Если ИдКолонки = ТекущаяКолонка **Тогда**

```
Форма.Активизировать (ИдКолонки, 0);
```

КонецЕсли;

Иначе

Колонка.Видимость (0) ;

КонецЕсли ;

КонецЦикла ;

КонецЕсли ;

КонецПроцедуры // УстановитьВидимостьКолонок

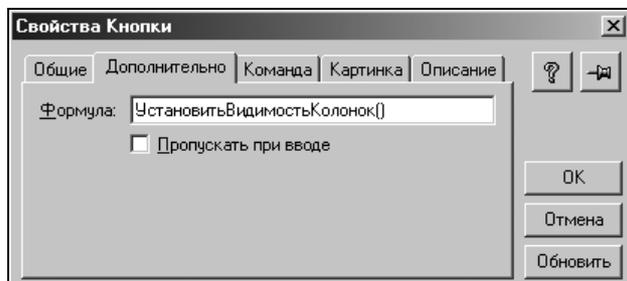


Рис. 3.57. Диалоговое окно свойства кнопки Колонки на вкладке Дополнительно

Процедура вызывается нажатием кнопки без параметров, поэтому выполняется сегмент с параметром ПриОткрытии = 0.

Создадим обработку **НастройкаКолонок** с тремя слоями (рис. 3.58).

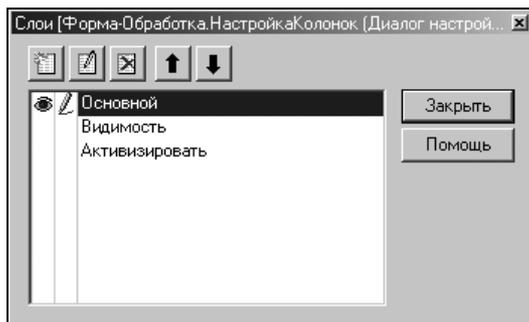


Рис. 3.58. Диалоговое окно слоев обработки НастройкаКолонок

Слой **Основной** содержит только кнопки (рис. 3.59).

Назначим процедуру Принять (Видим) кнопке **ОК** (рис. 3.60).

Слой **Активизировать** содержит список всех колонок (рис. 3.61), который передается в форму из процедуры УстановитьВидимостьКолонок().

Слой **Видимость** содержит список всех отмеченных колонок (рис. 3.62), который задается пользователем. Колонки этого списка будут видимыми в форме после нажатия кнопки **ОК**.

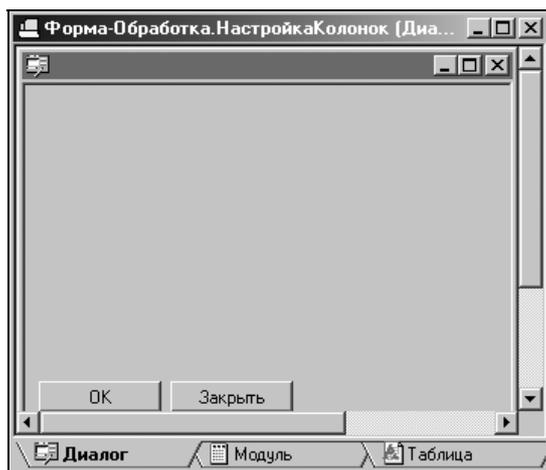


Рис. 3.59. Диалоговое окно обработки **НастройкаКолонок** (слой **Основной**)

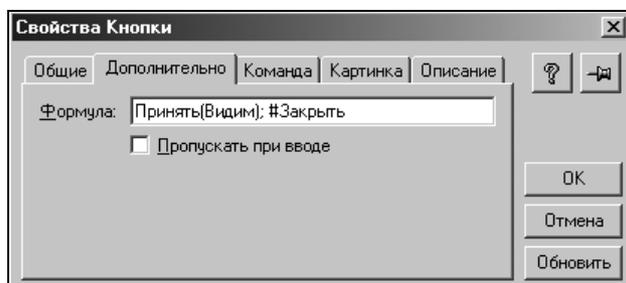


Рис. 3.60. Диалоговое окно свойств кнопки **OK** обработки **НастройкаКолонок** (слой **Основной**)

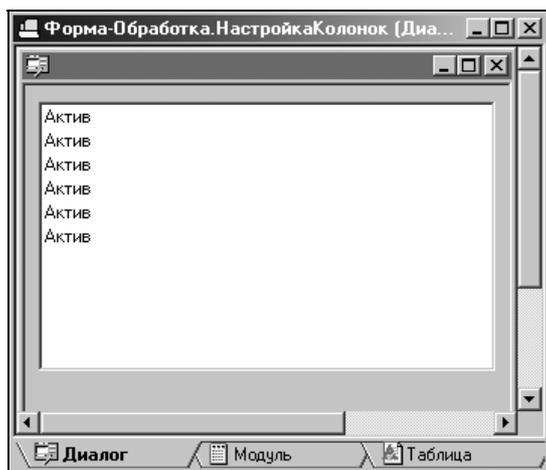


Рис. 3.61. Диалоговое окно обработки **НастройкаКолонок** (слой **Активизировать**)

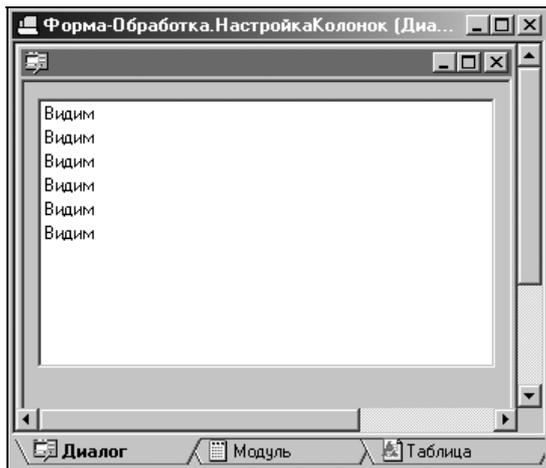


Рис. 3.62. Диалоговое окно обработки **НастройкаКолонок** (слой **Видимость**)

В модуле формы **НастройкаКолонок** определим процедуру `ПриОткрытии()`, которая будет выполняться после вызова метода `ОткрытьФормуМодально()` ("Обработка.НастройкаКолонок", `СписокПараметров`) модуля справочника "Квартиры":

Перем `СписокКолонок`, `ТекущаяКолонка`, `ИдКолонки`, `Колонка`;

Перем `Конт`, `ИдентификаторФормы`, `НачальныйСписок`, `ПриОткрытии`;

Перем `ВремТекКолонка`;

Процедура `ПриОткрытии()`

Если `ПустоеЗначение(Форма.Параметр)=1` **Тогда**

Возврат;

КонецЕсли;

`Конт` = `Форма.Параметр.ПолучитьЗначение(1)`;

`ИдентификаторФормы` = `Форма.Параметр.ПолучитьЗначение(2)`;

`НачальныйСписок` = `Форма.Параметр.ПолучитьЗначение(3)`;

`СписокКолонок`=`ВосстановитьЗначение("СписокКолонок_" + ИдентификаторФормы)`;

`ТекущаяКолонка`=`ВосстановитьЗначение("ТекущаяКолонока_" + ИдентификаторФормы)`;

Если `ПустоеЗначение(СписокКолонок) = 1` **Тогда**

`НачальныйСписок.Выгрузить(СписокКолонок)`;

`СохранитьЗначение("СписокКолонок_" + ИдентификаторФормы, СписокКолонок)`;

ИначеЕсли `ТипЗначенияСтр(СписокКолонок) = "СписокЗначений"` **Тогда**

Если `СписокКолонок.РазмерСписка() <>`

`НачальныйСписок.РазмерСписка()` **Тогда**

`НачальныйСписок.Выгрузить(СписокКолонок)`;

```
СохранитьЗначение ("СписокКолонок_" +
ИдентификаторФормы, СписокКолонок);
```

```
КонецЕсли;
```

```
КонецЕсли;
```

```
Форма.Заголовок ("Настройка колонок", 0);
```

```
Форма.ИспользоватьЗакладки (1);
```

```
Форма.Закладки.ДобавитьЗначение ("Видимость", "Видимость");
```

```
Форма.Закладки.ДобавитьЗначение ("Активизировать", "Активизировать при
открытии");
```

```
Форма.ИспользоватьСлой ("Основной, Видимость", 2);
```

```
СписокКолонок.Выгрузить (Видим);
```

```
ВремТекКолонка = Видим.ПолучитьЗначение (макс (Видим.НайтиЗначение
(ТекущаяКолонка), 1));
```

```
КонецПроцедуры // ПриОткрытии ()
```

Внимание!

Следует помнить, что переменные, которые используются во всем теле модуля в разных процедурах, определяются в начале модуля до определения первой процедуры.

Определим процедуру сохранения настройки `Принять ()`:

```
Процедура Принять (Видим);
```

```
Если Актив.ТекущаяСтрока () > 0 Тогда
```

```
ВремТекКолонка = Актив.ПолучитьЗначение (Макс
(Актив.ТекущаяСтрока (), 1));
```

```
КонецЕсли;
```

```
Если ПустоеЗначение (ВремТекКолонка) = 0 Тогда
```

```
ТекущаяКолонка = ВремТекКолонка;
```

```
КонецЕсли;
```

```
Для Сч = 1 По Видим.РазмерСписка () Цикл
```

```
ИдКолонки = Видим.ПолучитьЗначение (Сч);
```

```
Попытка
```

```
Колонка = Конт.Форма.ПолучитьАтрибут (ИдКолонки);
```

```
Исключение
```

```
Сообщить (ИдКолонки);
```

```
КонецПопытки;
```

```
Если Видим.Пометка (Сч) = 1 Тогда
```

```
Колонка.Видимость (1);
```

```
Если ИдКолонки = ТекущаяКолонка Тогда
```

```
Конт.Активизировать (ИдКолонки, 0);
```

```
КонецЕсли;
```

```
Иначе
```

```
Колонка.Видимость (0);
```

```
КонецЕсли;
```

```
КонецЦикла;
```

```
СохранитьЗначение ("СписокКолонок_" + ИдентификаторФормы, Видим) ;
СохранитьЗначение ("ТекущаяКолонока_" + ИдентификаторФормы,
    ТекущаяКолонка) ;
```

КонецПроцедуры // Принять ()

В приведенной процедуре ПриОткрытии () присутствует метод Форма.ИспользоватьЗакладки (1). В модуле формы нашей обработки **НастройкаКолонок** определим процедуру ПриВыбореЗакладки ():

```
Процедура ПриВыбореЗакладки (НомерСлоя, Слой)
    Форма.ИспользоватьСлой ("Основной, "+Слой, 2) ;
Если Слой = "Активизировать" Тогда
    Актив.УдалитьВсе () ;
Для Сч = 1 По Видим.РазмерСписка () Цикл
    Если Видим.Пометка (Сч) = 1 Тогда
        Представление = "";
        Значение = Видим.ПолучитьЗначение (Сч, Представление) ;
        Актив.ДобавитьЗначение (Значение, Представление) ;
    КонецЕсли ;
КонецЦикла ;
    Актив.ТекущаяСтрока (макс (Актив.НайтиЗначение (ВремГекКолонка), 1)) ;
Иначе
    ВремГекКолонка = Актив.ПолучитьЗначение (Макс
        (Актив.ТекущаяСтрока (), 1)) ;
КонецЕсли ;
КонецПроцедуры // ПриВыбореЗакладки ()
```

Глобальный модуль

Глобальный модуль — это, прежде всего, модуль основной стартовой формы программы. В Глобальном модуле определяются переменные, процедуры и функции, доступные для вызова из любых модулей форм и документов. Процедуры и функции, которые повторяются для ряда расчетов в других модулях, естественно определить один раз, настроив при этом соответствующим образом параметры и возвраты. Приведенный ранее вариант настройки колонок справочника "Квартиры" хотелось бы применить и к другим справочникам, не определяя в каждом модуле формы справочника процедуру УстановитьВидимостьКолонок (). Перенесем данную процедуру в Глобальный модуль, предварительно изменив настройку параметров.

Внимание!

Для лучшей читабельности текстов модулей названия процедур и функций в Глобальном модуле начинаются с префикса "гл", но это вовсе не означает син-

тактического ограничения на названия процедур Глобального модуля. На самом деле правила формирования названий процедур и функций в Глобальном модуле такие же, как и в остальных модулях форм и документов.

```
// УПРАВЛЕНИЕ ВИДИМОСТЬЮ КОЛОНОК В СПРАВОЧНИКАХ
// Параметры:
// Конт - контекст формы справочника, из которого вызвана процедура.
// ИдентификаторФормы - уникальный идентификатор формы, из которой
// выполнен вызов, используется для сохранения пользовательских
// настроек.
// НачальныйСписок - список колонок, видимость которых подлежит
// настройке
// ПриОткрытии - признак того, что вызов процедуры был сделан при
// открытии формы
Процедура глУстановитьВидимостьКолонок (Конт, ИдентификаторФормы,
    НачальныйСписок, ПриОткрытии=0) Экспорт
ПеремСписокКолонок, ТекущаяКолонка, ИдКолонки, Колонка;
ЕслиПриОткрытии=0Тогда
    СписокПараметров=СоздатьОбъект ("СписокЗначений");
    СписокПараметров.ДобавитьЗначение (Конт);
    СписокПараметров.ДобавитьЗначение (ИдентификаторФормы);
    СписокПараметров.ДобавитьЗначение (НачальныйСписок);
    ОткрытьФормуМодально ("Обработка.НастройкаКолонок", СписокПараметров);
Иначе
    СписокКолонок=ВосстановитьЗначение ("СписокКолонок_" +
        ИдентификаторФормы);
    ТекущаяКолонка=ВосстановитьЗначение ("ТекущаяКолонка_" +Идентификатор
        Формы);
ЕслиПустоеЗначение (СписокКолонок)=1Тогда
    НачальныйСписок.Выгрузить (СписокКолонок);
    СохранитьЗначение ("СписокКолонок_" +ИдентификаторФормы,
        СписокКолонок);
ИначеЕслиТипЗначенияСтр (СписокКолонок) ="СписокЗначений"Тогда
    ЕслиСписокКолонок.РазмерСписка () <>НачальныйСписок.РазмерСписка ()
        Тогда
        НачальныйСписок.Выгрузить (СписокКолонок);
        СохранитьЗначение ("СписокКолонок_" +ИдентификаторФормы,
            СписокКолонок);
КонецЕсли;
КонецЕсли;
ДляСч=1ПоСписокКолонок.РазмерСписка ()Цикл
    ИдКолонки = СписокКолонок.ПолучитьЗначение (Сч);
Попытка
    Колонка =Конт.Форма.ПолучитьАтрибут (ИдКолонки);
```

Исключение

Сообщить (ИдКолонки) ;

КонецПопытки;

Если СписокКолонок.Пометка (Сч) = 1 **Тогда**

Колонка.Видимость (1) ;

Если ИдКолонки = ТекущаяКолонка **Тогда**

Конт.Активизировать (ИдКолонки, 0) ;

КонецЕсли;**Иначе**

Колонка.Видимость (0) ;

КонецЕсли;**КонецЦикла;****КонецЕсли;**

КонецПроцедуры // глУстановитьВидимостьКолонок

Предопределенные процедуры Глобального модуля

Для стартовой формы программы, как и для любой другой формы, существуют предопределенные процедуры, которые определяются в Глобальном модуле:

- ПриНачалеРаботыСистемы();
- ПриЗавершенииРаботыСистемы().

Кроме того, в Глобальном модуле определяются предопределенные процедуры для форм, которые недоступны в Конфигураторе, т. е. формы истории и констант:

- ПриОткрытииИстории();
- ПриЗаписиИстории();
- ПриУдаленииИстории();
- ПриЗаписиКонстанты().

Ряд предопределенных процедур Глобального модуля предназначены для обработки интерактивных действий пользователя, влияющих на работу системы в целом:

- ПриУдаленииЭлемента();
- ПриУдаленииДокумента();
- ПриИзмененииВремениДокумента();
- ПриОтменеПроведенияДокумента();
- ПриУстановкеОтбора();
- ПриУдаленииСчета();
- ПриВыклВклПроводокОперации();

□ ПриИзмененииВремениДокумента ();

□ ПриСменеРасчетногоПериода ().

❖ Процедура ПриНачалеРаботыСистемы() выполняется при начале работы системы. Ее назначение — инициализация начальных значений некоторых справочников и констант при первом запуске, выполнение ряда действий при специфическом обновлении релиза программы, выполнение некоторых действий при анализе даты запуска, имени пользователя и т. п. Процедуру необходимо выполнять до начала работы пользователя с системой.

Рассмотрим анализ запуска программы на предмет первого запуска.

Процедура ПриНачалеРаботыСистемы()

ЭтоПервыйЗапуск = 0;

Если ПустоеЗначение(Константа.НомерРелиза) = 1 **Тогда**

 ЭтоПервыйЗапуск = 1;

 Док = СоздатьОбъект ("Документ");

Если Док.ВыбратьДокументы() = 1 **Тогда**

 ЭтоПервыйЗапуск = 0;

КонецЕсли;

Если ЭтоПервыйЗапуск = 1 **Тогда**

 ПоставляемыеСправочники = ",КлассификаторЕН,";

 Индекс = 1;

 КоличествоСправочников = Метаданные.Справочник();

Пока (ЭтоПервыйЗапуск=1) и (Индекс <= КоличествоСправочников) **Цикл**

 ВидСпр = Метаданные.Справочник(Индекс).Идентификатор;

Если Найти(ПоставляемыеСправочники, ","+ВидСпр+",") = 0 **Тогда**

 Спр = СоздатьОбъект ("Справочник."+ВидСпр);

Если Спр.ВыбратьЭлементы(0) = 1 **Тогда**

 ЭтоПервыйЗапуск = 0;

КонецЕсли;

КонецЕсли;

 Индекс = Индекс+1;

КонецЦикла;

КонецЕсли;

Если ЭтоПервыйЗапуск = 1 **Тогда**

 Константа.НомерРелиза = Лев(Метаданные.Комментарий, 8);

 ОткрытьФормуМодально ("Обработка.ПервыйЗапуск", 1);

Иначе

 Константа.НомерРелиза = "7.70.001";

КонецЕсли;

КонецЕсли;

ПервыйЗапускНаРабочемМесте = ВосстановитьЗначение
 ("ПервыйЗапускНаРабочемМесте");

Если ПустоеЗначение (ПервыйЗапускНаРабочемМесте) = 1 **Тогда**
 СохранитьЗначение ("ОсновнаяВалюта", Константа.ОсновнаяВалюта);
 СохранитьЗначение ("ОсновнойСклад", Константа.ОсновнойСклад);
 СохранитьЗначение ("ОсновнойБанковскийСчет",
 Константа.ОсновнойБанковскийСчет);
 СохранитьЗначение ("ОсновноеПодразделение",
 Константа.ОсновноеПодразделение);
 СохранитьЗначение ("ОсновнаяСтавкаНДС", Константа.ОсновнаяСтавкаНДС);
 СохранитьЗначение ("ОсновнаяСтавкаНП", Константа.ОсновнаяСтавкаНП);
 СохранитьЗначение ("ОсновнаяЕдиницаИзмерения",
 Константа.ОсновнаяЕдиницаИзмерения);
 СохранитьЗначение ("ПоказПутеводителяПриЗапуске", 1);
 СохранитьЗначение ("ПервыйЗапускНаРабочемМесте", 1);
 РазворотСубконто = СоздатьОбъект ("ТаблицаЗначений");
 РазворотСубконто.НоваяКолонка ("Счет",,,, "Счет", 16);
 РазворотСубконто.НоваяКолонка ("Субсчета",,,, "Субсчета", 9);
 РазворотСубконто.НоваяКолонка ("Субконто",,,, "Субконто", 9);
 РазворотСубконто.НоваяКолонка ("Список");
 РазворотСубконто.НоваяСтрока ();
 РазворотСубконто.Счет = СчетПоКоду (51);
 РазворотСубконто.Субсчета = "-";
 РазворотСубконто.Субконто = "X";
 РазворотСубконто.Список = СоздатьОбъект ("СписокЗначений");
 РазворотСубконто.Список.ДобавитьЗначение (ВидыСубконто.БанковскиеСчета);
 РазворотСубконто.Список.Пометка (1, 1);
 СохранитьЗначение ("ОСВРазворотСубконто", РазворотСубконто);
 СохранитьЗначение ("ОСВДанныеПоСубсчетам", 1);
 СохранитьЗначение ("ОСВЗабалансовыеСчета", 1);

КонецЕсли;

Если Константа.НомерРелиза <> Лев (Метаданные.Комментарий, 8) **Тогда**
 ОткрытьФорму ("Обработка.ОбновлениеИБ");

КонецЕсли;

ЗаголовковСистемы (СокрЛП (Константа.НазваниеОрганизации));

КонецПроцедуры // ПриНачалеРаботыСистемы ()

❖ Процедура ПриЗавершенииРаботыСистемы () выполняется при завершении работы пользователя. Рассмотрим ее на предмет запроса на подтверждение окончания работы. Пользователь часто "промахивается", нажимая на кнопку закрытия программы вместо кнопки закрытия текущего окна в полноэкранном режиме.

Процедура ПриЗавершенииРаботыСистемы ()

Перем ЗапрашиватьПодтверждениеПриВыходе;

```
ЗапрашиватьПодтверждениеПриВыходе =
ПустоеЗначение ( ВосстановитьЗначение
("НеЗапрашиватьПодтверждениеПриВыходе") );
```

Если ЗапрашиватьПодтверждениеПриВыходе = 1 **Тогда**

```
    Если Вопрос ("Закончить работу с программой?", "Да+Нет", 60) = "Нет"
```

Тогда

```
    СтатусВозврата (0) ;
```

```
    Возврат ;
```

```
    КонецЕсли ;
```

```
    КонецЕсли ;
```

```
КонецПроцедуры // ПриЗавершенииРаботыСистемы()
```

❖ **Процедура** ПриУдаленииЭлемента (<УдалЭлем>, <Режим>) выполняется при интерактивном удалении элемента справочника.

Параметры:

□ УдалЭлем — удаляемый элемент;

□ Режим — режим удаления:

- 1 — непосредственное удаление;
- 0 — пометка на удаление.

Данная предопределенная процедура практически не используется, т. к. механизм удаления элементов справочника с отслеживанием подчинения и участия в качестве субконто в проводках достаточно хорошо организован в программе. Однако бывают ситуации, когда необходимо организовать непосредственное удаление элементов.

Рассмотрим простейший пример:

Процедура ПриУдаленииЭлемента (УдалЭлем, Режим)

```
    Если УдЭл.ПометкаУдаления () = 0 Тогда
```

```
        Если Вопрос ("Объект будет удален безвозвратно! Удалить?", 1, 30) = 1
```

тогда

```
            Сп=СоздатьОбъект ("Справочник." + УдЭл.ПредставлениеВида ());
```

```
            Сп.НайтиПоКоду (УдЭл.Код, 0) ;
```

```
            Сп.Удалить (1) ;
```

```
        КонецЕсли ;
```

```
    КонецЕсли ;
```

```
КонецПроцедуры // ПриУдаленииЭлемента
```

❖ **Процедура** ПриОткрытииИстории (<ТипОбъекта>, <Объект>, <ТолькоПросмотр>) выполняется при открытии окна истории значения периодического реквизита элемента справочника или периодической константы.

Параметры:

- ❑ <ТипОбъекта> — строка с названием периодического объекта конфигурации (периодического реквизита справочника или периодической константы);
- ❑ <Объект> — элемент справочника, для которого выполняется открытие окна истории периодического реквизита (для константы не используется);
- ❑ <ТолькоПросмотр> — флаг установки режима **Только просмотр** для окна истории значения. Если значение этого параметра установить в 1 (в предопределенной процедуре), то окно истории будет открыто только для чтения. Установка значения в 0 введет режим по умолчанию (определенный правами). Открытое окно истории соответствующим образом изменит режим (если это случай, когда процедура обрабатывает не открытие окна, а смену отображаемого объекта). Значение по умолчанию режима **Только просмотр** — 0.

❖ **Процедура** ПриЗаписиИстории (<ТипОбъекта>, <Объект>, <Значение>, <ДатаИстории>) выполняется при записи в окне истории значения периодического реквизита элемента справочника или периодической константы.

Параметры:

- ❑ <ТипОбъекта> — строка с названием периодического объекта конфигурации (периодического реквизита справочника или периодической константы);
- ❑ <Объект> — элемент справочника, у которого выполняется изменение значения периодического реквизита (для константы не используется);
- ❑ <Значение> — введенное значение;
- ❑ <ДатаИстории> — дата, на которую введено значение.

Рассмотрим пример контроля записи в историю, если пользователем не было задано конкретное значение. Пользователь должен подтвердить запись в историю нулевого значения константы **ПределСуммыДолга**:

Процедура ПриЗаписиИстории (ТипОбъекта, Объект, Значение, ДатаИстории)

Если ТипОбъекта = "Константа.ПределСуммыДолга" **Тогда**

Если ПустоеЗначение (Значение) = 1 **Тогда**

Предупреждение ("Сумма долга не может равняться 0!", 30);

СтатусВозврата (0);

КонецЕсли;

КонецЕсли;

КонецПроцедуры // ПриЗаписиИстории ()

❖ **Процедура** ПриУдаленииИстории (<ТипОбъекта>, <Объект>, <Значение>, <ДатаИстории>) выполняется при удалении в окне истории значения периодического реквизита элемента справочника или периодической константы.

Параметры:

- ❑ <ТипОбъекта> — строка с названием периодического объекта конфигурации (периодического реквизита справочника или периодической константы);
- ❑ <Объект> — элемент справочника, у которого выполняется удаление значения периодического реквизита (для константы не используется);
- ❑ <Значение> — удаляемое значение;
- ❑ <ДатаИстории> — дата, на которую было записано значение.

❖ **Процедура** ПриЗаписиКонстанты (<ИмяКонстанты>, <Значение>) выполняется при интерактивном вводе значения константы.

Параметры:

- ❑ <ИмяКонстанты> — строка с названием константы;
- ❑ <Значение> — введенное значение.

Процедура применяется аналогично процедуре ПриЗаписиИстории(), например:

Процедура ПриЗаписиКонстанты(ИмяКонстанты, Значение)

Если ИмяКонстанты = "НомерРелиза" **Тогда**

Предупреждение("Константа ""Номер релиза"" является служебной и
не может быть изменена вручную.");

СтатусВозврата(0);

КонецПроцедуры // ПриЗаписиКонстанты

❖ **Процедура** ПриУдаленииДокумента (<УдалДокум>, <Режим>) выполняется при интерактивном удалении документа.

Параметры:

- ❑ <УдалДокум> — удаляемый документ;
- ❑ <Режим> — режим удаления:
 - 1 — непосредственное удаление;
 - 0 — пометка на удаление.

Используем процедуру ПриУдаленииДокумента() для организации запрета удаления документа:

Процедура ПриУдаленииДокумента(Докум, Режим)

Если Докум.ДатаДок <= Константа.ДатаЗапретаРедактирования **Тогда**

СтатусВозврата (0) ;

Если Докум.ПометкаУдаления() = 0 **Тогда**

Предупреждение ("Нельзя удалять документы с датой, более ранней, чем дата запрета редактирования документов!");

Иначе

Предупреждение ("Нельзя отменять удаление документов с датой, более ранней, чем дата запрета редактирования документов!");

КонецЕсли ;

КонецЕсли ;

КонецПроцедуры

❖Процедура ПриИзмененииВремениДокумента (<Докум>) выполняется при интерактивном изменении времени документа. Обычно применяется с компонентой **Оперативный учет**.

Параметр <Докум> — обрабатываемый документ.

❖Процедура ПриОтменеПроведенияДокумента (<Докум>) выполняется при интерактивной отмене проведения документа.

Параметр <Докум> — обрабатываемый документ.

Используем процедуру ПриУдаленииДокумента() для организации запрета отмены проведения документа:

Процедура ПриОтменеПроведенияДокумента (Докум)

Если Докум.ДатаДок <= Константа.ДатаЗапретаРедактирования **Тогда**

Предупреждение ("Нельзя отменять проведение документов с датой, более ранней, чем дата запрета редактирования документов!");

СтатусВозврата (0) ;

Возврат ;

КонецЕсли ;

КонецПроцедуры

❖Процедура ПриУстановкеОтбора (<ИмяРеквизОтбора>, <Значение>) выполняется при интерактивной установке отбора документов в журнале.

Параметры:

□ <ИмяРеквизОтбора> — строка с названием общего реквизита документа, по которому производится отбор;

□ <Значение> — значение реквизита отбора.

Процедуру можно использовать для задания дополнительного условия отбора.

❖ Процедура `ПриУдаленииСчета (<УдалСчет>, <Режим>)` выполняется при удалении бухгалтерского счета.

Параметры:

- `<УдалСчет>` — значение типа `Счет` — удаляемый бухгалтерский счет;
- `<Режим>` — режим удаления. Может принимать значения:
 - 1 — счет будет удален;
 - 0 — счет будет помечен на удаление.

❖ Процедура `ПриВыклВклПроводокОперации (<Документ>)` выполняется при выключении или включении бухгалтерских проводок.

Параметр `<Документ>` — значение типа `Документ` — документ, которому принадлежит операция.

Используем процедуру `ПриУдаленииДокумента ()` для организации запрета выключения и включения проводок:

Процедура `ПриВыклВклПроводокОперации (Докум)`

Если `Докум.ДатаДок <= Константа.ДатаЗапретаРедактирования Тогда`
`СтатусВозврата (0) ;`

Если `Докум.Операция.ВключитьПроводки () = 0 Тогда`

Предупреждение ("`Нельзя включить проводки документов с датой, более ранней, чем дата запрета редактирования документов!`") ;

Иначе

Предупреждение ("`Нельзя выключить проводки документов с датой, более ранней, чем дата запрета редактирования документов!`") ;

КонецЕсли ;

КонецЕсли ;

КонецПроцедуры

❖ Процедура `ПриСменеРасчетногоПериода (<ЖурналРасчетов>, <Период>)` выполняется при смене текущего расчетного периода журнала расчетов.

Параметры:

- `<ЖурналРасчетов>` — журнал расчетов, период которого изменяется (агрегатный объект типа `ЖурналРасчетов`);
- `<Период>` — устанавливаемый расчетный период (значение типа `ПериодРасчета`).

Организуем запуск помощника для смены периода:

Процедура `ПриСменеРасчетногоПериода (ЖурналРасчетов, Период)`

Если `ВидЖурнала = "Начисления" Тогда`

Если ЖурналРасчетов.КонецТекущегоПериода () +1=Период.ДатаНачала **Тогда**
 ОткрытьФорму ("Обработка.ПомощникПереходаНаСледующийПериод");
 СтатусВозврата (0);

КонецЕсли;

КонецЕсли;

КонецПроцедуры // ПриСменеРасчетногоПериода

❖ **Процедура** ОбработкаЯчейкиТаблицы (<Значение>, <ФлагСтандОбраб>, <Таблица>, <Адрес>) обработки ячейки таблицы выполняется двойным щелчком мыши или нажатием клавиши <Enter> в табличном документе на выбранной ячейке.

Параметры:

- <Значение> — здесь в процедуру передается вычисленное значение ячейки (в Конфигураторе задается: "Свойства", "Текст", "Значение");
- <ФлагСтандОбраб> — флаг обработки ячейки: установка в 1 приведет к выполнению стандартной обработки значения ячейки по завершении процедуры (открытию документа, элемента справочника и т. п.);
- <Таблица> — имя переменной (необязательный параметр), куда система передаст объект типа Таблица. С помощью значения этого контекста можно произвольно манипулировать данной таблицей, пока она открыта. Пока данный объект "Таблица" существует, тип значения данного параметра равен 100, если закрыт — 0;
- <Адрес> — имя переменной (необязательный параметр), куда система передаст адрес ячейки/объекта в формате "R1C1:R2C2".

Внимание!

Объект типа СписокЗначений' может записываться в поле "Значение" ячейки таблицы и использоваться затем процедурой ОбработкаЯчейкиТаблицы. Если данная процедура описана в модуле формы, то вызывается именно она, иначе система запускает одноименную процедуру из Глобального модуля.

Данная предопределенная процедура НЕ вызывается при выборе ячейки таблицы в режиме ввода данных. Для этого случая вызывается предопределенная процедура ПриВыбореЯчейкиТаблицы.

Приведем в качестве примера процедуру ОбработкаЯчейкиТаблицы () из типовой конфигурации, как наиболее иллюстративный вариант:

Процедура ОбработкаЯчейкиТаблицы (Расшифровка, СтандартнаяОбработка, Таблица) Экспорт

Перем Отчет;

Если ТипЗначенияСтр (Расшифровка) <> "СписокЗначений" **Тогда**

СтандартнаяОбработка = 1;

```

    Возврат;
КонецЕсли;
// Расшифровка отчета
Док = Расшифровка.Получить ("Документ");
Если ТипЗначения (Док) <> 0 Тогда
    ОткрытьФорму ("Операция", , Док,
Расшифровка.Получить ("НомерПроводки"),
    Расшифровка.Получить ("НомерКорреспонденции"), -1);
Иначе
    Меню = Расшифровка.Получить ("Меню");
Если ТипЗначенияСтр (Меню) = "СписокЗначений" Тогда
    Если Меню.ВыбратьЗначение (Отчет, "", , 1) = 0 Тогда
        Возврат;
    КонецЕсли;
Иначе
    Отчет = Расшифровка.Получить ("Отчет");
КонецЕсли;
глРасшифровка = Расшифровка;
глФлагРасшифровки = 1;
глОбновить = Число (Расшифровка.Получить ("Обновить"));
Если глОбновить <> 0 Тогда
    глТаблица = Таблица;
КонецЕсли;
Если Отчет = "ОборотноСальдоваяВедомостьПоСчету" Тогда
    Счет = Расшифровка.Получить ("Счет");
    Если Счет.КоличествоСубконто () = 0 Тогда
        Расшифровка.Установить ("ДанныеПоСубсчетам",
Счет.ЭтоГруппа ());
    КонецЕсли;
КонецЕсли;
Если Расшифровка.Получить ("ОткрытьЖурналДокументов") = 1 Тогда
    Док = Расшифровка.Получить ("ДокументЖурнала");
    Конт = ПолучитьПустоеЗначение ("");
    ОткрытьФорму ("Журнал.Общий#ЖДОБ", Конт);
    Конт.УстановитьИнтервал (Док.ДатаДок - 30, Док.ДатаДок + 30);
    Конт.АктивизироватьОбъект (Док);
Иначе
    Попытка
        Если Метаданные.Отчет (Отчет).Выбран () = 1 Тогда
            ОткрытьФорму ("Отчет."+Отчет+"#");
        ИначеЕсли Метаданные.Обработка (Отчет).Выбран () = 1 Тогда
            ОткрытьФорму ("Обработка."+Отчет+"#");
        ИначеЕсли ФС.СуществуетФайл (Отчет) = 1 Тогда
            ОткрытьФорму ("Отчет", , Отчет);
        КонецЕсли;

```

Исключение

Сообщить (ОписаниеОшибки ()) ;

КонецПопытки ;**КонецЕсли ;**

глФлагРасшифровки = 0 ;

глРасшифровка = 0 ;

глОбновить = 0 ;

КонецЕсли ;**КонецПроцедуры** // ОбработкаЯчейкиТаблицы

Навигация в теле модуля

Тело модуля формы или документа порой содержит множество процедур и функций со сложной вложенной структурой. Структура модуля представлена в специальном окне **Процедуры и функции модуля** (рис. 3.63).

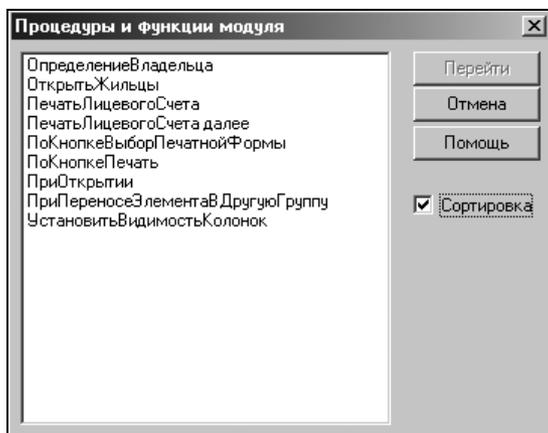


Рис. 3.63. Диалоговое окно выбора процедур и функций модуля формы списка справочника "Квартиры"

Окно вызывается из меню **Действия** пункт **Процедуры и функции модуля** в режиме редактирования модуля. Двойным щелчком мыши на выбранном пункте указатель устанавливается на начало соответствующей процедуры или функции.

Синтакс-Помощник

Синтакс-Помощник — средство, облегчающее разработку модулей. Основная задача Синтакс-Помощника — предоставить специалисту, выполняюще-

му конфигурирование системы 1С:Предприятие, оперативную подсказку по встроенному языку.

Для вызова Синтакс-Помощника выберите пункт **Синтакс-Помощник** в меню **Сервис** главного меню Конфигуратора.

Окно Синтакс-Помощника состоит из двух частей. В верхней части в виде дерева выдается список элементов встроенного языка системы 1С:Предприятие: операторов, управляющих конструкций, процедур и функций, системных констант и др. Для удобства все элементы встроенного языка объединены в тематические разделы, представленные в виде ветвей дерева. Помимо элементов встроенного языка, дерево в верхней части окна Синтакс-

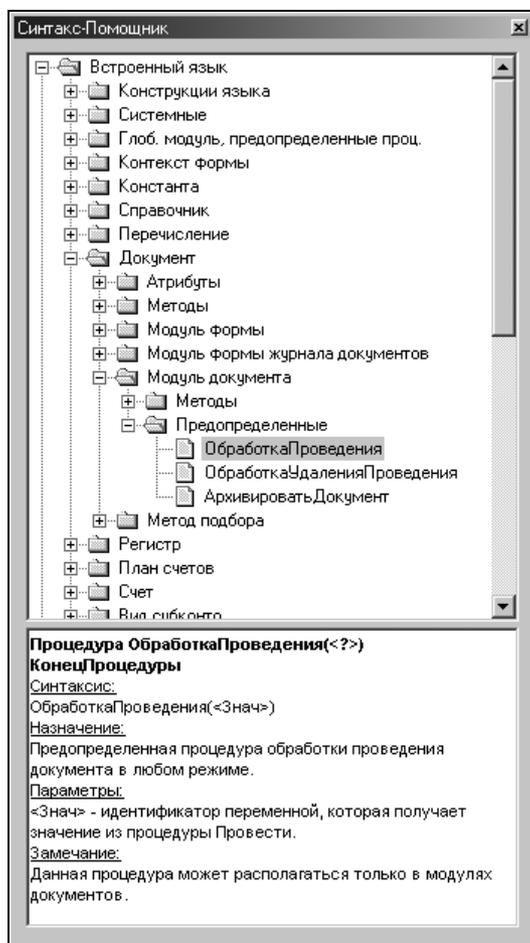
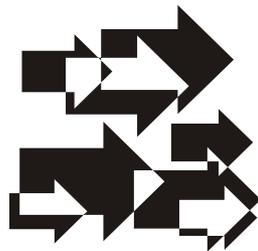


Рис. 3.64. Диалоговое окно Синтакс-Помощник

Помощника содержит список существующих шаблонов. В нижней части окна Синтакс-Помощника выдается краткое описание элемента встроенного языка, выбранного в верхнем окне. Для получения описания следует дважды щелкнуть мышью на наименовании элемента языка в верхней части окна Синтакс-Помощника. Для нахождения описания предопределенных процедур модуля документа необходимо раскрыть пункт **Встроенный язык**, затем раскрыть пункт **Документ**, пункт **Модуль документа** и пункт **Предопределенные** (рис. 3.64). Для получения описания необходимо дважды щелкнуть мышью на выбранной процедуре.

ГЛАВА 4



Назначение метаданных

В главах с 1 по 3 мы упражнялись в освоении техники задания интерфейса с пользователем. Уровень, который мы освоили, можно сравнить с конструированием ребенком случайных уродцев из кубиков. Настоящее конструирование только начинается. В конфигурации определены типы метаданных, назначение которых пока неясно, тем более неясны их место и роль в будущей конфигурации. В данной главе мы с учетом приобретенных навыков по шагам пройдем построение конфигурации от постановки задачи до реализации в виде конфигурации.

Предполагается, что хозяйственная деятельность торгового предприятия может быть описана данными следующих видов.

- *Константы* — постоянные (условно-постоянные) величины. Константы хранят информацию, которая не изменяется или изменяется достаточно редко: название организации, ее почтовый адрес и т. д.
- *Справочники* — списки однородных элементов данных. Используются для хранения нормативно-справочной информации.
- *Документы* — служат для ввода информации о совершенных хозяйственных операциях.
- *Журналы* — списки объектов данных типа Документ. Служат для работы с документами.
- *Перечисления* — списки значений, задаваемые на этапе конфигурирования.
- *Отчеты* — средство получения выходной информации. Источником данных для построения отчетов служат документы, справочники и регистры, используется информация, хранящаяся в константах.
- *Обработка* — используется для выполнения различных действий над информационной базой.

- *Регистры* — средство накопления оперативной информации о наличии и движении средств.
- *Журналы расчетов* — средство проведения периодических расчетов и хранения результатов расчета прошлых периодов. Каждый журнал связан с одним из справочников, который называется справочником объектов расчета. Каждая строка журнала — единичный акт расчета, который производится для определенного объекта по определенному алгоритму, имеет протяженность во времени и результат.
- *Виды расчетов* — алгоритмы, по которым могут проводиться расчеты записей журналов расчета.
- *Группы расчетов* — средство логического объединения видов расчета по тем или иным признакам.
- *Календари* — средство построения графиков работы предприятия, внешних организаций или отдельных категорий работников.

Создание структуры данных производится в окне **Конфигурация** на вкладке **Метаданные**. Окно вызывается на экран выбором пункта **Открыть конфигурацию** в меню **Конфигурация** главного меню Конфигуратора. Структура данных представлена в виде дерева и позволяет просматривать и редактировать характеристики всех типов данных.

Для создания нового объекта метаданных необходимо выполнить следующие действия:

1. В дереве метаданных выделите наименование типа метаданных или любого из существующих объектов метаданных того типа, который должен быть у создаваемого объекта.
2. Нажатием правой кнопки мыши вызовите контекстное меню.
3. В контекстном меню выберите пункт **Новый...** ("...справочник", "...документ", "...реквизит" и т. п.); конкретное наименование этого пункта зависит от типа создаваемого объекта метаданных.

На экран будет выдан пустой диалог для определения характеристик типа данных. Вид диалога и набор задаваемых в нем характеристик зависят от типа данных. Редактирование объекта метаданных выполняется в специальном диалоге. Для вызова этого диалога на экран необходимо дважды щелкнуть левой кнопкой мыши на идентификаторе объекта метаданных, свойства которого требуется отредактировать.

Для удаления объекта метаданных необходимо выполнить следующие действия:

1. В окне **Конфигурация** на вкладке **Метаданные** выделите объект метаданных, который предполагается удалить.
2. Нажмите клавишу .

В выданном на экран запросе подтвердите необходимость удаления объекта метаданных. После подтверждения указанный объект метаданных будет удален. Если удаляемый объект метаданных используется другими объектами, на экран будет выдано предупреждение, и выбранный объект метаданных удален не будет. Если действительно необходимо удалить выбранный объект метаданных, следует исключить его использование другими объектами. Для поиска ссылок на удаляемый объект метаданных из других объектов можно использовать информацию из сообщений, выдаваемых при попытке удаления объекта метаданных.

Внимание!

Следует иметь в виду, что физическое удаление объекта метаданных из структуры метаданных и реорганизация информационной базы происходят при выполнении процедуры сохранения сделанных изменений. Поэтому, если эта процедура еще не выполнялась, существует возможность восстановить удаленный объект метаданных. Для восстановления удаленного объекта метаданных следует закрыть окно **Конфигурация**, отказавшись от сохранения сделанных изменений и реорганизации информационной базы, и затем вновь открыть это окно. При этом следует учитывать, что будут утеряны все изменения в метаданных, которые были сделаны с момента предыдущего сохранения.

Наименования элементов данных можно расставить в требуемом порядке. Для этого используйте кнопки  (Переместить вниз),  (Переместить вверх) и  (Упорядочить) на панели инструментов окна **Метаданные**.

В задании объектов метаданных определяется ряд полезных зависимостей, предназначенных для ускорения поиска при выборе элементов справочников, быстрое заполнение документов на основании уже существующих и т. д. Зависимости объектов метаданных задаются в свойствах объектов метаданных.

Справочники

Справочники представляют собой списки однородных элементов данных. В нашей конфигурации такими объектами являются дома, квартиры, жильцы, услуги по эксплуатации дома, а также коммунальные услуги. Кроме того, некоторые справочники служат для упрощения расчетов, а также используются для хранения нормативно-справочной информации. Рассмотрим справочник "Жильцы" с точки зрения его взаимодействия с другими справочниками (рис. 4.1).

Кроме свойств реквизитов пользователя, которые задаются в группе **Реквизиты**, необходимо определить свойства полей, которые всегда определены для справочника.

Справочник Жильцы

Идентификатор: Жильцы Подчинен: Квартиры

Комментарий: Жильцы Синоним: Жильцы

Кол-во. уровней: 2 Размещать группы сверху

Длина кода: 5 Автоматическая нумерация

Длина наименования: 25 Контроль уникальности

Серию кодов

Во всем справочнике

В пределах подчинения

Тип кода

Числовой

Текстовый

Основное представление

В виде кода

В виде наименования

Реквизиты:

- ФИО
- Паспорт
- ДатаРождения
- СтатусПрописки
- СемейныйСтатус
- ДатаПрописки
- ДатаВыбытия
- Контрагент

Новый Изменить Удалить

Одна форма для элемента и группы Редактировать: В диалоге

Описание Форма элемента Форма группы Формы списка

Рис. 4.1. Диалоговое окно Справочник Жильцы

Поле **Идентификатор** определяет имя справочника. Поле представляет собой строку длиной не более 128 символов, состоящую из букв, цифр и знаков подчеркивания. Строка не должна начинаться с цифры и содержать пробелы. Идентификатор позволяет обращаться к справочнику из модулей.

Поле **Комментарий** для расшифровки идентификатора. Иногда **Идентификатор** не позволяет задавать логически связанное наименование.

Поле **Синоним** представляет произвольную последовательность символов и задает представление имени справочника во всех пользовательских диалогах и списках вместо идентификатора.

Поле **Кол-во уровней** задает количество уровней вложенности в многоуровневых справочниках. Использование многоуровневых справочников дает возможность получать сводную информацию с различной степенью детализации.

Поле **Длина кода** задает количество символов для кода элемента справочника.

Поле **Размещать группы сверху** задает признак сортировки справочника относительно групп. Если флажок установлен, то при работе с системой 1С:Предприятие при отображении справочника в виде иерархического списка группы окажутся в верхних строчках списка, а элементы справочника будут

располагаться ниже. Если этот флажок снят, расположение групп и элементов будет подчиняться установленным правилам сортировки (по коду, наименованию и пр.).

Поле **Длина наименования** задает максимально возможную длину наименования элемента справочника.

Поле **Автоматическая нумерация** задает признак формирования кода нового элемента справочника автоматически.

Поле **Контроль уникальности** устанавливает диапазон проверки уникальности кода. Если установлен признак **Во всем справочнике**, код элемента справочника должен быть уникален в пределах всего справочника. Если установлен признак **В пределах подчинения**, код элемента справочника должен быть уникален в пределах своего уровня вложенности.

Переключатель **Тип кода** задает тип значения для кода элемента справочника:

- Числовой** — в качестве кодов элементов справочника используются только числа;
- Текстовый** — в качестве кода элемента справочника может использоваться произвольная комбинация символов.

Переключатель **Основное представление** устанавливает реквизит справочника, который будет отображаться в качестве значения реквизита типа Справочник (например, в документе):

- В виде кода** — реквизит будет выдаваться в виде кода элемента справочника;
- В виде наименования** — реквизит будет выдаваться в виде наименования элемента справочника.

Признак **Одна форма для элемента и группы** определяет использование одной формы для элемента справочника и группы. Данный флажок имеет смысл включать в тех случаях, когда пользовательских реквизитов нет.

Поле **Редактировать** задает способ редактирования элемента при работе со справочником:

- В списке** — редактирование элементов справочника осуществляется только в ячейке таблицы справочника;
- В диалоге** — редактирование элементов справочника осуществляется только в диалоге;
- Обоими способами** — редактирование элементов справочника может осуществляться по выбору: либо непосредственно в ячейке таблицы справочника, либо в диалоге.

Подчинение

Конфигуратор системы 1С:Предприятие позволяет устанавливать отношения подчинения между справочниками. В нашей конфигурации существует справочник квартир. Для задания подчиненного ему справочника "Жильцы" необходимо в поле **Подчинен** задать идентификатор справочника-хозяина, в данном случае справочника "Квартиры". При необходимости выбрать элемент из справочника "Жильцы" система 1С:Предприятие сначала предложит выбрать квартиру, а после этого выдаст для выбора список жильцов этой квартиры. На практике удобнее задавать два элемента диалога: элемент диалога типа справочника-хозяина и элемент диалога типа подчиненного справочника. В нашем случае элементом типа справочника-хозяина является элемент типа **Справочник.Квартиры** (рис. 4.2).

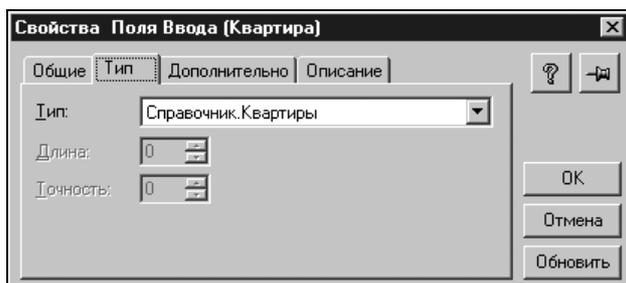


Рис. 4.2. Диалоговое окно свойств элемента типа Справочник.Квартиры

Для того чтобы в элементе типа подчиненного справочника не предлагался для выбора справочник-хозяин, необходимо на вкладке **Дополнительно** диалогового окна свойств элемента в поле **Связан с** задать идентификатор элемента типа справочника-хозяина, в нашем случае — **Квартира** (рис. 4.3).

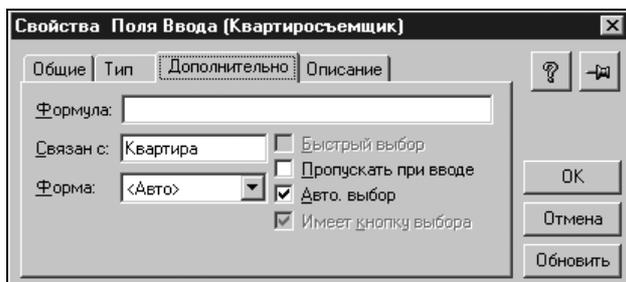


Рис. 4.3. Диалоговое окно свойств элемента типа Справочник.Жильцы

Реквизиты

Любой справочник можно представить как таблицу, которая имеет две обязательных графы: код элемента справочника и его наименование. 1С:Предприятие позволяет устанавливать для элемента справочника дополнительный набор характеристик, которые называются реквизитами. При работе со справочником реквизиты будут изображаться как дополнительные колонки в справочнике.

Управляющие элементы группы "Реквизиты" служат для управления списком реквизитов справочника и определения их свойств.

Управляющий элемент **Новый** вызывает на экран диалог для задания свойств нового реквизита справочника — краткого и полного наименования, типа значения, длины, других характеристик.

Управляющий элемент **Изменить** открывает заполненный диалог для изменения свойств реквизита справочника.

Управляющий элемент **Удалить** удаляет реквизит из списка.

Управляющий элемент  сдвигает реквизит в списке на одну позицию вверх.

Управляющий элемент  сдвигает реквизит в списке на одну позицию вниз.

Управляющий элемент  сортирует список реквизитов по алфавиту.

Управляющий элемент **Описание** вызывает текстовый редактор для создания пользовательского описания справочника.

Управляющий элемент **Форма элемента** вызывает редактор форм для создания формы ввода элемента в справочник.

Управляющий элемент **Форма группы** вызывает редактор форм для создания формы для ввода и редактирования реквизитов группы в справочнике.

Управляющий элемент **Форма списков** открывает окно со списком диалогов, позволяет редактировать формы выдачи справочника на экран для просмотра и редактирования и установить, в какой форме справочник будет выдаваться на экран при просмотре или при выборе из справочника какого-либо элемента.

При нажатии кнопки **Формы списков** предлагается меню со списком всех существующих форм справочника и пунктом **Редактировать**. При создании нового справочника всегда предлагается основная форма справочника с названием по умолчанию **ФормаСписка** (рис. 4.4).

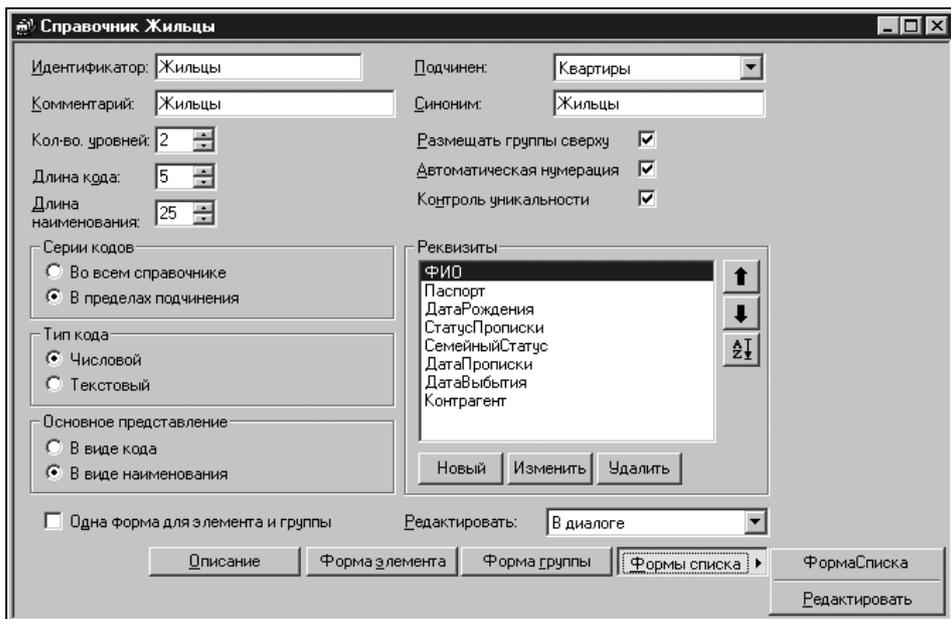


Рис. 4.4. Диалоговое окно справочника "Жильцы" с меню кнопки **Формы списка**

Часто для выбора элемента справочника необходима либо более сокращенная форма, чем основная, либо расширенная для подбора элементов. Например, в справочнике материалов или номенклатуры при подборе элементов необходимо отображать остатки на складе. Для задания дополнительных форм справочника или редактирования свойств существующих предназначен диалог **Формы списков**, который вызывается выбором в меню кнопки **Формы списка** пункта **Редактировать**.

Диалоговое окно **Формы списков** (рис. 4.5) содержит список форм для показа справочника, а также кнопки для управления списком, вызова редактора форм и выбора момента использования конкретной формы справочника.

При нажатии кнопки **Новый** на экран будет выдан запрос, в который требуется ввести идентификатор и комментарий создаваемой формы списка (рис. 4.6).

Нажатие кнопки **Изменить** вызывает на экран запрос для редактирования идентификатора и комментария формы списка (рис. 4.6).

Нажатие кнопки **Удалить** удаляет форму из списка.

Нажатие кнопки **Закончить** закрывает диалог.

Нажатие кнопки **Основная** назначает выбранную форму списка для использования справочника для просмотра или редактирования его значения. Название формы выдается в рамке слева от кнопки.

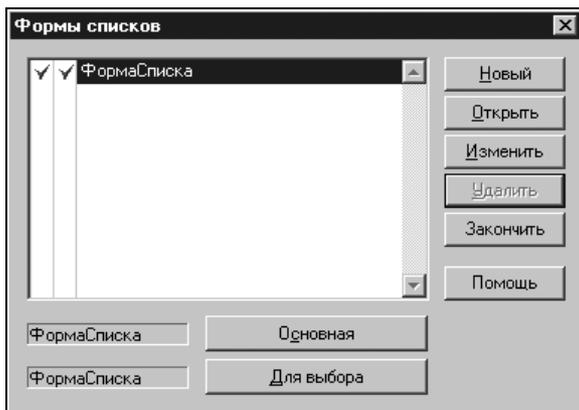
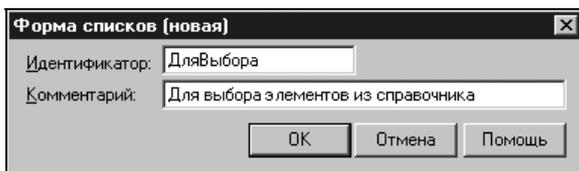
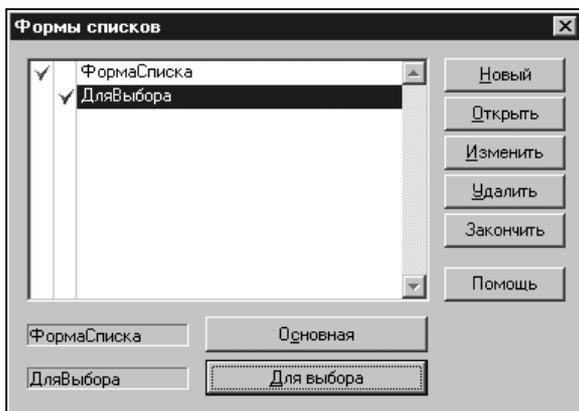
Рис. 4.5. Диалоговое окно **Формы списков**

Рис. 4.6. Диалоговое окно задания наименования формы

Нажатие кнопки **Для выбора** назначает выбранную форму списка для использования справочника для выбора из него требуемого элемента. Название формы выдается в рамке слева от кнопки.

Нажатие кнопки **Открыть** вызывает редактор форм для редактирования выбранной формы списка.

Рис. 4.7. Диалоговое окно **Формы списков** после задания формы для выбора

В диалоге **Формы списков** в первой колонке галочкой отмечена основная форма списка, во второй колонке — форма для выбора (рис. 4.7).

В меню кнопки **Формы списков** появится пункт с именем заданной формы для вызова диалогового окна редактирования данной формы (рис. 4.8).

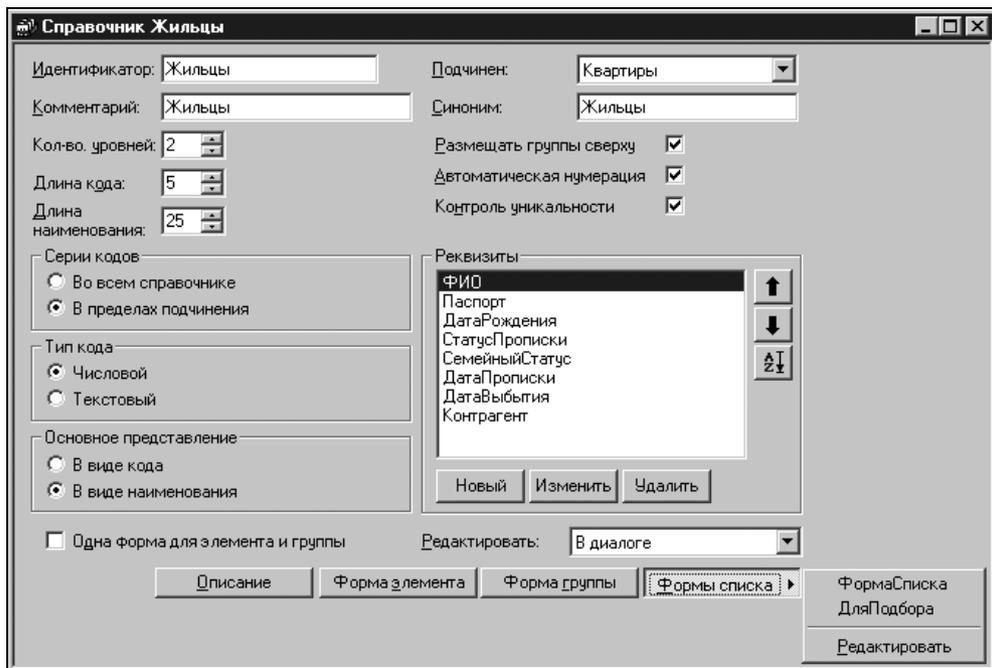


Рис. 4.8. Диалоговое окно **Справочник Жильцы** с меню после задания формы для выбора

Документы

Документы представляют собой набор реквизитов в форме, удобной для ввода информации об операции. Нередко в форме документа предусмотрена печать документа в стандартной форме. Создадим новый документ для учета заявки на ремонт "Заявка" (рис. 4.9).

Как и для справочников, в документах присутствуют следующие реквизиты: идентификатор, синоним и комментарий.

Поле **Идентификатор** содержит краткое наименование вида документов. Идентификатор используется для обозначения вида документов в списках и диалогах, а также используется в модулях для обращения к реквизитам документа. Идентификатор представляет собой строку длиной не более 128 сим-

волов, состоящую из букв, цифр и знаков подчеркивания. Идентификатор не должен начинаться с цифры и содержать пробелы. В идентификаторе допускается смешивать прописные и строчные буквы.

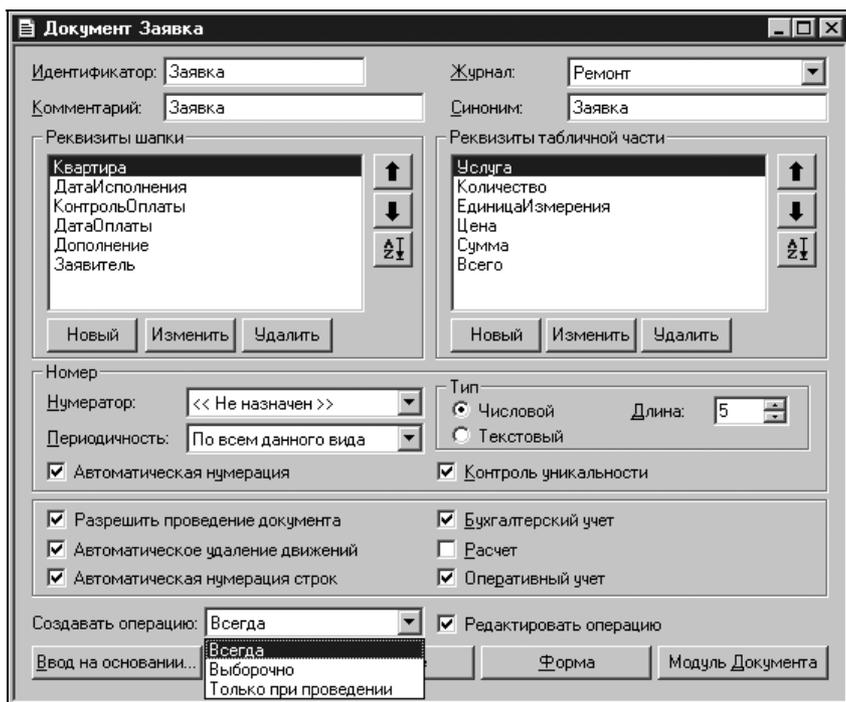


Рис. 4.9. Диалоговое окно **Документ Заявка**

Поле **Комментарий** содержит полное наименование документа, служит для расшифровки идентификатора. При работе с системой 1С:Предприятие комментарий выдается в скобках вслед за идентификатором документа в различных диалогах и списках. Комментарий представляет собой произвольную строку символов и может содержать любой текст, раскрывающий смысл идентификатора. Каких-либо ограничений на длину комментария не накладывается, здесь следует руководствоваться принципом разумной достаточности.

Поле **Синоним** содержит произвольную последовательность символом. Если синоним указан, он будет выдаваться во всех диалогах и списках вместо идентификатора.

Поле **Журнал** содержит идентификатор журнала, в который будут помещаться документы данного вида при использовании конфигурации. Значение выбирается из списка существующих в конфигурации журналов. При созда-

нии нового вида документов по умолчанию предлагается журнал "Прочие". Подробнее об объектах метаданных типа "Журнал" изложено в *разд. "Журналы" данной главы*.

Помимо указанного журнала все документы помещаются в журнал с условным наименованием "Полный".

Реквизиты

Реквизиты любого документа можно условно объединить в две группы:

- *реквизиты шапки* — встречаются в документе только один раз, независимо от их места в бланке документа;
- *табличная часть* — группа реквизитов, повторяющаяся некоторое число раз.

Описание структуры документа заключается в определении набора реквизитов шапки и табличной части. Эта работа производится с помощью управляющих элементов групп "Реквизиты табличной части" и "Реквизиты шапки". С функциональной точки зрения элементы этих групп одинаковы.

Управляющий элемент **Новый** вызывает на экран диалог для задания свойств нового реквизита документа — краткого и полного наименования, типа значения, длины, других характеристик.

Управляющий элемент **Изменить** открывает заполненный диалог для изменения свойств реквизита документа.

Управляющий элемент **Удалить** удаляет реквизит из списка.

Управляющий элемент  сдвигает реквизит в списке на одну позицию вверх.

Управляющий элемент  сдвигает реквизит в списке на одну позицию вниз.

Управляющий элемент  сортирует список реквизитов по алфавиту.

Управляющий элемент **Описание** вызывает текстовый редактор для создания пользовательского описания документа.

Управляющий элемент **Форма** вызывает редактор форм для создания формы ввода данных в документ.

Правила нумерации документа

Поле **Нумератор** содержит идентификатор нумератора, назначенного для последовательности документов данного вида. Документу может быть назна-

чен нумератор из числа уже существующих в конфигурации. В этом случае прочие элементы группы **Номер** за исключением признака **Автоматическая нумерация** станут недоступны, т. е. правила нумерации документов данного вида будут полностью определяться назначенным нумератором.

Для назначения документу нумератора из числа существующих в конфигурации следует выбрать идентификатор нумератора в реквизите **Нумератор**.

Использование нумераторов позволяет организовать сквозную нумерацию документов разных видов. Для этого всем документам, для которых требуется иметь сквозную нумерацию, должен быть назначен одинаковый нумератор. Контроль уникальности и присвоение очередного номера будет выполняться с учетом всех документов, для которых назначен этот нумератор.

Поле **Периодичность** устанавливает две важные характеристики номера документа:

- пределы контроля уникальности номеров документов;
- период повторяемости номеров.

Если включен признак контроля уникальности номеров документов, реквизит **Периодичность** устанавливает, в каких пределах осуществлять этот контроль. Например, если установлена периодичность **В пределах дня**, то уникальность номеров документов будет контролироваться в пределах суток: на следующие сутки номера документов могут повторяться, но в пределах суток они будут уникальны.

При включенном признаке автоматической нумерации документов система 1С:Предприятие будет присваивать очередной порядковый номер каждому новому документу. После завершения периода, установленного в реквизите **Периодичность**, нумерация документов начнется с 1.

Поле **Длина** устанавливает максимальную длину номера документа. Если документу назначен нумератор, этот реквизит диалога становится недоступным.

Переключатель **Тип** позволяет выбрать тип значения для номера документа — числовой или текстовый.

Выбор текстового типа номера бывает полезен, когда используется сложная система нумерации документов, и номер документа может включать помимо цифр буквы и символы-разделители. Если документу назначен нумератор, этот реквизит диалога становится недоступным.

Автоматическая нумерация устанавливает, будет ли система 1С:Предприятие автоматически присваивать очередной порядковый номер каждому новому документу. Автоматически присвоенный номер можно будет исправить.

Контроль уникальности устанавливает контроль уникальности номеров документов данного вида. Если эта опция включена, то при вводе нового документа его номер проверяется на уникальность в пределах, установленных в реквизите **Периодичность**. Если документу назначен нумератор, этот реквизит диалога становится недоступным.

Разрешить проведение документа устанавливает признак присутствия операции у данного документа. Если флажок установлен, документы этого вида будут проводиться.

Автоматическое удаление движений отменяет при повторном вводе предыдущую операцию. Если флажок установлен, при отмене проведения документа будут также удаляться все записи, отражающие документ в различных механизмах учета, которые документ записал в процессе проведения.

Автоматическая нумерация строк устанавливает режим нумерации строк табличной части документа. Если этот режим включен, то при редактировании документа в форме и при работе с ним средствами встроенного языка существует возможность изменения порядка расположения строк, однако, при любых перестановках строк их номера всегда будут отражать последовательность расположения строк без пропусков и нарушения порядка номеров. Если режим отключен, то номера строк могут редактироваться при вводе документа в форме и также средствами встроенного языка. При редактировании номеров строк система будет автоматически располагать строки в порядке возрастания номеров, однако последовательность номеров строк может иметь пропуски.

Проведение документа

Процедура проведения документа предназначена для создания и хранения данных документа в специальном образом организованных структурах — *проводах, регистрах или расчетах*.

Включение опции **Бухгалтерский учет** означает возможность документа при проведении формировать операции и проводки. Данный реквизит отсутствует, если не установлена компонента "Бухгалтерский учет".

Включение опции **Оперативный учет** означает возможность документа при проведении изменять регистры. Данный реквизит отсутствует, если не установлена компонента "Оперативный учет".

Включение опции **Расчет** означает возможность документа при проведении работать с журналами расчетов. Данный реквизит отсутствует, если не установлена компонента "Расчет".

Управляющий элемент **Создавать Операцию по документу** становится доступным, если для документа установлен флажок **Бухгалтерский учет**. Он определяет режим записи операции для документов этого вида:

- ❑ **Всегда** — все документы данного вида будут иметь операции, т. е. операция будет записываться в момент записи документа, и документ будет всегда присутствовать в журнале операций;
- ❑ **Выборочно** — конкретные документы данного вида могут при записи или при проведении создавать операцию. Наличие операции для конкретного документа может быть задано вызовом метода СуществуетОперация;
- ❑ **Только при проведении** — для документов данного вида операция может быть записана средствами встроенного языка только в процедуре ОбработкаПроведения. То есть только проведенные документы этого вида могут иметь операцию.

Редактировать операцию становится доступно, когда в свойствах объекта метаданных **Операция** выбран режим **Редактировать операции документов | Выборочно** (рис. 4.10). Установка или снятие этого флажка, соответственно, разрешает или запрещает пользователю редактировать ручную операцию, сформированные документами этого вида.

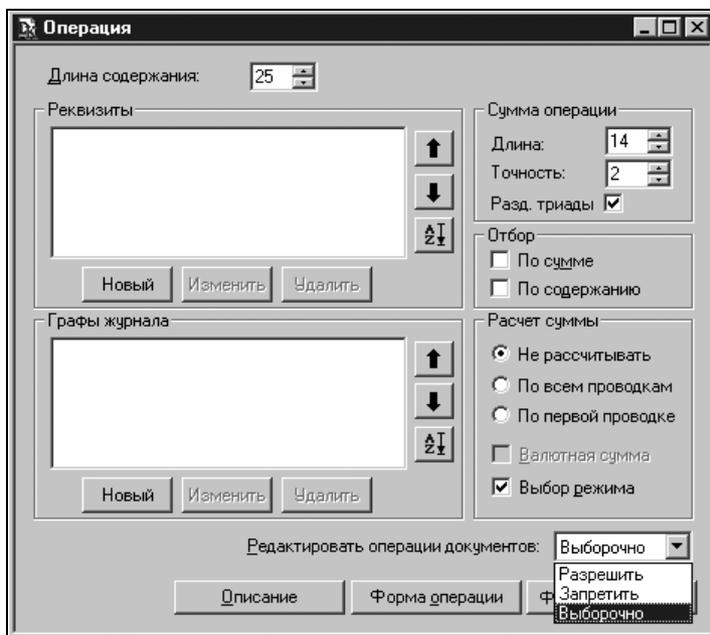


Рис. 4.10. Диалоговое окно операции

В документе **Заявка** нажатием кнопки **Модуль документа** (см. рис. 4.9) откроем модуль проведения, в котором в специальную заготовку добавим текст процедуры `ОбработкаПроведения()` для формирования соответствующих записей в регистр **Заявки**:

Процедура ОбработкаПроведения ()

Регистр.Заявки.Квартира = Квартира;

Регистр.Заявки.ДатаИсполнения = ДатаИсполнения;

Регистр.Заявки.Заявка = ТекущийДокумент ();

Регистр.Заявки.Долг = 0;

ВыбратьСтроки ();

Пока ПолучитьСтроку () = 1 **Цикл**

Если ПустоеЗначение (Услуга) = 1 **Тогда**

Сообщить ("Строка:" + СокрЛП (НомерСтроки) + ", не указан товар.
Строка НЕ обработана!");

Иначе

// Движение количества по регистру заявок

Регистр.Заявки.Услуга = Услуга;

Регистр.Заявки.Количество = Количество;

Регистр.Заявки.ДвижениеПриходВыполнить ();

КонецЕсли;

КонецЦикла;

// Движение сумм по регистру заявок

Если КонтрольОплаты = 1 **Тогда**

Регистр.Заявки.ДатаИсполнения = ДатаОплаты;

Регистр.Заявки.Услуга = "";

Регистр.Заявки.Количество = 0;

Регистр.Заявки.Долг = Итог ("Всего");

Регистр.Заявки.ДвижениеПриходВыполнить ();

КонецЕсли;

КонецПроцедуры

Текст этой процедуры должен быть разработан специалистом, осуществляющим конфигурирование системы. В тексте процедуры следует предусмотреть выполнение тех или иных операций по формированию записей в специальные структуры в зависимости от вида документа, а также любые другие необходимые действия.

Ввод на основании

Механизм ввода нового документа на основании уже существующего документа ускоряет работу пользователя и, что самое существенное, исключает вероятность ошибок.

Создадим документ **ВыполнениеЗаявки**, который вводится после обслуживания принятой заявки (рис. 4.11).

Все реквизиты документа **ВыполнениеЗаявки** присутствуют в документе **Заявка**.

Рис. 4.11. Диалоговое окно **Документ ВыполнениеЗаявки**

Пользователю придется повторно ввести данные реквизиты, кроме даты исполнения заявки. Документ, введенный на основании уже существующего (документа-основания), будет заполнен данными из выбранного документа.

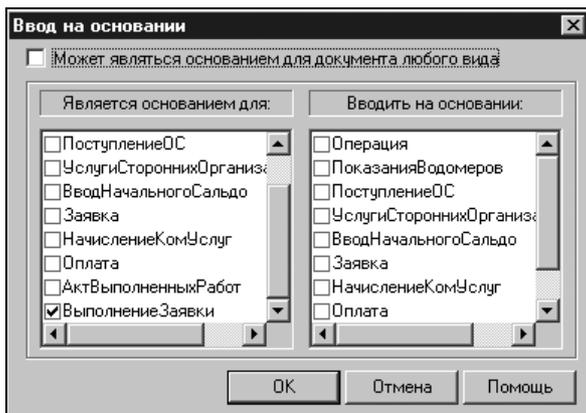
Для задания ввода документа **ВыполнениеЗаявки** на основании документа **Заявка** необходимо в документе **Заявка** или в документе **ВыполнениеЗаявки** нажать кнопку **Ввод на основании**. Нажатие кнопки **Ввод на основании** вызывает диалог для настройки режима ввода "на основании" (рис. 4.12).

Диалог позволяет создать две группы документов:

- документы, на основании которых будет вводиться редактируемый документ;
- документы, которые будут вводиться на основании редактируемого документа.

Выбор двух групп документов, выполненный в этом диалоге, позволяет ограничить список документов, выдаваемых пользователю при выполнении ввода документа "на основании".

В группе **Является основанием для** необходимо установить флажок для тех документов, которые будут вводиться на основании редактируемого документа.

Рис. 4.12. Диалоговое окно **Ввод на основании**

В группе **Вводить на основании** необходимо установить флажок для тех документов, на основании которых будет вводиться редактируемый документ. После отметки пункта **ВыполнениеЗаявки** в списке **Является основанием для документа Заявка** в списке **Вводить на основании документа ВыполнениеЗаявки** автоматически отметится пункт **Заявка**.

Включение опции **Может являться основанием для любого документа** равносильно тому, что в списке **Является основанием для** будут выбраны все документы.

Для организации ввода документа "на основании" необходимо в модуле формы документа создать процедуру с зарезервированным именем `ВводНаОсновании()`. Эта процедура выполняется после выбора команды **Ввести на основании** из меню **Действия** главного меню системы 1С:Предприятие и последующего выбора из списка нужного вида документов либо после выбора пункта пользовательского меню, выполняющего команду **Ввод на основании**. При вызове процедуры в качестве параметра ей передается документ-образец.

Текст этой процедуры должен быть разработан специалистом, осуществляющим конфигурирование системы. В тексте процедуры следует предусмотреть выполнение тех или иных операций по переносу информации — в зависимости от вида документа-образца, а также любые другие необходимые действия. В нашем случае в модуле формы документа **ВыполнениеЗаявки** должна присутствовать процедура:

Процедура `ВводНаОсновании (ДокументОснование)`

Если `ДокументОснование.Вид()="Заявка"` **тогда**

`Квартира = ДокументОснование.Квартира;`

`ДатаИсполнения = ДокументОснование.ДатаИсполнения;`

```

Заявитель = ДокументОснование.Заявитель ;
ДокументОснование.ВыбратьСтроки ( ) ;
Пока ДокументОснование.ПолучитьСтроку ( ) = 1 цикл
    НоваяСтрока ( ) ;
    Услуга = ДокументОснование.Услуга ;
    Количество = ДокументОснование.Количество ;
    ЕдиницаИзмерения = ДокументОснование.ЕдиницаИзмерения ;
    Цена = ДокументОснование.Цена ;
    Сумма = ДокументОснование.Сумма ;
    Всего = ДокументОснование.Всего ;
КонецЦикла ;
КонецЕсли ;
КонецПроцедуры

```

Внимание!

Если при работе пользователей с системой 1С:Предприятие предполагается частое применение режима ввода "на основании" (при большом документообороте), можно ввести в пользовательский интерфейс пункты меню или кнопки панелей инструментов, выполняющие команду **Ввод на основании** для конкретных документов. Использование таких пунктов меню (кнопки) позволит при вводе документа "на основании" исключить этап выбора вида документа из выдаваемого на экран списка и, таким образом, ускорить работу.

Константы

Константы относятся к разряду постоянной информации, но некоторые значения все же могут быть изменены. Немаловажным свойством таких констант является ведение истории изменения значений. В нашей конфигурации введем необходимую для формирования списка должников константу **ПределСуммыДолга**. Для ввода новой константы необходимо установить указатель на пункт **Константы** и в меню **Действия** выбрать пункт **Новый элемент**. Можно также воспользоваться выбором в контекстном меню пункта **Новая Константа** (рис. 4.13).

Диалог свойств организован в виде картотеки. Все свойства, которые вы можете определить, объединены в несколько групп, расположенных на вкладках. Для доступа к свойствам конкретной группы щелкните мышью на соответствующей вкладке в окне диалога свойств. В зависимости от типа описываемых данных состав свойств той или иной вкладки может меняться, кроме того, отдельные свойства во вкладках могут закрываться от использования. Недоступные для редактирования свойства выводятся серым цветом (рис. 4.14).

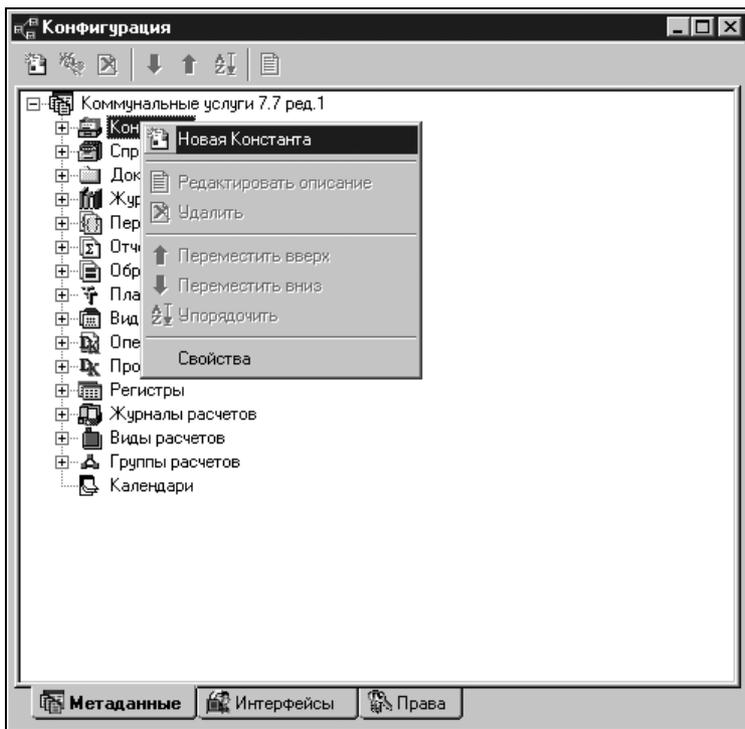
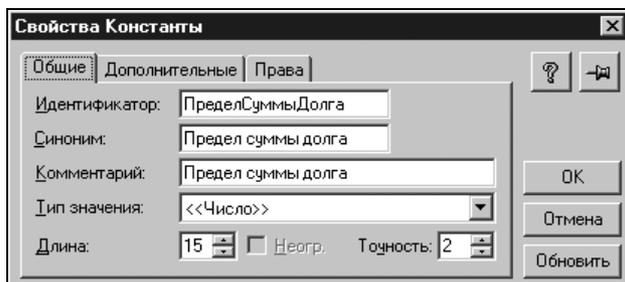


Рис. 4.13. Контекстное меню констант

Рис. 4.14. Диалоговое окно **Свойства Константы** на вкладке **Общие**

Вкладка **Общие** содержит:

- **Идентификатор** — имя объекта. Представляет собой строку длиной не более 128 символов, состоящую из букв, цифр и знаков подчеркивания. Не должен начинаться с цифры и/или содержать пробелы. Идентификатор позволяет обращаться к объекту из программных модулей;
- **Комментарий** — краткое описание объекта;

- Тип значения** — тип значения данного объекта:
- **Число** — значением объекта может быть натуральное число (положительное или отрицательное). При вводе используются цифры от 0 до 9 и знак минус (-);
 - **Строка** — значением объекта может быть произвольная последовательность символов;
 - **Дата** — значением объекта может быть дата в формате ДД.ММ.ГГ;
 - **Справочник** — в качестве значения объекта используется элемент данных из какого-либо определенного в системе Справочника. Кроме указания в качестве типа значения объекта **Справочника вообще**, можно также указать наименование конкретного вида Справочника;
 - **Документ** — в качестве значения объекта используется ссылка на Документ. Так же, как и для типа значения Справочник, в качестве типа значения объекта можно выбрать наименование конкретного вида Документа;
 - **Перечисление** — в качестве значения объекта используется элемент Перечисления. При указании этого типа значения из выдаваемого на экран списка необходимо выбрать наименование конкретного Перечисления;
- Длина** — максимальное количество позиций, которое отводится для ввода значения объекта. Длина указывается только для значений типа Строка и Число. Для значения типа Дата его длина всегда равна 8 (ДД.ММ.ГГ), а для значений типа Справочник и Документ указания длины значения не требуется;
- Точность** — максимально возможное количество знаков после десятичной точки для числового реквизита.

Вкладка **Дополнительные** содержит:

- Разделять триады** — признак разделения триад при отображении значения объекта;
- Неотрицательный** — признак неотрицательного значения объекта;
- Периодический** — признак периодического объекта (каждое новое значение объекта привязывается к дате его установки).

Очевидно, что в условиях постоянного роста цен данная константа должна ежегодно увеличиваться. Зададим признак ведения истории изменения константы признаком **Периодический** (рис. 4.15).

По умолчанию доступ к диалоговому окну констант в пользовательском интерфейсе осуществляется выбором пункта **Константы** меню **Операция** (рис. 4.16).

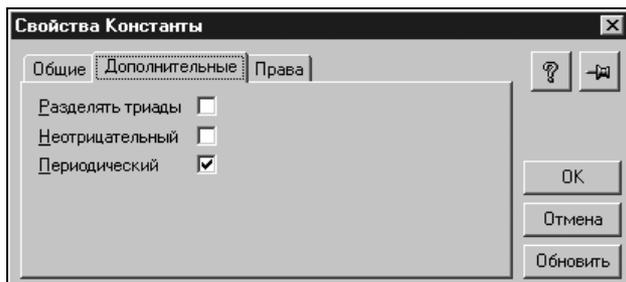


Рис. 4.15. Диалоговое окно Свойства Константы на вкладке Дополнительные

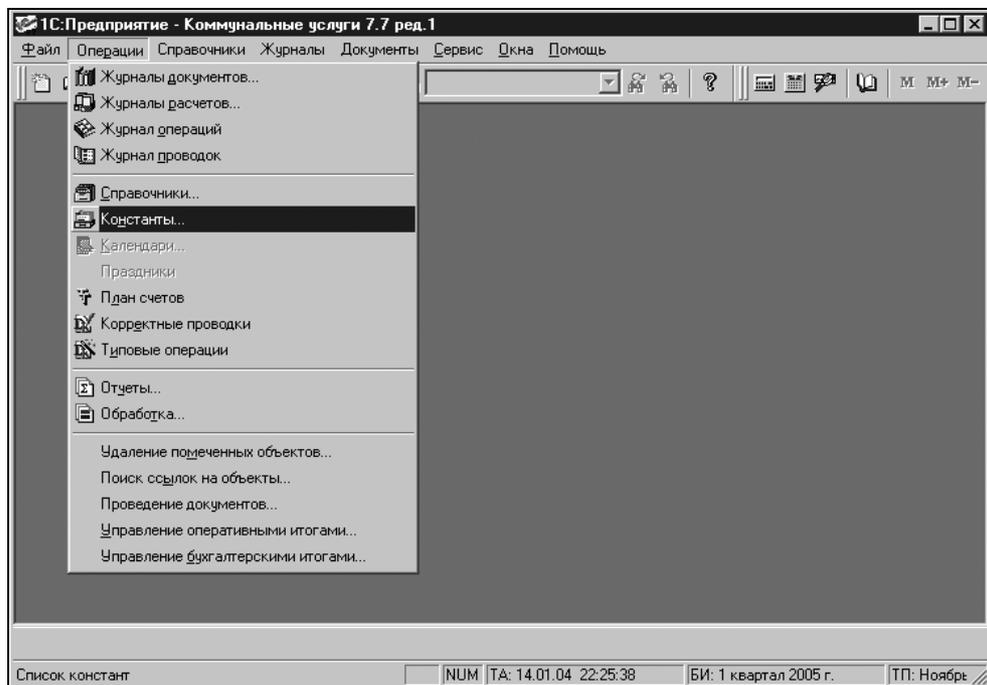


Рис. 4.16. Меню Операции в пользовательском интерфейсе

Следует обратить особое внимание на то, что в диалоговом окне **Список констант** в колонке **Код** отображается поле **Синоним**, а в колонке **Наименование** — поле **Комментарий** (рис. 4.17).

Внимание!

Изменить форму и содержание диалогового окна констант невозможно!

The screenshot shows a window titled 'Список констант (26.01.05)'. Inside the window is a table with three columns: 'Код', 'Наименование', and 'Значение'. The table contains several rows of data, with the first row highlighted.

Код	Наименование	Значение
Предел суммы долга	Предел суммы долга	100.00
ИНН	ИНН	2309072928
Основной вид номенклатуры	Основной вид номенклатуры	Коммунальные услуги
Наименование организации	Наименование организации	г.Краснодар МУ "Жилзаказчик"
Основной банковский счет	Основной банковский счет	Населения

Рис. 4.17. Диалоговое окно **Список констант** в пользовательском интерфейсе

Журналы

Журналы предназначены для систематизации и хранения документов. Пользователь должен иметь возможность быстро найти нужный документ или отобрать группу документов по ряду параметров. В нашей конфигурации введем необходимый для систематизации документов начисления журнал **Начисления**. Для ввода нового журнала необходимо установить указатель на пункт **Журналы** и в меню **Действия** выбрать пункт **Новый журнал**. Можно также воспользоваться выбором в контекстном меню пункта **Новый журнал** (рис. 4.18).

Новый журнал можно создать также с помощью конструктора журнала. Конструктор удобен тем, что прямо в нем можно выполнить ряд дополнительных действий после создания нового журнала. Особенно удобно в конструкторе использование расширенных подсказок (рис. 4.19). Конструктор вызывается аналогично заданию нового журнала, только выбирается пункт **Конструктор журнала документов**.

Это первый диалог **Конструктора Журнала документов**. В этом диалоге нужно обязательно указать идентификатор для создаваемого объекта. Можно также указать синоним идентификатора, его комментарий и изменить способ создания новых объектов метаданных:

- **Идентификатор** — краткое наименование объекта. **Идентификатор** используется для обозначения объекта метаданных в списках и диалогах, а также для обращения к нему в программных модулях. **Идентификатор** — строка длиной не более 128 символов из букв, цифр и знаков подчеркивания, не должен начинаться с цифры и содержать пробелы;

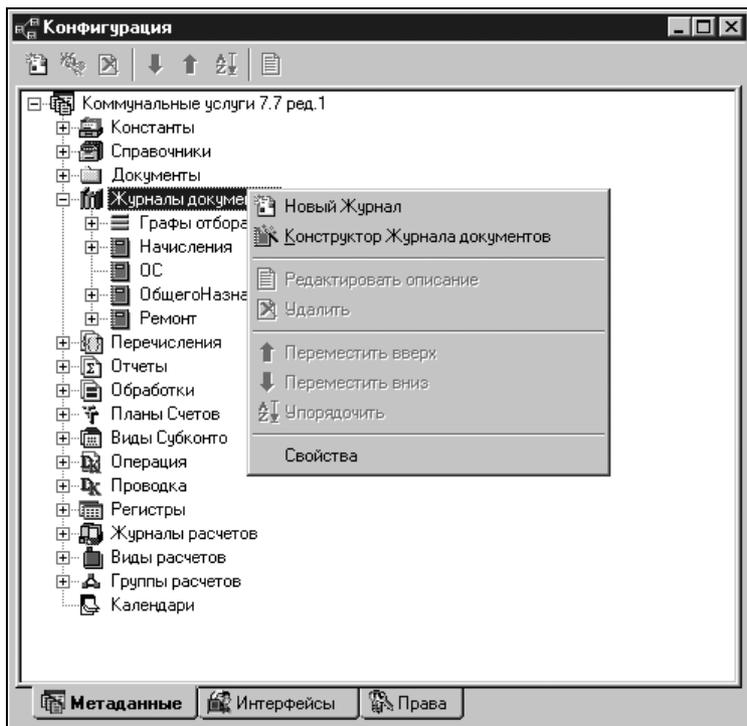


Рис. 4.18. Контекстное меню журналов

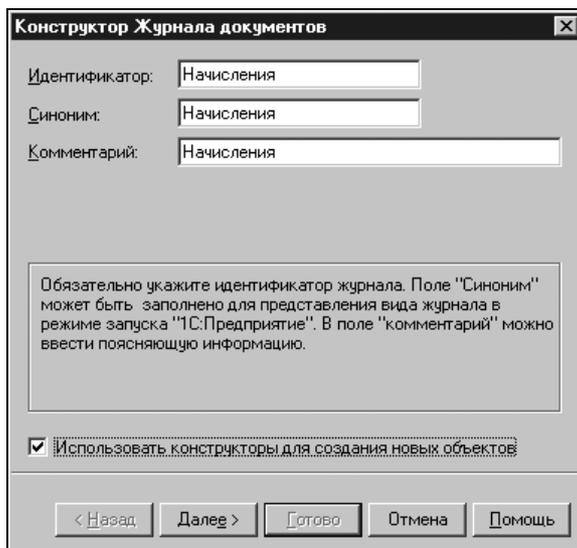


Рис. 4.19. Диалоговое окно Конструктор Журнала документов на этапе 1

- ❑ **Синоним** — если существует синоним, то при работе с системой 1С:Предприятие он будет выдаваться во всех диалогах и списках вместо идентификатора. Не имеет ограничений на использование символов;
- ❑ **Комментарий** — служит для расшифровки идентификатора. При работе с системой 1С:Предприятие комментарий выдается в скобках вслед за идентификатором (или синонимом) в различных диалогах и списках. **Комментарий** — произвольная строка символов. Может содержать любой текст, раскрывающий смысл идентификатора. Ограничений на длину строки не накладывается.

Если флажок **Использовать конструкторы для создания новых объектов** установлен, при создании нового объекта метаданных типа Справочник, Документ, Журнал документов, Отчет, Вид Субконто или Журнал расчетов автоматически запускается соответствующий **Конструктор**. Если флажок не установлен, при создании нового объекта метаданных будет открыто окно редактирования нового объекта метаданных. Этот флажок одновременно снимается и устанавливается в первом диалоге всех перечисленных **Конструкторов**. Установка или снятие флажка запоминается системой вне зависимости от способа выхода из окна **Конструктора** (например, по кнопке **Отмена** или клавише <Esc>).

Для перехода ко второму этапу необходимо нажать кнопку **Далее** (рис. 4.20).

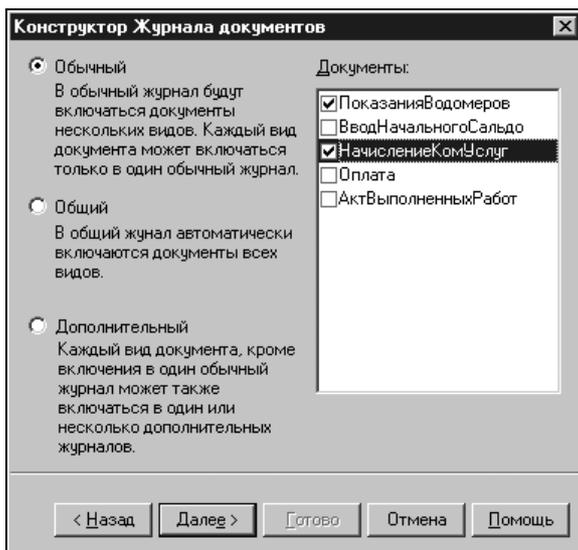


Рис. 4.20. Диалоговое окно Конструктор Журнала документов на этапе 2

В этом диалоге можно выбрать тип журнала:

- Обычный** — при этом значении переключателя в списке справа будут выданы все документы, не отнесенные пока ни к какому обычному журналу (их можно найти в журнале **Прочие**). Нужно установить флажки тем документам, которые требуется включить в создаваемый обычный журнал;
- Общий** — при этом значении переключателя в списке справа будут для справки выданы все документы конфигурации. Флажки в списке установлены и недоступны: документы из создаваемого общего журнала исключать нельзя;
- Дополнительный** — при этом значении переключателя в списке справа будут выданы все документы конфигурации.

В этом диалоге можно также определить документы, входящие в него. Флажки в списке по умолчанию не установлены, но доступны: нужно установить флажки тем документам, которые требуется включить в создаваемый дополнительный журнал.

После задания нужных опций можно нажать кнопку **Далее** для перехода к третьему этапу (рис. 4.21).

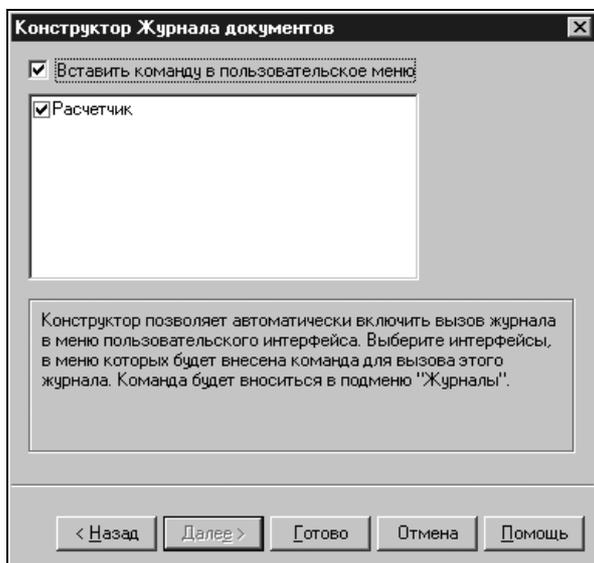


Рис. 4.21. Диалоговое окно **Конструктор Журнала документов** на этапе 3

В диалоге **Конструктор Журнала документов** можно вставить команду вызова создаваемого журнала в пользовательское меню.

Если флажок **Вставить команду в пользовательское меню** установлен, команда вызова журнала будет вставлена в указанные ниже пользовательские меню (в подменю **Журналы**).

Список пользовательских меню доступен, если установлен флажок над ним. Флажками в списке нужно отметить пользовательские меню, в которые необходимо вставить команду вызова журнала.

По умолчанию в состав журнала входят графы:

- Дата** — дата создания документа, как она указана в документе;
- Время** — время создания документа;
- Документ** — вид документа;
- Номер** — номер документа.

Для задания списка и свойств дополнительных граф журнала служит окно редактирования (рис. 4.22).

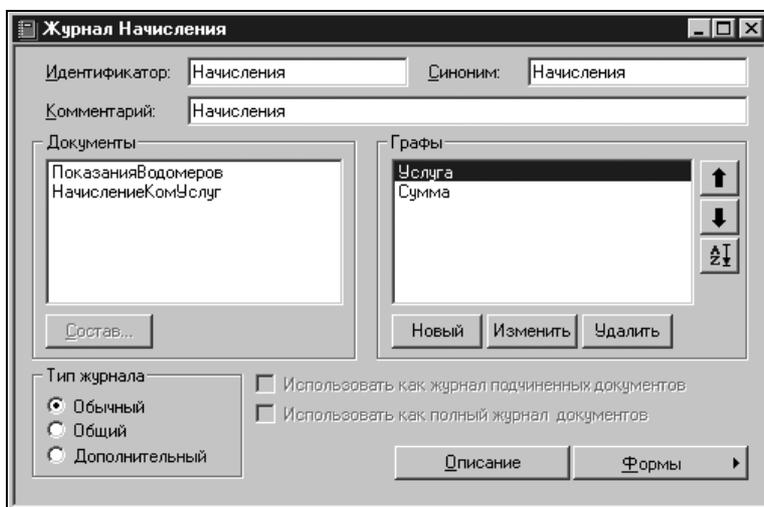


Рис. 4.22. Диалоговое окно Журнал Начисления

Для вызова окна редактирования созданного журнала **Начисления** дважды щелкните левой кнопкой мыши на идентификаторе журнала в дереве метаданных.

В диалоговом окне журнала уже присутствуют заданные в конструкторе реквизиты **Идентификатор**, **Синоним**, **Комментарий**, **Тип журнала** и группа **Документы**.

Управляющие элементы группы выполняют различные функции в зависимости от типа редактируемого журнала.

Управляющие элементы группы **Тип журнала** позволяют назначить тип журнала:

- **Обычный** — основное средство для работы с документами. При создании документа в Конфигураторе можно указать, в каком журнале будет осуществляться работа с документами этого вида при использовании конфигурации. Один обычный журнал может быть назначен одновременно нескольким видам документов, но документы одного вида всегда будут доступны только в одном обычном журнале.
- **Общий** — позволяет работать с документами любых видов, существующих в конфигурации, а также выполнять отбор документов по значениям их реквизитов.
- **Дополнительный** — позволяет работать с документами любых видов, существующих в конфигурации. В отличие от обычных журналов, документы одного вида могут быть доступны одновременно в нескольких дополнительных журналах.

Внимание!

Не следует злоупотреблять созданием дополнительных журналов. Для поддержки таких журналов система 1С:Предприятие тратит определенные системные ресурсы, поэтому большое число журналов способно снижать общую производительность системы.

Если редактируется обычный журнал документов, в группе **Документы** для информации выдается список документов, которые будут доступны при работе с этим журналом. Изменить этот список нельзя: включение в него документа определяется свойствами самого документа и задается в окне его редактирования.

Если редактируется общий журнал документов, в группе **Документы** для информации выдается список всех документов, существующих в конфигурации. Изменить этот список нельзя. Активизируются признаки **Использовать как журнал подчиненных документов** и **Использовать как полный журнал документов**. Один из общих журналов, существующих в конфигурации, может быть назначен в качестве журнала подчиненных документов. Если включен флажок **Использовать как журнал подчиненных документов**, данный журнал будет использоваться при работе пользователя со списком документов, подчиненных выбранному документу.

Один из общих журналов, существующих в конфигурации, может быть назначен для показа полного журнала. Если включен флажок **Использовать как полный журнал документов**, данный журнал будет определять состав граф и форму полного журнала.

Если редактируется дополнительный журнал документов, то управляющие элементы группы **Документы** используются для выбора видов документов, которые будут доступны в редактируемом журнале. Список документов дополнительного журнала задается в окне **Состав документов** с помощью нажатия кнопки **Состав**. В диалоговом окне предлагается для выбора список видов документов, которые будут доступны в редактируемом дополнительном журнале (рис. 4.23).

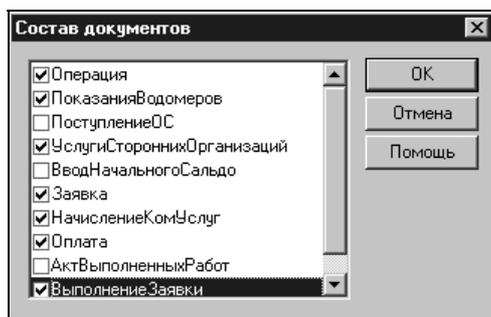


Рис. 4.23. Диалоговое окно журнала **Состав документов**

Элементы группы **Графы** служат для управления списком дополнительных граф журнала, а также для редактирования их свойств:

- Новый** — вызывает на экран окно редактирования свойств дополнительной графы журнала;
- Изменить** — открывает окно редактирования свойств дополнительной графы журнала для выбранной графы журнала;
- Удалить** — удаляет наименование графы из списка;
-  — сдвигает наименование графы в списке на одну позицию вверх;
-  — сдвигает наименование графы в списке на одну позицию вниз;
-  — сортирует список граф по алфавиту;
- Описание** — вызывает текстовый редактор для создания пользовательского описания журнала;
- Формы** — вызывает редактор форм для создания экранной и печатной формы журнала.

Создадим новую графу журнала **Начисления** с помощью кнопки **Новый**. Для задания реквизитов документов журнала, значения которых будут отображаться в графе, предназначен диалог **Графа журнала** (рис. 4.24).

Диалог вызывается на экран нажатием кнопки **Новый** или **Изменить** из группы **Графы** диалога **Журнал**.

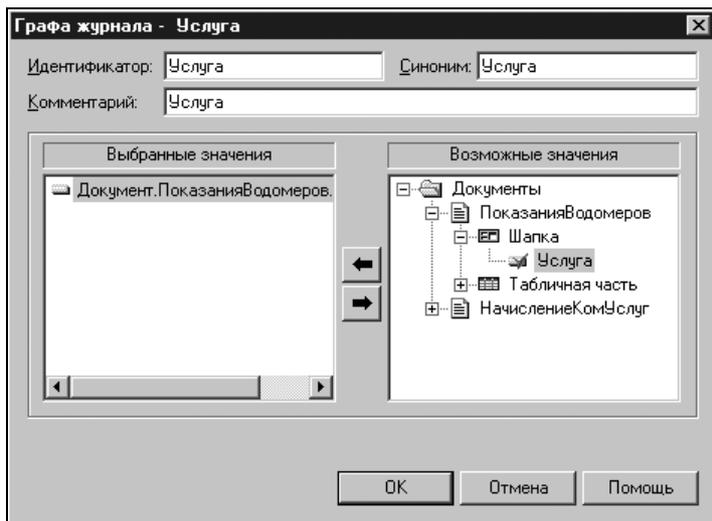


Рис. 4.24. Диалоговое окно журнала Графа журнала - Услуга

Идентификатор задает имя дополнительной графы журнала. **Идентификатор** представляет собой строку длиной не более 16 символов, состоящую из букв, цифр и знаков подчеркивания. Имя не должно начинаться с цифры и содержать пробелы. **Идентификатор** позволяет обращаться к значению графы из модулей.

Комментарий задает полное наименование дополнительной графы журнала, служит для расшифровки идентификатора.

Группа **Выбранные значения** содержит список тех реквизитов документов, которые будут отображаться в данной графе журнала.

Группа **Возможные значения** содержит список всех реквизитов тех документов, которые будут храниться в данном журнале. Список выдается в виде дерева, на первом уровне которого находятся наименования документов. Знак (+) в узле ветви дерева означает, что данная ветвь имеет подуровни, и щелчком мыши на (+) можно раскрыть следующий уровень. На самом нижнем уровне дерева находятся наименования реквизитов документов, которые можно выбрать для показа в дополнительной графе журнала, когда журнал выдается на экран для просмотра и редактирования.

Нажатием кнопки  выбранный в списке возможных значений реквизит документа добавляется в список **Выбранные значения**.

Нажатие кнопки  убирает реквизит документа из списка **Выбранные значения**.

В дополнительной графе журнала может отображаться информация из разных видов документов. Определение состава дополнительной графы журнала заключается в создании списка реквизитов, которые будут отображаться в данной графе журнала при его просмотре или редактировании. Когда диалог вызывается нажатием кнопки **Новый**, список **Выбранные значения** пуст. Чтобы выбрать реквизит для показа в журнале, можно воспользоваться одним из следующих способов:

- установить указатель мыши на наименовании нужного реквизита в списке **Возможные значения** и дважды щелкнуть левой кнопкой мыши;
- выбрать наименование реквизита в списке **Возможные значения** и нажать кнопку .

Полное имя реквизита будет помещено в список **Выбранные значения**, а сам реквизит будет помечен знаком .

Внимание!

Для показа в одной графе журнала можно выбрать только один реквизит документа. Если один из реквизитов документа уже был ранее выбран для отображения в данной графе, то выбор другого реквизита того же документа приведет к тому, что имя ранее выбранного реквизита будет удалено из списка **Выбранные значения**, имя нового выбранного реквизита будет помещено в список, и возле него появится отметка.

Чтобы удалить имя реквизита из списка **Выбранные значения**, можно воспользоваться одним из следующих способов:

- установить указатель мыши на наименовании реквизита в списке **Выбранные значения** и дважды щелкнуть левой кнопкой мыши;
- выбрать наименование реквизита в списке **Выбранные значения** и нажать кнопку .

Внимание!

При задании реквизита табличной части документа следует установить флажок **Итог по колонке** в диалоговом окне свойств соответствующего реквизита на вкладке **Дополнительные**, чтобы суммы отображались в графе (рис. 4.25).

При нажатии кнопки **Формы** появляется меню со списком всех существующих форм журнала и пунктом **Редактировать**. При создании нового журнала всегда предлагается основная форма журнала с названием по умолчанию **ФормаСписка** (рис. 4.26).

Часто для выбора документа из журнала необходима либо более сокращенная форма, чем основная, либо расширенная с дополнительными графами. Для задания дополнительных форм журнала или редактирования свойств существ-

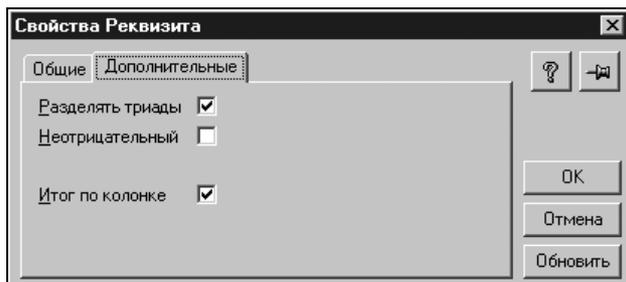


Рис. 4.25. Диалоговое окно Свойства Реквизита на вкладке Дополнительные



Рис. 4.26. Диалоговое окно журнала с меню кнопки Формы

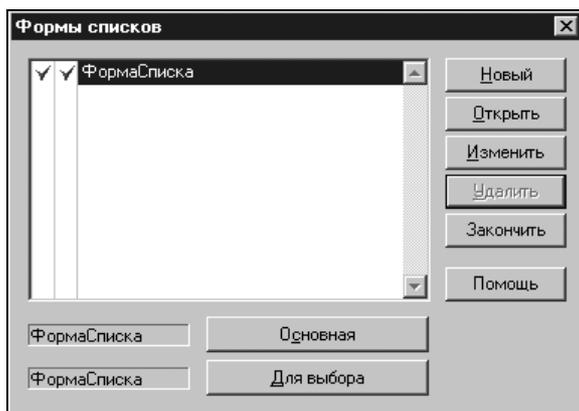


Рис. 4.27. Диалоговое окно Формы списков

вующих предназначен диалог **Формы списков**, который вызывается выбором в меню кнопки **Формы** пункта **Редактировать** (рис. 4.27).

Диалог **Формы списков** содержит список форм для показа журнала, а также кнопки для управления списком, вызова редактора форм и выбора момента использования конкретной формы журнала.

При нажатии кнопки **Новый** на экран будет выдан запрос, в который требуется ввести идентификатор и комментарий создаваемой формы списка (рис. 4.28).

Нажатие кнопки **Изменить** вызывает на экран запрос для редактирования идентификатора и комментария формы списка (рис. 4.28).

Нажатие кнопки **Удалить** удаляет форму из списка.

Нажатие кнопки **Закончить** закрывает диалог.

Нажатие кнопки **Основная** назначает выбранную форму списка для использования журнала для просмотра или редактирования его значения. Название формы выдается в рамке слева от кнопки.

Нажатие кнопки **Для выбора** назначает выбранную форму списка для использования журнала для выбора из него требуемого документа. Название формы выдается в рамке слева от кнопки.

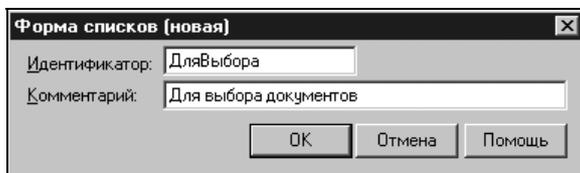


Рис. 4.28. Диалоговое окно задания наименования формы

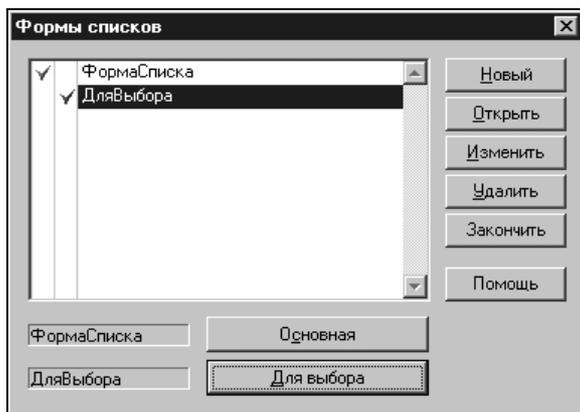


Рис. 4.29. Диалоговое окно **Формы списков** после задания формы для выбора

На рис. 4.31 в форме для выбора журнала **Начисления** отображена дополнительная графа **Сумма**.

Перечисления

Перечисления относятся к разряду постоянной информации, задаваемой только в Конфигураторе и не редактируемой пользователем. Перечисление представляет собой список значений какого-либо объекта. В нашей конфигурации введем необходимое для отслеживания сроков капитального ремонта перечисление **СостоянияДомов**. Для ввода нового перечисления необходимо установить указатель на пункт **Перечисления** и в меню **Действия** выбрать пункт **Новый элемент**. Можно также воспользоваться выбором в контекстном меню пункта **Новое Перечисление** (рис. 4.32).

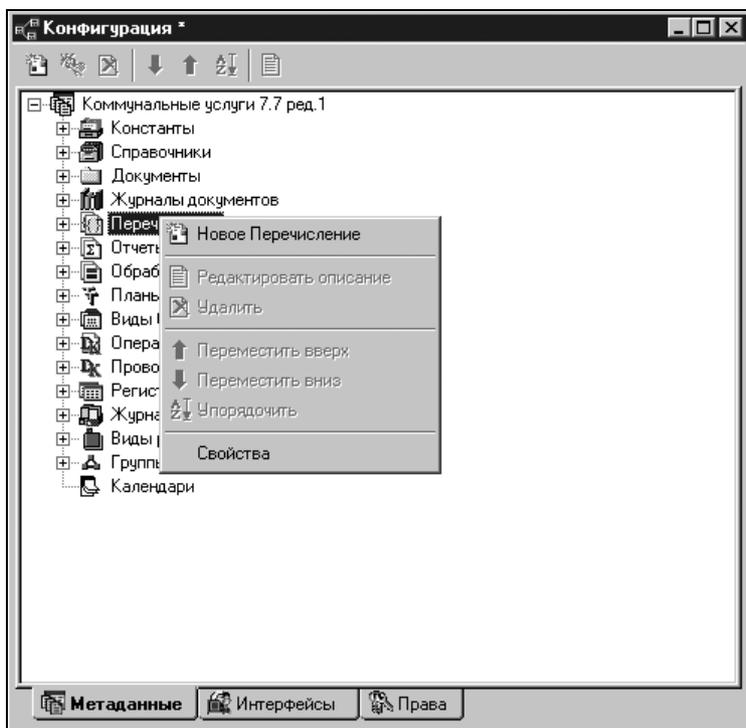


Рис. 4.32. Контекстное меню перечислений

Диалог **Перечисление** служит для создания перечисления и формирования списка его значений (рис. 4.33).

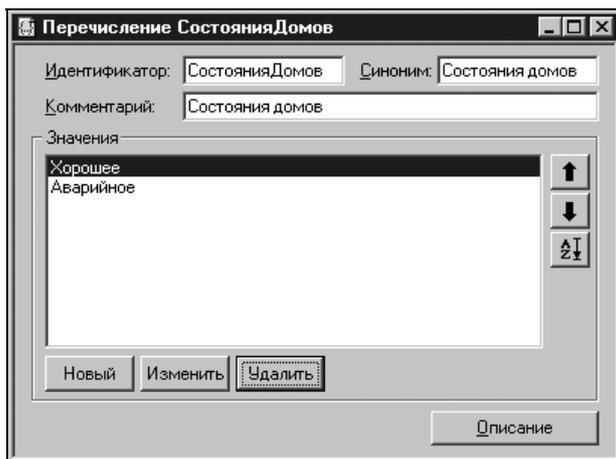


Рис. 4.33. Диалоговое окно перечисления

Диалог свойств содержит обычные для объектов метаданных реквизиты:

- **Идентификатор** — краткое наименование перечисления;
- **Комментарий** — полное наименование перечисления, служит для расшифровки идентификатора.

Управляющие элементы группы **Значения** служат для управления списком возможных значений перечисления и определения их свойств:

- **Новый** — вызывает на экран диалог для задания свойств нового значения перечисления;
- **Изменить** — открывает заполненный диалог для изменения свойств значения перечисления;
- **Удалить** — удаляет значение из списка.

Кнопки управления порядком следования значений:

- **Сдвинуть вверх**  — сдвигает значение в списке на одну позицию вверх;
- **Сдвинуть вниз**  — сдвигает значение в списке на одну позицию вниз;
- **Отсортировать список**  — сортирует список значений по алфавиту;
- **Описание** — вызывает текстовый редактор для создания пользовательского описания перечисления.

В таком же порядке список значений будет выдаваться пользователю при вводе реквизита типа Перечисление.

Добавим новое значение в список с помощью кнопки **Новый**. В диалоговом окне **Свойства Значения** введем описание нового значения (рис. 4.34).

Сдвинем введенное значение на одну позицию вверх (рис. 4.35).

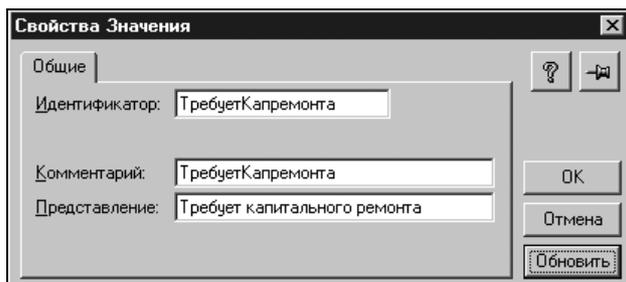


Рис. 4.34. Диалоговое окно свойства значения

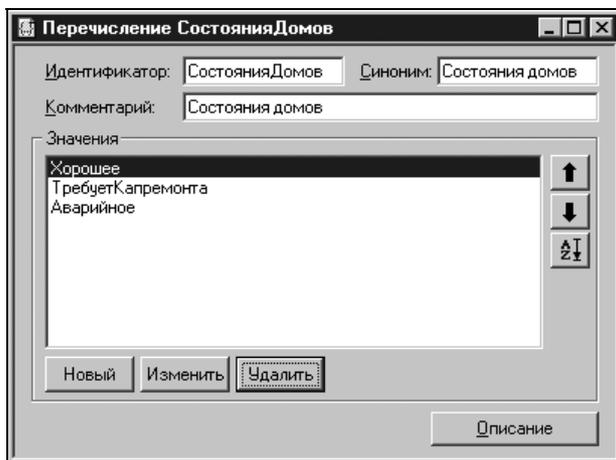


Рис. 4.35. Диалоговое окно перечисления после добавления нового значения

Отчеты

Отчеты являются, пожалуй, самыми востребованными объектами метаданных. Они используются для документирования процесса учета, а также для его аналитического учета. Отчеты условно делятся на несколько видов.

- Встроенные в документы отчеты.
- Оперативные отчеты.
- Внешние отчеты.
- Регламентированные отчеты.

Конструирование отчетов состоит из двух этапов: создание шаблона печатной формы и создание процедуры формирования отчета. Шаблон печатной формы формируется с помощью табличного редактора. Процедура формирования отчета назначается элементу диалога типа Кнопка. В процедуре оперативных, внешних и регламентированных отчетов формируются выборки информации из документов и справочников. В процедуре встроенных в документ отчетов используется информация текущего документа.

Процесс создания шаблона заключается в следующем. С помощью табличного редактора рисуется шаблон отчета. Сам шаблон представляет собой совокупность прямоугольных областей, каждая из которых служит для выдачи определенной части готового отчета: область для выдачи заголовочной части (наименования, даты и т. п.); область для выдачи шапки табличной части и т. д. Любой такой области отчета можно присвоить уникальное имя, и в дальнейшем на него ссылаться. Необходимо иметь в виду, что шаблон отчета не связан напрямую с готовым отчетом. Скорее, таблица шаблона представляет собой набор заготовок, из которых в процессе обработки алгоритма складывается готовый отчет.

Что такое таблица в системе 1С:Предприятие? *Таблицей* будем называть совокупность ячеек прямоугольной формы, организованных в строки и столбцы. Каждый столбец таблицы и каждая строка имеют свой уникальный номер. Строки и столбцы нумеруются независимо, нумерация начинается с 1 и ведется от левого верхнего угла таблицы. Таким образом, любая ячейка таблицы может быть обозначена парой чисел: номером строки и номером столбца, на пересечении которых она находится. Самая верхняя ячейка столбца, изображенная серым цветом, называется *заголовком столбца*. В ней выводится номер столбца, кроме того, она используется для выделения столбца. Аналогичное назначение имеет и самая левая ячейка строки, которая называется *заголовком строки*.

Название самой таблицы выводится на закладке, расположенной у нижнего обреза листа таблицы. Редактор формы, одной из составляющих которого и является редактор таблиц, может иметь несколько таблиц, содержащих различные шаблоны отчетов. Та таблица, с которой пользователь работает в настоящий момент, называется активной. Название активной таблицы на закладке всегда выводится жирным шрифтом. Создание шаблона выходной формы заключается в "рисовании" составных частей, заготовок, из которых затем будет "собрана" готовая выходная форма — отчет. Так как практически все деловые документы имеют "прямоугольную" структуру, удобнее создавать шаблоны таких документов в редакторе, способном манипулировать именно прямоугольными элементами.

Именно таким редактором и является табличный редактор, входящий в 1С:Предприятие. В процессе создания шаблона **Документа** или отчета вы

На вкладке **Таблица** представлено окно табличного редактора в виде электронной таблицы для создания шаблона отчета (рис. 4.37).

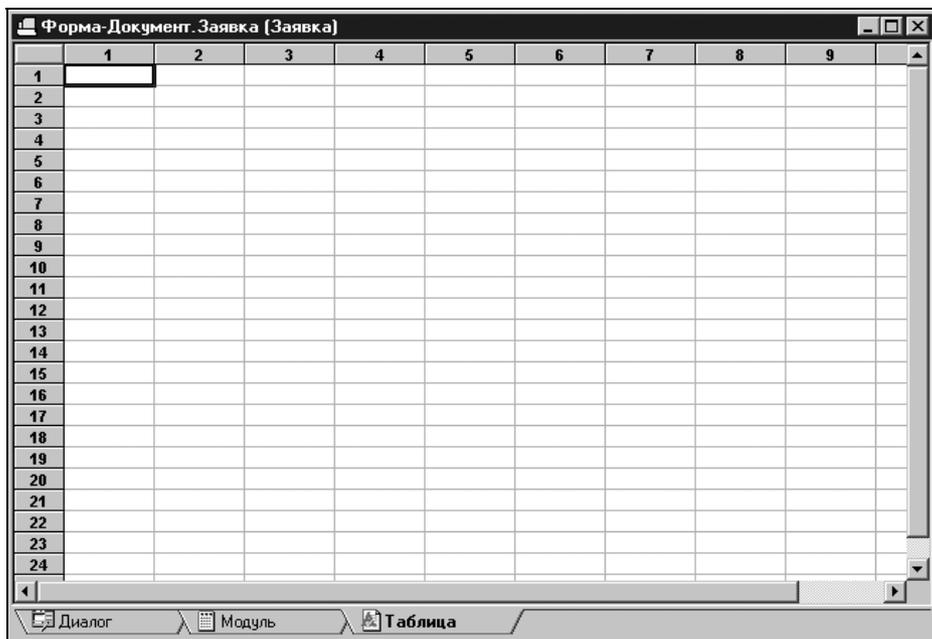


Рис. 4.37. Диалоговое окно формы документа Заявка на вкладке Таблица

Редактор форм может иметь несколько листов типа **Таблица**. Когда редактор формы вызывается в первый раз для вновь созданного элемента данных, он содержит только одну таблицу, но в процессе работы пользователь может добавлять и удалять таблицы.

Добавить новую таблицу можно выбором в контекстном меню. Для этого необходимо выбрать вкладку любого листа, нажатием правой кнопки мыши вызвать на экран контекстное меню и выбрать пункт **Добавить таблицу** (рис. 4.38).

Добавить таблицу также можно, выбрав пункт **Добавить таблицу** в меню **Окна** главного меню программы (рис. 4.39).

Все действия по редактированию содержимого таблицы производятся в активной таблице. Для того чтобы сделать таблицу активной, необходимо щелкнуть мышью по вкладке с названием нужной таблицы. Имя выбранной таблицы на вкладке будет выделено жирным шрифтом.

Вновь созданной таблице присваивается по умолчанию условное имя, состоящее из слова "Таблица" и порядкового номера этой таблицы. Естествен-

но, что это условное имя можно изменить. Сделать это можно одним из следующих способов:

- выбрать ярлык таблицы, которую необходимо переименовать, нажатием правой кнопки вызвать на экран контекстное меню и выбрать функцию **Задать имя таблицы** (см. рис. 4.38);
- в меню **Окна** главного меню программы выбрать функцию **Задать имя таблицы** (см. рис. 4.39).

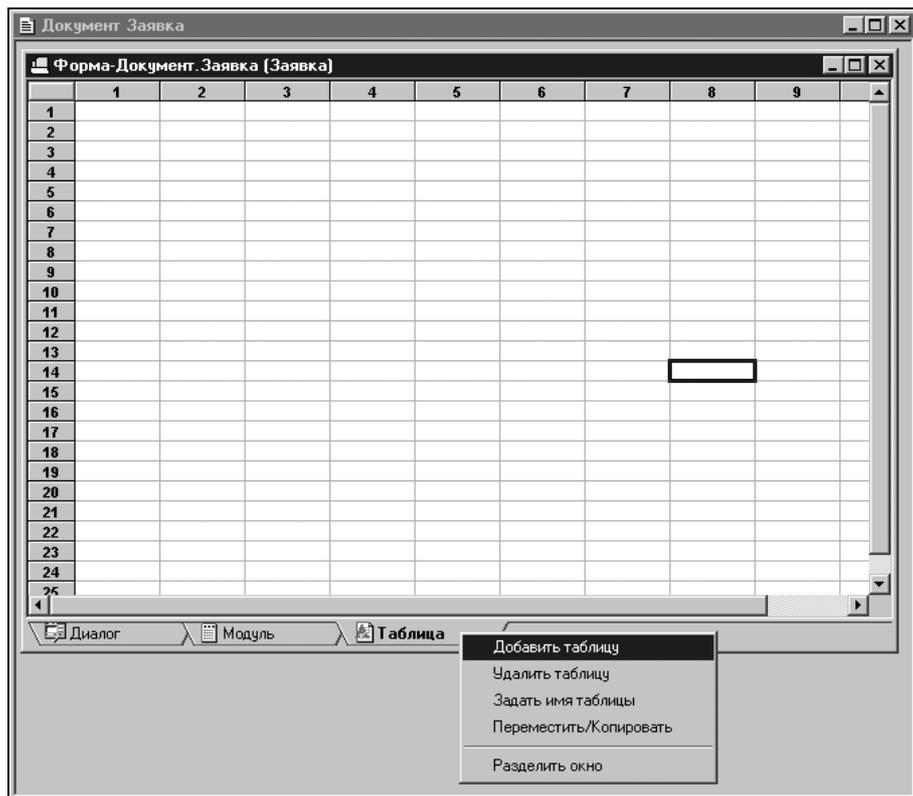


Рис. 4.38. Диалоговое окно формы документа Заявка на вкладке Таблица с меню таблицы

В выданном на экран запросе необходимо ввести новое имя таблицы. Имя должно иметь длину до 50 символов и может включать буквы, цифры, пробелы и знаки подчеркивания (рис. 4.40).

Удаление таблицы выполняется одним из следующих способов:

- нажатием правой кнопки мыши вызвать на экран контекстное меню и выбрать функцию **Удалить таблицу** (см. рис. 4.38);

□ в меню **Окна** главного меню программы выбрать функцию **Удалить таблицу** (см. рис. 4.39).

После ответа на запрос о необходимости удаления таблицы текущая таблица будет безвозвратно удалена.

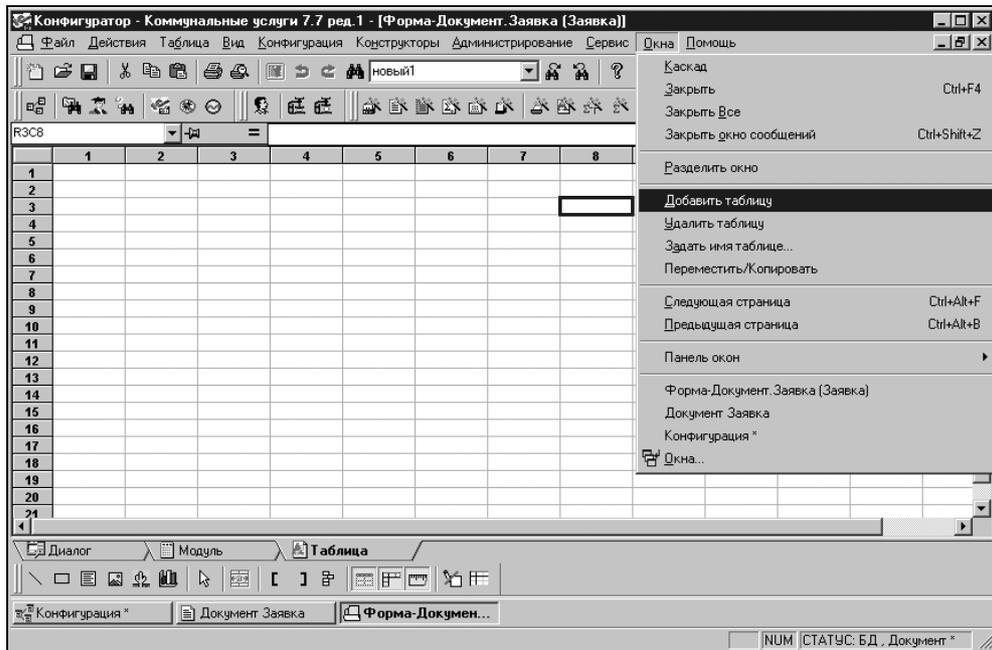


Рис. 4.39. Меню Окна главного меню программы

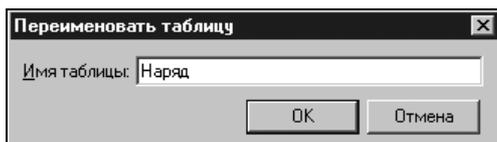


Рис. 4.40. Диалоговое окно Переименовать таблицу

Навигация в таблице

Согласно общей концепции интерфейса Windows, окно таблицы в любой момент времени показывает только часть таблицы, лежащей "под" этим окном. Для перехода к любой ячейке в видимой области таблицы достаточно щелкнуть на этой ячейке мышью. Выбранная ячейка становится активной. Для вывода на экран скрытых за границами окна областей таблицы используются линейки прокрутки. Основные действия с ячейками представлены табл. 4.1.

Таблица 4.1. Основные действия с ячейками

Операция	Действие
Переместиться на строку вверх или вниз	Щелкните мышью нижнюю или верхнюю кнопку со стрелкой на вертикальной линейке прокрутки
Переместиться на колонку вправо или влево	Щелкните мышью левую или правую кнопку со стрелкой на горизонтальной линейке прокрутки
Переместиться на страницу вверх или вниз	Щелкните мышью ниже или выше бегунка на вертикальной линейке прокрутки
Переместиться на страницу вправо или влево	Щелкните мышью левее или правее бегунка на горизонтальной линейке прокрутки
Переместиться на несколько страниц	Используя бегунки на линейках прокрутки, установите необходимое положение окна
Выделить ячейку	Щелкните на ячейке мышью или используйте клавиши управления курсором
Выделить диапазон ячеек	Нажмите левую кнопку мыши на первой ячейке диапазона; затем, не отпуская кнопку мыши, переместите указатель на последнюю ячейку диапазона и отпустите кнопку
Выделить несколько несмежных ячеек	Выделите первую ячейку или диапазон ячеек; затем нажмите на клавиатуре клавишу <Ctrl> и, не отпуская ее, выделите следующую ячейку или диапазон ячеек
Выделить строку	Щелкните мышью заголовок строки
Выделить столбец	Щелкните мышью заголовок столбца
Все ячейки в таблице	Щелкните кнопку в левом верхнем углу таблицы (это то место, где "пересекаются" заголовки строк и столбцов)
Снятия выделения	Щелкните мышью любую невыделенную ячейку
Изменить высоту строки	Установить указатель мыши на нижнюю границу заголовка строки, пока курсор мыши не примет форму  , перемещая мышью границу заголовка, установить нужную высоту строки
Изменить ширину столбца	Установить указатель мыши на правую границу заголовка столбца, пока курсор мыши не примет форму  , перемещая мышью границу заголовка, установить нужную ширину столбца таблицы

Все действия по редактированию данных, содержащихся в таблице, производятся в активной ячейке.

Текст может быть введен в любую ячейку таблицы. При формировании документа или отчета ячейки вместе с содержащимся в них текстом копируются из шаблона в готовый отчет без изменений. Для ввода текста в пустую ячейку просто начните набирать нужные символы на клавиатуре. Ячейка перейдет в режим редактирования — в ней появится текстовый курсор (мигающая вертикальная черта), и в ячейке будет отображаться вводимый вами текст. Для окончания ввода текста надо нажать клавишу <Enter>. Чтобы отменить сделанные вами изменения в тексте ячейки, необходимо нажать <Esc>.

Закончить редактирование текста можно также, щелкнув мышью любую невыделенную ячейку. Это действие равносильно нажатию клавиши <Enter>.

Если редактируемая ячейка уже содержала какой-либо текст, то ввод новых символов будет производиться в начало существующего текста. Если вы хотите продолжать ввод текста в ячейку не с начала, а с какой-либо другой позиции — перед вводом текста нажмите клавишу <Enter> или <F2>. Тогда ячейка перейдет в режим редактирования, и вы можете переместить курсор в нужную позицию текста, используя клавиши управления курсором.

Следует особо отметить комбинацию <Ctrl>+<Enter>. Нажатие этих клавиш позволяет начать новую строку, не прекращая редактирование ячейки.

Если текст полностью не помещается в ячейке, то для его показа будет использовано пространство соседних по горизонтали ячеек (слева, справа или с обеих сторон — это зависит от типа выравнивания текста в ячейке). В процессе редактирования текста в ячейке таблицы вы можете использовать те же клавиши и комбинации клавиш, которые используются в других приложениях Windows. В табл. 4.2 перечислены эти клавиши и приведено краткое описание совершаемых ими действий.

Таблица 4.2. Клавиши, используемые для редактирования

Клавиши	Действия
Клавиши со стрелками	Смещение курсора на 1 символ вправо, влево, вверх или вниз
<Ctrl>+<→>/<←>	Смещение курсора на слово вправо/влево
<Home>	Перемещение курсора в начало строки
<End>	Перемещение курсора в конец строки
	Удаление: символа справа от курсора или выделенного фрагмента текста
<Backspace>	Удаление: символа слева от курсора или выделенного фрагмента текста
<Ctrl>+	Удаление текста от курсора до конца строки

Любую ячейку таблицы можно переместить в другое место таблицы. Точнее, перемещается не сама ячейка, а содержащийся в ней текст и атрибуты форматирования ячейки. Также можно копировать ячейку, перенося таким образом атрибуты форматирования ячейки-образца в другие ячейки таблицы.

Для перемещения ячейки необходимо выполнить следующие действия:

1. С помощью клавиш или мыши выделите ячейку, которую предполагается перенести.
2. Укажите мышью на рамку, обрамляющую ячейку, так, чтобы указатель мыши принял форму стрелки.
3. Нажмите левую кнопку мыши.
4. Не отпуская левую кнопку мыши, перетащите ячейку на новое место.
5. После того как ячейка займет нужное положение, кнопку мыши можно отпустить.

Если необходимо переместить ячейку в ту часть таблицы, которая скрыта границами окна, — подведите указатель мыши к краю видимой части таблицы, и таблица начнет автоматически "прокручиваться".

Также с помощью мыши легко выполняется копирование ячейки:

1. С помощью клавиш или мыши выделите ячейку, которую предполагается копировать.
2. Укажите мышью на рамку, обрамляющую ячейку, так, чтобы указатель мыши принял форму стрелки.
3. Нажмите левую кнопку мыши и клавишу <Ctrl>.
4. Не отпуская левую кнопку мыши и клавишу <Ctrl>, перетащите ячейку на новое место.
5. После того как ячейка займет нужное положение, отпустите кнопку мыши и клавишу <Ctrl>.

Внимание!

Если в процессе переноса копии ячейки отпустить клавишу <Ctrl>, то ячейка будет перемещена, а не скопирована.

Все сказанное относительно перемещения и копирования отдельной ячейки справедливо и для диапазона ячеек. Однако нельзя копировать группу ячеек.

Для перемещения и копирования ячеек можно использовать также Буфер обмена (Clipboard) Windows. Для работы с Буфером обмена используются команды **Вырезать**, **Копировать** и **Вставить**, расположенные в меню **Действия** главного меню программы, а также соответствующие кнопки панели инструментов — ,  и .

Для перемещения ячейки (или диапазона ячеек) следует сначала применить к ней команду **Вырезать**. Эта команда удалит содержимое ячейки и поместит его в Буфер обмена. С помощью команды **Вставить** содержимое можно извлечь из Буфера обмена и поместить в текущую ячейку.

Для копирования ячейки необходимо воспользоваться парой команд **Копировать** — **Вставить**. Команда **Копировать** помещает в Буфер обмена копию ячейки, а командой **Вставить** содержимое Буфера обмена вставляется в текущую ячейку. Следует заметить, что копия ячейки остается в Буфере обмена, пока он не будет заполнен чем-либо еще. Это свойство можно использовать, когда необходимо создать несколько копий одной и той же ячейки.

Для вставки строки или столбца выполните следующие действия:

1. Выделите в таблице такое количество строк или столбцов, которое вы хотите вставить, и в том месте, которое должны будут занимать вставляемые строки или столбцы.
2. Поместите указатель мыши в пределах выделенного диапазона и нажатием правой кнопки мыши вызовите на экран контекстное меню.
3. В контекстном меню выберите пункт **Раздвинуть**.

Помимо функции **Раздвинуть** контекстного меню для выполнения вставки строк и столбцов можно использовать функцию **Раздвинуть** из меню **Действия** главного меню программы.

Внимание!

Следует учесть, что новые строки помещаются в таблицу сверху от выделенных, а новые столбцы — слева.

Для удаления строк или столбцов выполните следующие действия:

1. Выделите в таблице те строки или столбцы, которые необходимо удалить.
2. Поместите указатель мыши в пределах выделенного диапазона и нажатием правой кнопки мыши вызовите на экран контекстное меню.
3. В контекстном меню выберите пункт **Удалить**.

Выделенные строки или столбцы удаляются из таблицы вместе с содержащимся в них текстом. Их место занимают строки, расположенные снизу, или столбцы, расположенные справа.

Для очистки строк или столбцов выполните следующие действия:

1. Выделите в таблице те строки или столбцы, которые необходимо очистить.
2. Поместите указатель мыши в пределах выделенного диапазона и нажатием правой кнопки мыши вызовите на экран контекстное меню.

3. В контекстном меню выберите пункт **Очистить**. Из выделенных строк или столбцов удаляется все их содержимое. Если в контекстном меню выбрать пункт **Очистить текст**, то из выделенных строк или столбцов удаляется только содержащийся в них текст.

В случае необходимости можно вставить, удалить или очистить диапазон. Вставка диапазона несколько отличается от вставки строки или столбца.

Для вставки диапазона выполните следующие действия:

1. Выделите в таблице диапазон такого же размера, какой хотите вставить, и в том месте, которое должен будет занимать вставляемый диапазон.
2. Поместите указатель мыши в пределах выделенного диапазона и нажатием правой кнопки мыши вызовите на экран контекстное меню.
3. В контекстном меню выберите пункт **Раздвинуть**.
4. В выданном на экран запросе выберите, куда надо сдвинуть ячейки, чтобы освободить место для вставляемого диапазона: горизонтально — будут сдвинуты ячейки, расположенные справа от выделенного диапазона, вертикально — расположенные ниже.

Помимо функции **Раздвинуть** контекстного меню, для выполнения вставки диапазона можно использовать функцию **Раздвинуть** из меню **Действия** главного меню программы.

Для удаления диапазона выполните следующие действия:

1. Выделите в таблице диапазон, который необходимо удалить.
2. Поместите указатель мыши в пределах выделенного диапазона и нажатием правой кнопки мыши вызовите на экран контекстное меню.
3. В контекстном меню выберите пункт **Удалить**.
4. В выданном на экран запросе выберите, какой диапазон займет место удаленного: горизонтально — место удаленного займет диапазон, расположенный справа от него; вертикально — диапазон, расположенный ниже.
5. Нажмите кнопку **ОК**.

Очистка диапазона аналогична очистке строк и столбцов.

Команды управления движением курсора представлены в табл. 4.3.

Таблица 4.3. Команды управления движением курсора

Клавиши	Действие
<↑>	Переход курсора на строку вверх
<↓>	Переход курсора на следующую ячейку в строке

Таблица 4.3 (окончание)

Клавиши	Действие
<→>	Переход курсора на следующую ячейку в строке
<←>	Переход курсора на предыдущую ячейку в строке
<Shift>+<↑>	Отмечает еще одну ячейку таблицы от позиции курсора вверх
<Shift>+<↓>	Отмечает еще одну ячейку таблицы от позиции курсора вниз
<Shift>+<→>	Отмечает еще одну ячейку таблицы от позиции курсора вправо
<Shift>+<←>	Отмечает еще одну ячейку таблицы от позиции курсора влево
<Home>	Переход в начало таблицы
<End>	Переход в конец таблицы (задействованная часть таблицы)
<PgUp>	Переход на страницу вверх
<PgDn>	Переход на страницу вниз
<Shift>+<Home>	Отмечает ячейки от позиции курсора до начала таблицы
<Shift>+<End>	Отмечает ячейки от позиции курсора до конца таблицы
<Shift>+<PgUp>	Отмечает ячейки вверх от позиции курсора на страницу
<Shift>+<PgDn>	Отмечает ячейки вниз от позиции курсора на страницу
	Удаление следующего за курсором символа или отмеченного блока (если есть)
<Backspace>	Удаление предыдущего символа или отмеченного блока (если есть)

Форматирование ячеек

Большое значение для создания выразительных и информативных документов приобретает возможность использования разнообразных оформительских средств: шрифтов разнообразных гарнитур и размеров, обрамлений, различных фоновых рисунков, цветовых средств. Редактор таблиц системы 1С:Предприятие позволяет определять для любой ячейки или группы ячеек таблицы различные атрибуты форматирования. Установка атрибутов осуществляется в диалоге **Свойства**. Для вызова этого диалога на экран можно воспользоваться одним из следующих способов:

- нажатием правой кнопки мыши вызвать на экран контекстное меню; в этом меню выбрать пункт **Свойства**;

- в главном меню программы выбрать пункт **Действия**; в открывшемся подменю выбрать функцию **Свойства**.

Диалог **Свойства** организован в виде картотеки. Все свойства, которые вы можете определить для ячейки, объединены в 4 группы (рис. 4.41). Для доступа к управляющим элементам конкретной группы щелкните мышью на нужной вкладке в окне диалога. Зададим заголовок отчета, состоящий из номера и даты документа.

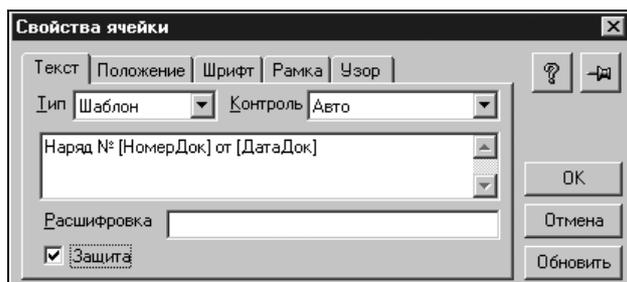


Рис. 4.41. Диалоговое окно **Свойства ячейки** на вкладке **Текст**

Пункты списка **Тип** устанавливают, какого рода информация введена в ячейку. Они не изменяют внешний вид ячейки, а используются только в процессе обработки шаблона при формировании готового отчета (рис. 4.42).

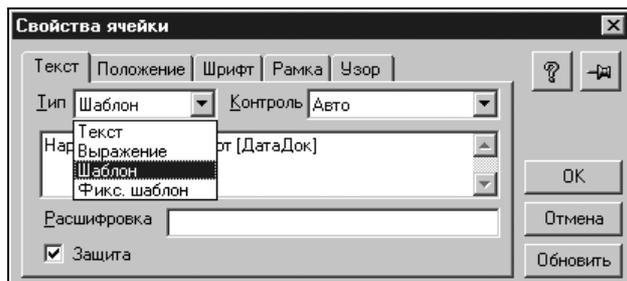


Рис. 4.42. Диалоговое окно **Свойства ячейки** на вкладке **Текст** со списком **Тип**

Тип влияет на формирование информации в ячейке следующим образом:

- **Текст** — текст, введенный в ячейку, при построении отчета будет копироваться в форму готового отчета без всякой дополнительной обработки;
- **Выражение** — текст, введенный в ячейку, при построении отчета будет интерпретироваться как выражение, вычисленное значение которого будет перенесено в форму готового отчета;
- **Шаблон** — информация в ячейке представляет собой текст с включенными в него выражениями на встроенном языке системы 1С:Предприятие,

заклученными в квадратные скобки. При формировании выражения из исходной таблицы будут вычислены, а их результаты будут включены в текст вместо самих выражений. Место, отводимое в тексте для вывода результатов выражений, определяется длиной этих результатов;

- **Фикс. шаблон** — фиксированный шаблон. Отличается от шаблона тем, что место, отводимое в тексте для вывода результатов выражений, определяется расстоянием между квадратными скобками, ограничивающими выражения в исходном тексте.

В нашем случае **НомерДок** и **ДатаДок** — реквизиты документа **Наряд**. Заданный шаблон "Наряд № [НомерДок] от [ДатаДок]" в процессе формирования отчета заполнится конкретными значениями номера и даты документа (рис. 4.43).

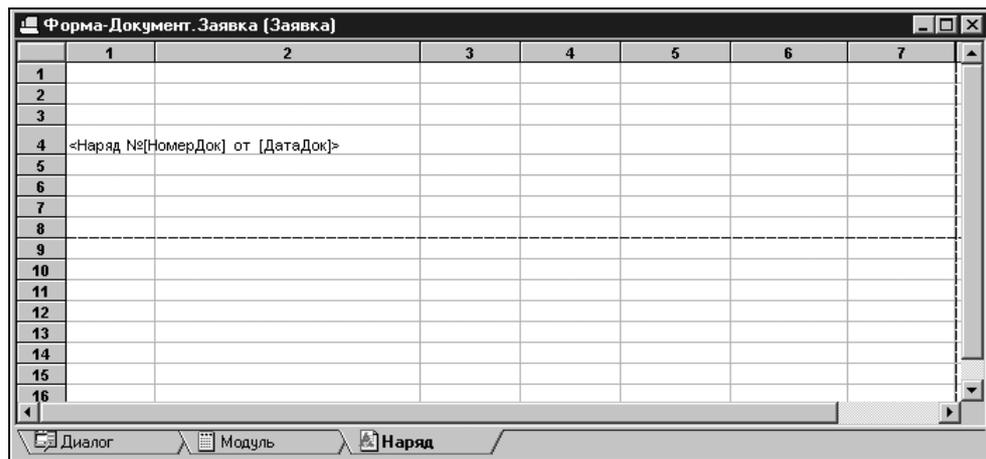


Рис. 4.43. Диалоговое окно формы документа **Заявка** на вкладке **Таблица** с заголовком

Пункты списка **Контроль** позволяют установить, как показывать текст на экране, если он не помещается в ячейке целиком (рис. 4.44).

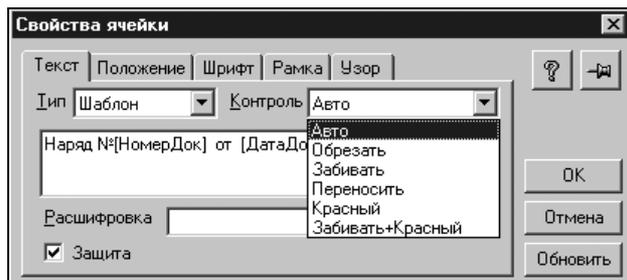


Рис. 4.44. Диалоговое окно **Свойства ячейки** на вкладке **Текст** со списком **Контроль**

Контроль влияет на формирование информации в ячейке следующим образом:

- Авто** — не помещающийся текст активной ячейки будет показан в соседних по горизонтали ячейках, если в них, в свою очередь, нет текста;
- Обрезать** — видимая часть текста будет ограничена левой и правой границами ячейки;
- Забивать** — видимая часть текста будет ограничена левой и правой границами ячейки, при этом все символы видимой части текста будут заменяться на символ #;
- Переносить** — не уместяющийся по ширине текст будет переноситься на следующие строки ячейки;
- Красный** — если первым символом является минус, весь текст изображается шрифтом красного цвета, независимо от того, какой цвет шрифта установлен во вкладке **Шрифт**;
- Забивать + Красный** — сочетание двух способов показа **Забивать** и **Красный**.

При создании шаблона табличного документа в Конфигураторе в поле **Расшифровка** можно указать выражение на встроенном языке системы 1С:Предприятие. При формировании табличного документа это выражение будет вычислено и его результат будет записан в поле **Расшифровка** ячейки готового документа. Когда готовый табличный документ открыт в режиме **Только просмотр**, то, если дважды щелкнуть левой кнопкой мыши на такой ячейке (или сделать ее активной и нажать клавишу <Enter>), значение будет выдано на экран для просмотра. Кроме того, для динамической обработки значения ячейки в модуль соответствующей формы может быть помещена предопределенная процедура со стандартным именем `ОбработкаЯчейкиТаблицы()`.

Если флажок **Защита** установлен, запрещается редактирование данной ячейки таблицы в режиме **Только просмотр** при работе 1С:Предприятия.

Зададим в ячейке 6-й строки, 1-го столбца таблицы адрес заявки с типом ячейки **Шаблон**, с контролем **Авто**. Адрес будет выводиться полностью в последующих ячейках (рис. 4.45). Зададим последовательно в ячейках восьмой строки таблицы шапку табличной части отчета со следующими заголовками с учетом вынужденных сокращений:

- № п/п;
- Услуга;
- Кол-во;
- Ед. изм.;

- Цена;
- Сумма;
- С учетом льгот.

Очевидно, что установленное по умолчанию значение контроля **Авто** позволило вылезти заголовку с учетом льгот за рамки ячейки шапки. Для правильного формирования заголовка необходимо сделать следующее:

1. В контекстном меню выбрать пункт **Свойства**.
2. В диалоговом окне **Свойства ячейки** выбрать вкладку **Тип**.
3. В поле **Контроль** выбрать значение **Переносить по словам**.
4. Растянуть строку шапки вниз на расстояние, достаточное для размещения перенесенной части заголовка (рис. 4.45).

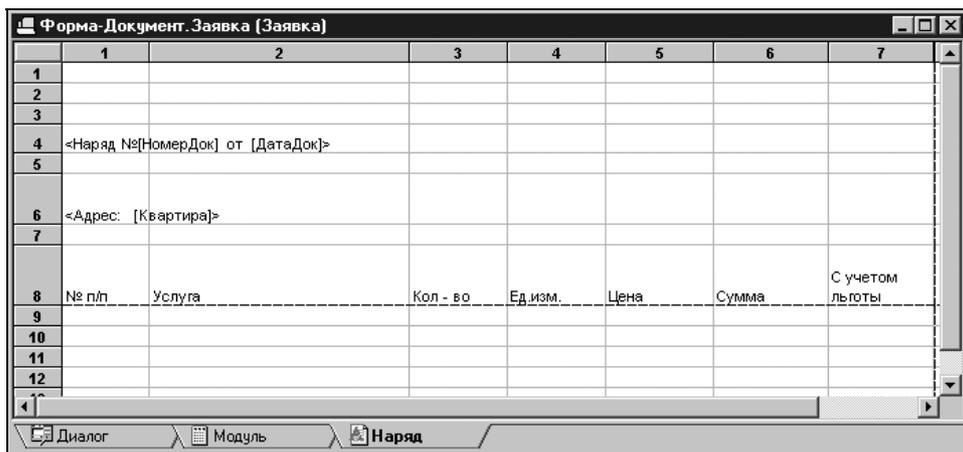


Рис. 4.45. Диалоговое окно формы документа Заявка на вкладке Таблица с шапкой

Для установки характеристик выравнивания текста в ячейке таблицы служат управляющие элементы вкладки **Положение** (рис. 4.46).

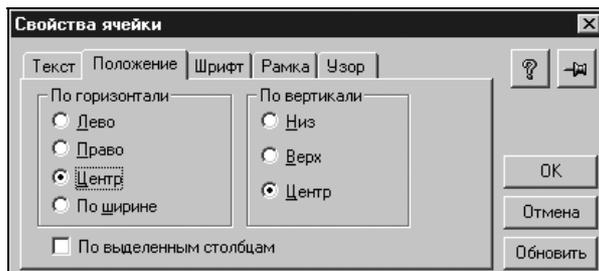


Рис. 4.46. Диалоговое окно Свойства ячейки на вкладке Положение

Вкладка **Положение** содержит:

□ По горизонтали:

- **Лево** — выравнивание текста по левой границе ячейки;
- **Право** — выравнивание текста по правой границе ячейки;
- **Центр** — текст размещается вдоль вертикальной линии, проходящей через центр ячейки;
- **По ширине** — текст выравнивается одновременно по правой и по левой границам ячейки. Высота и ширина ячейки автоматически устанавливаются такими, чтобы текст в ячейке был виден полностью.

□ По вертикали:

- **Низ** — текст располагается вдоль нижней границы ячейки;
- **Верх** — текст располагается вдоль верхней границы ячейки;
- **Центр** — текст размещается вдоль горизонтальной линии, проходящей через центр ячейки.

Внимание!

Независимо от вида выравнивания текст всегда обрезается по верхней и нижней границам ячейки.

Для расположения заголовка в центре отчета необходимо задать горизонтальное выравнивание по центру семи ячеек. Для задания такого выравнивания необходимо выполнить следующие действия:

1. Выделить ячейки с 1-го по 7-й столбцы.
2. В меню **Таблица** выбрать пункт **Объединить**.
3. Правой кнопкой мыши вызвать контекстное меню.
4. В контекстном меню выбрать пункт **Свойства**.
5. В диалоговом окне **Свойства ячейки** выбрать вкладку **Положение**.
6. В переключателе **По горизонтали** задать пункт **Центр**.

Для выравнивания текста в ячейках шапки необходимо выполнить следующие действия:

1. Выделить ячейки с 1-го по 7-й столбцы.
2. Правой кнопкой мыши вызвать контекстное меню.
3. В контекстном меню выбрать пункт **Свойства**.
4. В диалоговом окне **Свойства ячейки** выбрать вкладку **Положение**.
5. В переключателях **По горизонтали** и **По вертикали** задать пункт **Центр** (рис. 4.47).



Рис. 4.47. Диалоговое окно формы документа Заявка на вкладке Таблица с шапкой после выравнивания

Для задания атрибутов шрифта для текста активной ячейки служат управляющие элементы вкладки **Шрифт** в диалоговом окне **Свойства ячейки**. Они позволяют выбрать гарнитуру (начертание) шрифта, его размер и стиль (жирный, курсив, подчеркнутый) (рис. 4.48).

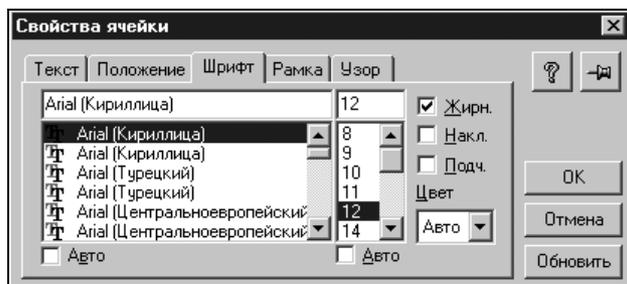


Рис. 4.48. Диалоговое окно Свойства ячейки на вкладке Шрифт

Вкладка **Шрифт** содержит стандартный набор реквизитов для выбора шрифта: список типов шрифта, список размеров шрифта, Жирный, Наклонный, Подчеркнутый. Поле **Цвет** позволяет установить цвет, которым будет изображаться шрифт на экране и при печати. Если в этом поле вместо конкретного цвета выбрать пункт **Авто**, то цвет шрифта будет определяться установками операционной системы Windows.

Зададим в ячейке заголовка шрифт Arial (Кириллица) высотой 12 с признаком Жирн. (рис. 4.48). Для ячейки адреса зададим шрифт Arial (Кириллица) высотой 10 с признаком Жирн. Для ячеек шапки зададим шрифт Courier (Кирил-

лица) высотой 10 с признаком Жирн., предварительно выделив все ячейки шапки (рис. 4.49).

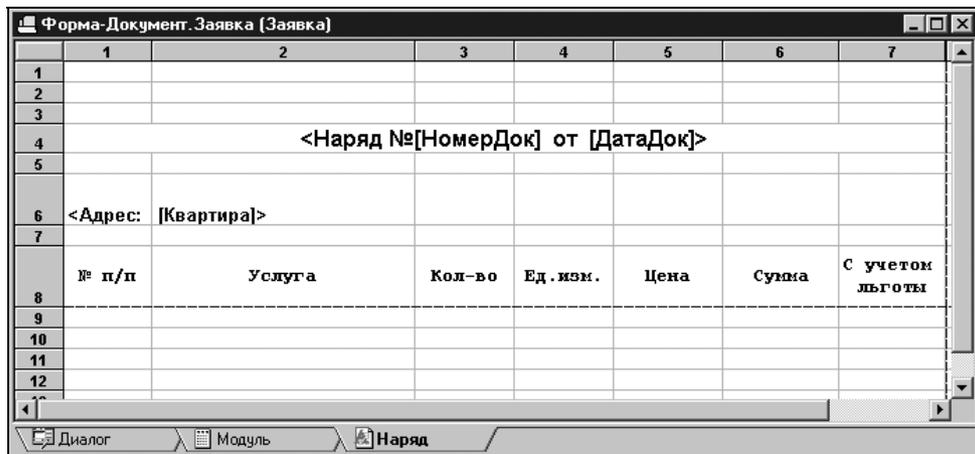


Рис. 4.49. Диалоговое окно формы документа Заявка на вкладке Таблица с измененным шрифтом

Управляющие элементы вкладки **Рамка** служат для создания оформления вокруг ячейки или группы ячеек таблицы (рис. 4.50).

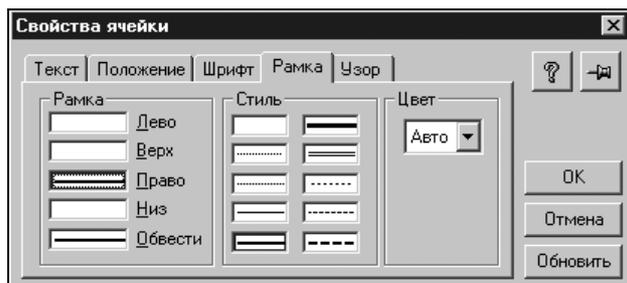


Рис. 4.50. Диалоговое окно Свойства ячейки на вкладке Рамка

Элементы группы **Стиль** предназначены для задания стиля линии оформления. Для установки стиля необходимо выбрать мышью нужный образец.

Управляющие элементы, сведенные в группу **Рамка**, позволяют установить, вдоль каких границ ячейки будет идти оформление. Для включения оформления достаточно выбрать мышью поле с названием нужной стороны ячейки. В выбранном поле будет изображен образец линии оформления.

Выбирая поля из групп **Рамка** и **Стиль** в произвольных сочетаниях, можно устанавливать разные стили линии оформления для различных сторон ячейки.

Особую функцию несет поле **Обвести**. Создание обрамления с помощью этого поля несколько отличается от создания обрамления с помощью других управляющих элементов этой закладки. При выборе мышью поля **Обвести** обрамление выбранного стиля выводится сразу вдоль всех сторон активной ячейки. Если вы не меняли активной ячейки, то повторным выбором этого же поля обрамление можно снять. Если вы выделили диапазон ячеек, то при выборе поля **Обвести** рамка будет помещена вокруг всего выделения, т. е. обрамление будет проходить по внешним сторонам крайних ячеек выделенной области. Если выделено несколько диапазонов ячеек, рамка устанавливается независимо для каждого выделенного диапазона. Если же вы хотите установить одинаковое обрамление для всех ячеек выделенного диапазона — используйте поля **Лево**, **Верх**, **Право**, **Низ** из группы **Рамка**. Если вы создали обрамление, используя поле **Обвести**, и сменили активную ячейку, то повторно вернуться к ячейке и отменить обрамление уже нельзя — после смены активной ячейки рамка превращается в совокупность четырех линий, и работать с ней придется обычным образом.

Оформим рамку шапки, для этого необходимо выполнить следующие действия:

1. Выделить ячейки шапки.
2. Правой кнопкой мыши вызвать контекстное меню.
3. В контекстном меню выбрать пункт **Свойства**.
4. В диалоговом окне **Свойства ячейки** выбрать вкладку **Рамка**.
5. В группе **Стиль** выбрать нижнюю позицию.
6. В группе **Рамка** задать линию щелчком мыши в позиции **Обвести** и **Право** (см. рис. 4.50).
7. Шапка отчета приняла официально допустимый вид (рис. 4.51).

Задания цвета фона ячейки, а также цвет и рисунок заполнения производится с помощью управляющих элементов вкладки **Узор** (рис. 4.52).

Вкладка **Узор** содержит следующие поля:

- Фон** — позволяет установить цвет, которым будет изображаться фон ячейки на экране и при печати. Если в этом поле вместо конкретного цвета выбрать пункт **Авто**, то цвет фона будет определяться установками операционной системы Windows;
- Узор** — позволяет установить узор, который будет изображаться в ячейке на экране и при печати. Если в этом поле вместо конкретного цвета выбрать пункт **Авто**, то узор будет определяться установками операционной системы Windows;

1	2	3	4	5	6	7	
1							
2							
3							
4	<Наряд №[НомерДок] от [ДатаДок]>						
5							
6	<Адрес: [Квартира]>						
7							
8	№ п/п	Услуга	Кол-во	Ед. изм.	Цена	Сумма	С учетом льготы
9							
10							
11							
12							

Рис. 4.51. Диалоговое окно формы документа Заявка на вкладке Таблица с выделенной шапкой



Рис. 4.52. Диалоговое окно Свойства ячейки на вкладке Узор

- **Цвет** — позволяет установить цвет, которым будет изображаться узор на экране и при печати. Если в этом поле вместо конкретного цвета выбрать пункт **Авто**, то цвет узора будет определяться установками операционной системы Windows.

Секции

В табличном редакторе 1С:Предприятие существует возможность создания именованных горизонтальных и вертикальных секций таблицы. При построении выходной формы обращение к различным частям шаблона из процедуры печати производится именно по присвоенным именам секций. Секции бывают горизонтальные (включают в себя отмеченные строки) и вертикальные (включают в себя отмеченные столбцы).

Создадим секцию **Шапка**. Для присвоения имени какой-либо секции таблицы нужно сначала отметить необходимые строки или столбцы. В нашей шап-

ке выделим строки с 1-й по 9-ю. После того как выделена область таблицы, которой будет присваиваться имя, необходимо в главном меню программы открыть пункт **Таблица**, и из появившегося меню выбрать функцию **Включить в секцию**. На экран будет выдан диалог **Идентификатор секции** (рис. 4.53).

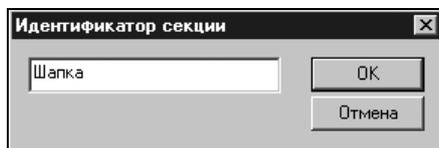


Рис. 4.53. Диалоговое окно **Идентификатор секции**

По умолчанию каждой новой секции присваивается имя **Секция** с очередным порядковым номером. Это имя можно изменить. Для присвоения нового имени области таблицы необходимо ввести желаемое имя в поле ввода и нажать кнопку **ОК**. Диалог будет убран с экрана, а поименованная область будет обведена тонкой рамкой красного цвета (рис. 4.54).

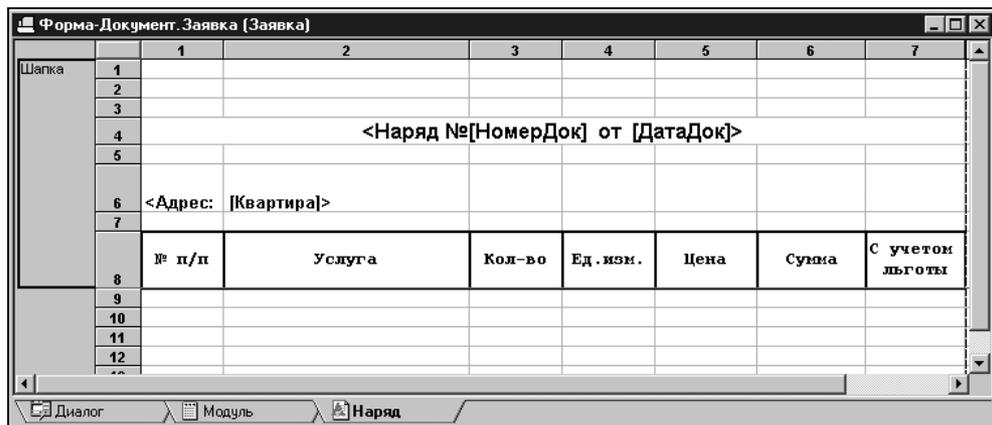


Рис. 4.54. Диалоговое окно формы документа **Заявка** на вкладке **Таблица** с секцией **Шапка**

Внимание!

Нельзя поименовать группу ячеек. Если выделить группу ячеек и попытаться присвоить ей имя, то имя будет присвоено секции, содержащей все строки или столбцы выбранного диапазона ячеек.

При работе с именованными секциями можно их переименовывать, удалять, расширять (включать в них дополнительно соседние строки или столбцы), создавать вложенные или внешние секции.

Для переименования или удаления именованной секции выберите из главного меню команду **Таблица**, в ней меню **Секции**, подменю **Горизонтальные** или **Вертикальные**. В появившемся диалоговом окне **Секции** можно выбрать нужную секцию, переименовать ее или удалить. Диалог содержит окно со списком уже присвоенных имен и управляющие элементы — кнопки.

Кнопка **Выбрать** служит для выделения именованной секции. Эту функцию удобно использовать, когда необходимо произвести какие-либо действия над всеми ячейками, входящими в секцию, например, изменить гарнитуру или размер шрифта.

Для переименования какой-либо секции таблицы служит кнопка **Изменить**. Предварительно необходимо в списке существующих имен выбрать имя, которое вы желаете изменить. Нажатие на кнопку **Изменить** вызывает диалог **Идентификатор секции**. Для присвоения нового имени области таблицы необходимо ввести желаемое имя в поле ввода и нажать кнопку **ОК**.

Для удаления какой-либо секции таблицы служит кнопка **Удалить**. Предварительно необходимо в списке существующих имен выбрать имя, которое вы желаете удалить. Нажатие на кнопку **Удалить** удаляет имя области из списка, красная рамка вокруг именованной области пропадает, но содержимое именованной области остается без изменений.

Внимание!

При удалении именованной секции содержимое ячеек не меняется, просто удаляется привязка имени к выделенной области строк или столбцов.

Кнопка **Отмена** служит для закрытия диалога **Секции**.

При работе с именованными секциями можно изменять размеры секции (расширять — включать дополнительно соседние строки или столбцы; сужать — исключать крайние строки или столбцы), создавать вложенные или внешние секции.

Для расширения секции отметьте нужные строки или столбцы. Выберите из главного меню пункт **Таблица**, меню **Включить в секцию**. В данном случае система выполняет одно из следующих действий:

- если отмеченная область строк или столбцов не пересекается с существующими секциями, то будет создана новая именованная секция;
- если отмеченная область строк или столбцов пересекается с существующей секцией, но охватывает только часть ее плюс соседние строки или столбцы, то существующая секция будет расширена за счет соседних отмеченных строк или столбцов;
- если отмеченная область строк или столбцов пересекается с существующей секцией, но не охватывает ее полностью, а только часть существую-

щей без соседних строк или столбцов, то будет создана новая секция, вложенная в существующую;

- если отмеченная область строк или столбцов охватывает полностью существующую секцию, то будет создана новая секция, внешняя для существующей, так что существующая будет вложена в новую.

Для сужения секции отметьте нужные строки или столбцы. Выберите из главного меню пункт **Таблица**, меню **Исключить из секции**. В данном случае система выполняет одно из следующих действий:

- если отмеченная область строк или столбцов не пересекается с существующими секциями, то никаких действий не произойдет;
- если отмеченная область строк или столбцов пересекается с существующей секцией, но охватывает ее не полностью, а только внутреннюю область без крайних строк или столбцов, то никаких действий не произойдет;
- если отмеченная область строк или столбцов пересекается с существующей секцией, но не охватывает ее полностью, а только крайнюю часть, то секция будет сужена за счет крайних отмеченных строк или столбцов;
- если отмеченная область строк или столбцов охватывает полностью существующую секцию, то секция будет удалена полностью.

Создадим табличную часть отчета. Поскольку все строки табличной части с точки зрения вида информации и форматирования идентичны, достаточно задать шаблон одной строки и динамически формировать строки по заданному шаблону из процедуры печати. Зададим ячейки строки табличной части отчета со следующими свойствами:

- тип ячейки — "Выражение" для всех ячеек;
- контроль — "Переносить по словам" для ячейки колонки "Услуга"; "Забывать" — для всех остальных;
- выравнивание по горизонтали:
 - № п/п — **центр**;
 - Услуга — **лево**;
 - Кол-во — **центр**;
 - Ед. изм. — **центр**;
 - Цена — **право**;
 - Сумма — **право**;
 - С учетом льгот — **право**;

- выравнивание по вертикали — **центр** для всех ячеек;
- шрифт — Arial (Кириллица), высота — 8;
- рамка — **обвести, право** для выделенных ячеек.

Значения ячеек будут сформированы из реквизитов документа из следующих выражений:

- № п/п — "к";
- Услуга — "Услуга";
- Кол-во — "Количество";
- Ед. изм. — "ЕдиницаИзмерения";
- Цена — "Цена";
- Сумма — "Цена*Количество";
- С учетом льгот — "Цена*Количество*КоэффициентЛьготы".

Включим созданную строку табличной части отчета в секцию с названием "Строка" (рис. 4.55).

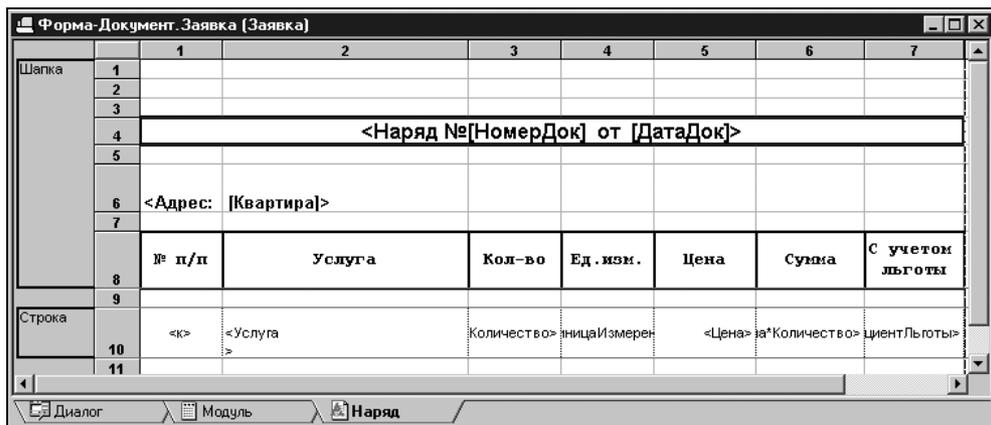


Рис. 4.55. Диалоговое окно формы документа Заявка на вкладке Таблица с секцией Строка

Обычно в отчетах в конце выводятся итоги и подписи ответственных лиц. Объединим ячейки 12-й строки, зададим шрифт Arial (Кириллица), высота — 8 **Жирн.**, выравнивание по горизонтали — **право**. Объединенной ячейке зададим тип Шаблон со значением — "Итого: [Итого("Всего")]".

Внимание!

Функция `Итого()` суммирует значения колонки табличной части документа только при условии, что в свойствах соответствующего реквизита документа установлен признак **Итого по колонке**.

Аналогично тому, как мы задавали адрес, зададим заявителя, дополнение и подпись мастера.

Чтобы **Итоги** вывелись в отчет, необходимо включить их в новую секцию (рис. 4.56).

	1	2	3	4	5	6	7	
Шапка	1							
	2							
	3							
	4	<Наряд №[НомерДок] от [ДатаДок]>						
	5							
	6	<Адрес: [Квартира]>						
	7							
	8	№ п/п	Услуга	Кол-во	Ед. изм.	Цена	Сумма	
	9						С учетом льготы	
Строка	10	<к>	<Услуга	Количество>	зницаИзмерен	<Цена> за*	Количество>	циентЛьготы>
	11		>					
Итоги	12	<Итого: [Итого("Всего")]>						
	13	<Заявитель: [Заявитель]>			<Дополнение			
	14	<Мастер / [Константа.ОтветственноеЛицо] >						
	15							
	16							

Рис. 4.56. Диалоговое окно формы документа Заявка на вкладке Таблица с секцией Итоги

Просмотр

В Конфигураторе предусмотрен режим просмотра созданного отчета на листе. Для вызова режима просмотра необходимо в меню **Файл** выбрать пункт **Просмотр** или на панели инструментов нажать кнопку . На рис. 4.57 отоброжен отчет на печатном листе в режиме просмотра.

Графические объекты

При создании шаблонов отчетов можно использовать графические объекты. Их применение позволяет улучшить оформление документов или внести в описание формы мелкие добавления, не затрагивая структуры шаблона.

К графическим элементам относятся линии, прямоугольники, надписи и внедренные объекты (OLE-объекты). Надписью называется прямоугольник с текстом.

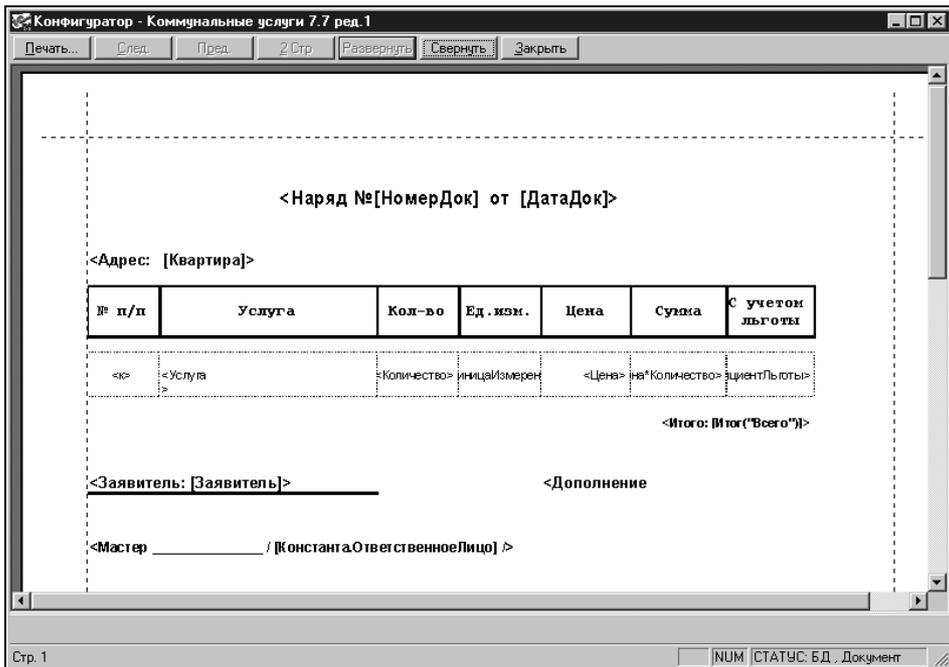


Рис. 4.57. Шаблон отчета "Наряд" в режиме просмотра

Добавление графических элементов в таблицу производится из главного меню программы: пункт **Таблица** меню **Вставить рисунок**. В меню **Вставить рисунок** следует выбрать мышью название того графического объекта, который вы хотите поместить в шаблон выходной формы. Теперь, если указатель мыши находится на рабочем поле таблицы, он примет вид тонкого перекрестия.

Для рисования линии:

1. Поместите указатель в ту позицию на рабочем поле таблицы, от которой вы хотите начать рисование линии.
2. Нажмите левую кнопку мыши и, не отпуская ее, переместите указатель мыши в позицию, где будет заканчиваться линия. Пунктирная линия, идущая от начальной позиции до указателя мыши, показывает положение линии после окончания рисования.
3. Отпустите левую кнопку мыши.

На рабочем поле появится линия, а курсор примет стандартный вид .

Для рисования прямоугольника или надписи:

1. Поместите указатель в ту позицию на рабочем поле таблицы, где должен находиться левый верхний угол прямоугольника.

2. Нажмите левую кнопку мыши и, не отпуская ее, переместите указатель мыши в позицию, где будет располагаться правый нижний угол прямоугольника. По мере движения указателя мыши пунктирная линия будет показывать примерное расположение и размеры будущего прямоугольника.
3. Отпустите левую кнопку мыши.

Если вы рисовали надпись, то после окончания рисования внутри прямоугольника появится текстовый курсор (мигающая вертикальная полоска), и можно будет начать ввод текста надписи.

Перед тем как проводить какие-либо манипуляции с графическим объектом, его необходимо выделить. Для выделения одного объекта надо поместить указатель мыши в границах объекта так, чтобы указатель принял форму стрелки, и нажать левую кнопку мыши. Если выделяется прямоугольник или надпись, то вокруг него появятся восемь небольших квадратных маркеров черного цвета: четыре из них расположены в середине каждой из четырех сторон рамки, и четыре — в ее углах. Эти маркеры служат для изменения размеров объекта. Если выделяется линия, то маркеры появятся только в ее начальной и конечной точках.

Для выделения нескольких графических объектов используйте следующий способ: выделите первый объект, затем нажмите клавишу <Shift> и, не отпуская ее, выделите остальные объекты. Восемь маркеров появятся на воображаемой рамке, включающей все выделенные объекты.

Для снятия выделения с объекта или объектов достаточно щелкнуть мышью в любом свободном от объектов месте таблицы.

Ничего страшного не случилось, если при создании графического объекта вы поместили его не в то место, в какое хотели. Любой графический объект можно перемещать по таблице, а также копировать, создавая несколько объектов на основе одного объекта-образца. Для перемещения графического объекта надо поместить на него указатель мыши, пока указатель не примет форму стрелки, нажать левую кнопку мыши, и, не отпуская ее, передвинуть объект мышью в новое положение. Пунктирная рамка (или линия — если перемещается линия), которая обозначает объект при движении мыши, поможет вам точнее разместить его на рабочем поле. Если необходимо переместить объект в ту часть таблицы, которая скрыта границами окна, — подведите указатель мыши к краю видимой части таблицы и таблица начнет автоматически "прокручиваться".

Копирование графического объекта выполняется почти так же, как и перемещение, но с некоторыми отличиями. Итак:

1. Поместите на объект указатель мыши, пока указатель не примет форму стрелки.

2. Нажмите и удерживайте левую кнопку мыши.
3. Нажмите и удерживайте клавишу <Ctrl>.
4. Мышью перенесите копию объекта в нужное положение.
5. Отпустите левую кнопку мыши.
6. Отпустите клавишу <Ctrl>.

Внимание!

Если в процессе переноса копии объекта отпустить клавишу <Ctrl>, объект будет перемещен, а не скопирован.

Для перемещения и копирования графических объектов можно использовать Буфер обмена (Clipboard) Windows. Для работы с Буфером обмена используются команды **Вырезать**, **Копировать** и **Вставить**, расположенные в меню **Редактирование** главного меню программы, или соответствующие кнопки панели инструментов ,  и .

Для перемещения графического объекта (или группы объектов) следует сначала применить к нему команду **Вырезать**. Эта команда удалит объект с рабочего поля и поместит его в Буфер обмена. С помощью команды **Вставить** объект можно извлечь из Буфера обмена и вновь поместить на рабочее поле. Положение вставленного объекта будет определяться следующим образом: в центр активной ячейки таблицы помещается левый верхний угол воображаемой прямоугольной рамки, проведенной вокруг вставляемых объектов.

Для копирования объекта необходимо поместить в Буфер обмена не сам объект, а его копию. Это делается командой **Копировать**. Вставляется объект так же, как это было описано ранее.

Следует заметить, что в любом случае копия объекта остается в Буфере обмена Windows. Это свойство можно использовать, когда необходимо создать несколько копий одного и того же объекта.

Размер любого графического объекта нетрудно изменить. Для этого используются маркеры, которые появляются вокруг объекта (или группы объектов) при его выделении.

Технология изменения размеров несколько отличается для линии и, скажем так, "прямоугольных" объектов. При выделении линии два маркера появляются у ее концов. Перемещая мышью любой маркер, можно изменять не только размер, но и наклон линии. При выделении "прямоугольного" объекта вокруг него появляется 8 маркеров. Четыре из них, расположенные на сторонах объекта, служат для изменения одного размера — высоты или ширины; другие четыре, расположенные в углах, используются для одновременного

изменения обоих размеров. Для изменения размеров объекта достаточно перетащить мышью один из маркеров.

Можно изменить размеры группы объектов. При выделении мышью нескольких объектов маркеры изменения размеров появляются на воображаемой рамке, охватывающей все выделенные объекты. Для изменения размеров выделенных объектов достаточно перетащить мышью один из маркеров, при этом размеры всех выделенных объектов изменятся пропорционально соотношению между начальным и конечным размерами области, занятой выделенными объектами.

Когда несколько графических объектов частично или полностью перекрывают друг друга, важно задать для них порядок расположения для достижения требуемого изобразительного эффекта. Новый рисованный объект всегда помещается на передний план и перекрывает объекты, оказавшиеся под ним. Но всегда можно принудительно поместить объект позади всех других объектов, и, наоборот, перенести вперед всех.

Для переноса объекта необходимо:

1. Выбрать объект мышью.
2. Нажатием правой кнопки мыши вызвать на экран контекстное меню.
3. В меню выбрать одну из функций:
 - **На передний план** — для переноса объекта вперед всех объектов;
 - **На задний план** — для размещения объекта позади других объектов.

Также для переноса объектов можно использовать главное меню программы пункт **Таблица**, меню **Положение рисунка**.

Для удаления рисованного объекта необходимо выделить его мышью и нажать клавишу . Также можно воспользоваться функцией **Удалить** из меню **Действия** главного меню программы.

Введем в верхний левый угол отчета возможный логотип предприятия, предварительно созданный в файле. Для вставки готового рисунка необходимо выполнить следующие действия:

1. В меню **Таблица** главного меню программы выбрать функцию **Вставить рисунок** и из открывшегося подменю выбрать пункт **Картинка**.
2. Нарисовать мышью прямоугольник желаемого размера в том месте, где предполагается поместить рисунок.
3. В предложенном диалоговом окне выбрать файл, содержащий рисунок (рис. 4.58).

Теперь вы можете изменить размеры рисунка, переместить его в любое место отчета, но отредактировать его в отчете невозможно (рис. 4.59).

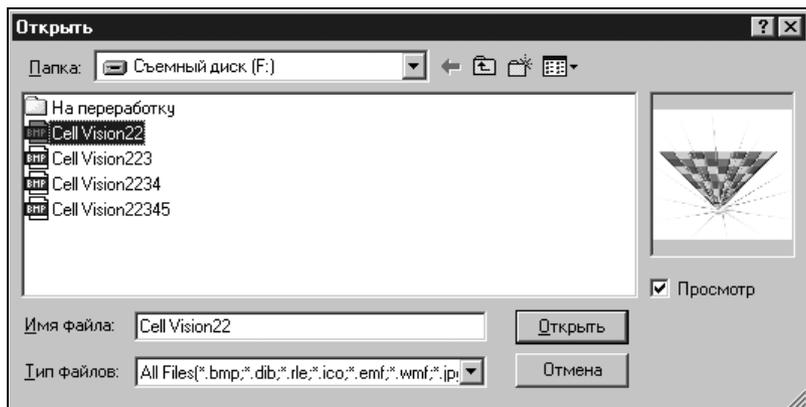


Рис. 4.58. Диалоговое окно выбора файла

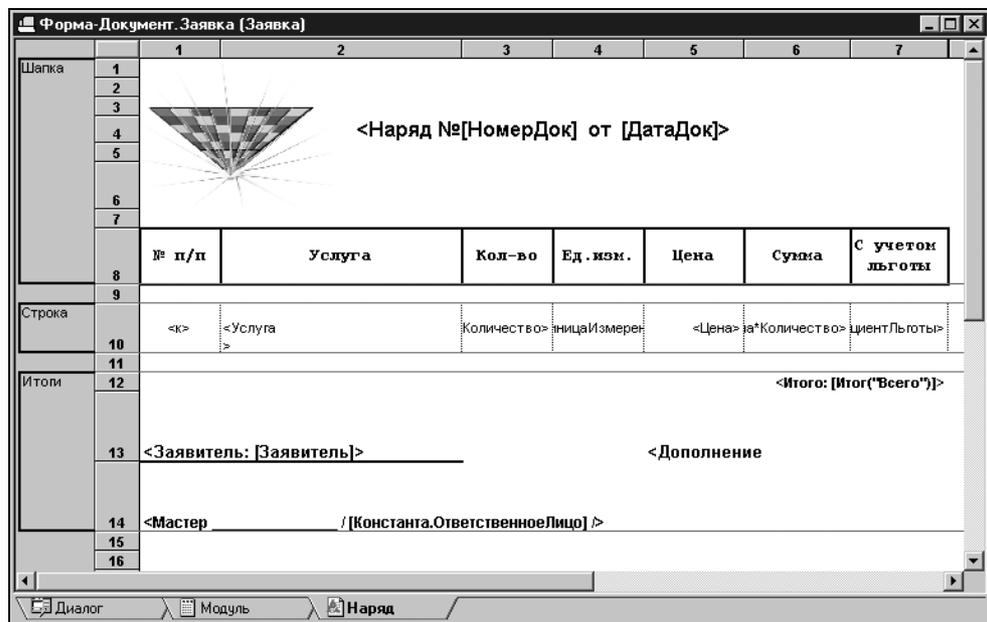


Рис. 4.59. Диалоговое окно формы документа Заявка на вкладке Таблица с рисунком

Следует обратить особое внимание на то, что рисунки помещаются в таблицу в рамке, что не всегда удобно. Для отмены рамки на рисунке необходимо выполнить следующие действия:

1. На рисунке вызвать правой кнопкой мыши контекстное меню.
2. В контекстном меню выбрать пункт **Свойства**.

3. В диалоге **Свойства рисунка** выбрать вкладку **Линия**.
В поле **Стиль** выбрать пустую строку (рис. 4.60).

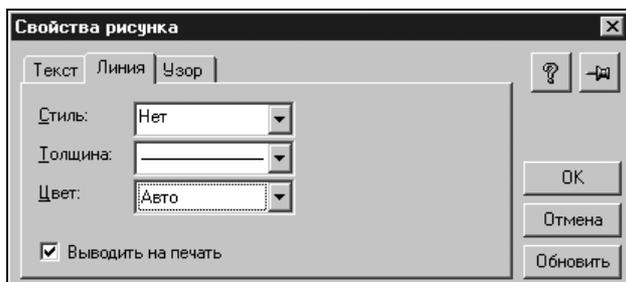


Рис. 4.60. Диалоговое окно **Свойства рисунка**

Одной из универсальных возможностей приложений, работающих в операционной системе Windows, является возможность использования внедренных объектов, называемых также OLE-объектами. OLE-объект — это часть текущего документа, которая создана другим Windows-приложением. Такой OLE-объект, как и любой рисованный объект, является полноценной частью документа с той лишь разницей, что для редактирования такого объекта будет загружено приложение, которым этот объект создан.

Для создания OLE-объекта необходимо выполнить следующие действия:

1. В меню **Таблица** главного меню программы выбрать функцию **Вставить рисунок** и из открывшегося подменю выбрать пункт **Объект**.
2. Нарисовать мышью прямоугольник желаемого размера в том месте, где предполагается поместить OLE-объект.
3. Из выданного на экран запроса выставить переключатель в положение **Создать из файла** (рис. 4.61).
4. Нажать кнопку **Обзор**.
5. В предложенном диалоговом окне выбрать файл, содержащий рисунок (см. рис. 4.58).
6. Нажать кнопку **ОК** (рис. 4.61).

Внимание!

Программы для создания OLE-объектов не входят в комплект поставки. Это отдельные Windows-приложения, являющиеся лицензионными программными продуктами. Об использовании вызванной программы следует читать в документации, поставляемой с ней.

После создания OLE-объекта он обладает всеми свойствами, присущими "обычным" рисованным объектам: вы можете менять его размер и положение.

ние, копировать, используя мышь или Буфер обмена, как угодно компоновать с другими объектами и удалять за ненадобностью. Но кроме этого OLE-объект может быть отредактирован — его содержимое можно изменить. Для редактирования OLE-объекта необходимо дважды щелкнуть на нем мышью.

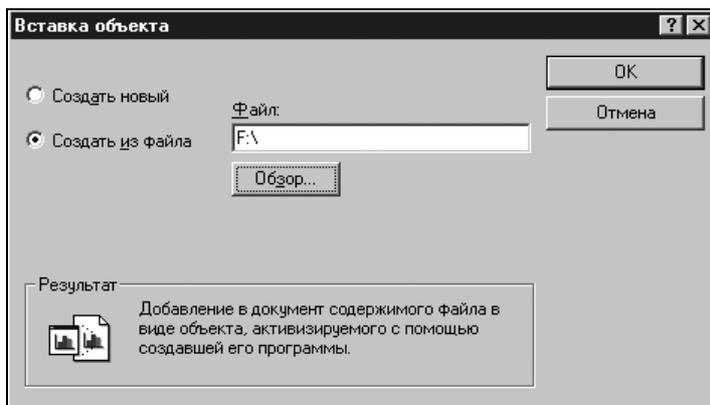


Рис. 4.61. Диалоговое окно Вставка объекта

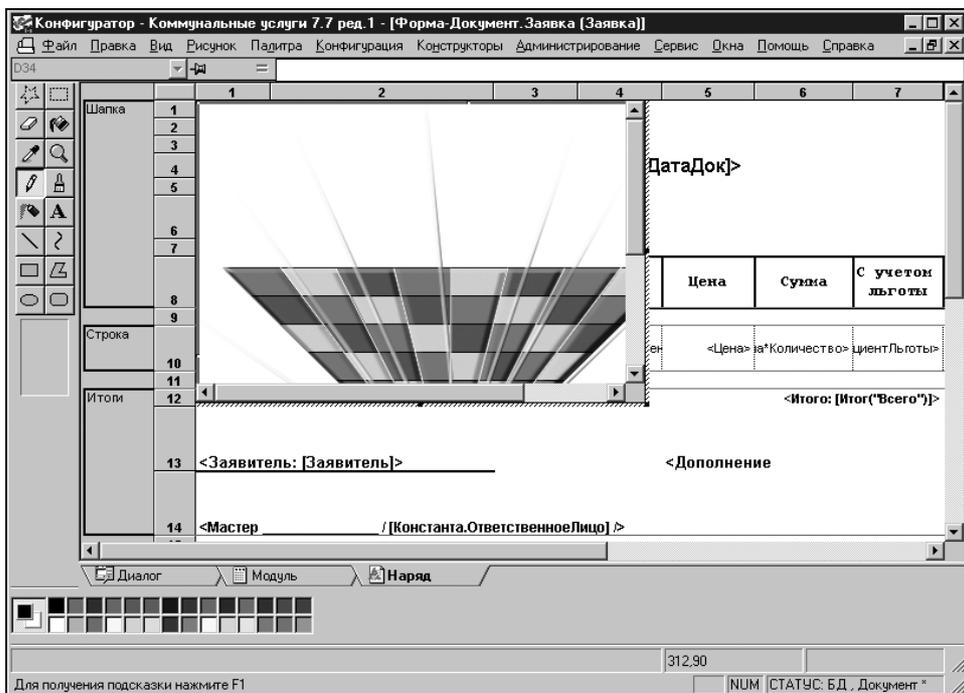


Рис. 4.62. Диалоговое окно формы документа Заявка в режиме редактирования OLE-объекта

В результате будет загружена та Windows-программа, в которой был создан данный объект, и вы можете отредактировать внедренный объект. При этом внешний вид окна 1С:Предприятие несколько изменится: в главном и контекстных меню появятся функции, а на рабочем поле — панели инструментов, характерные для вызванной программы. Теперь вы можете отредактировать внедренный объект (рис. 4.62).

Колонтитулы

Колонтитулы — это области сверху и внизу страницы, служащие для вывода какого-либо повторяющегося текста на каждой странице документа: номера страницы, названия документа, даты и времени создания и другой информации.

Создадим колонтитул, в котором выведем дату создания отчета и имя пользователя. Для создания колонтитула используется функция **Колонтитул** из меню **Таблица** главного меню программы. Из выданного на экран подменю вы можете выбрать, какой колонтитул необходимо создать — верхний или нижний.

Все управление внешним видом колонтитула осуществляется в выданном на экран диалоге (рис. 4.63).

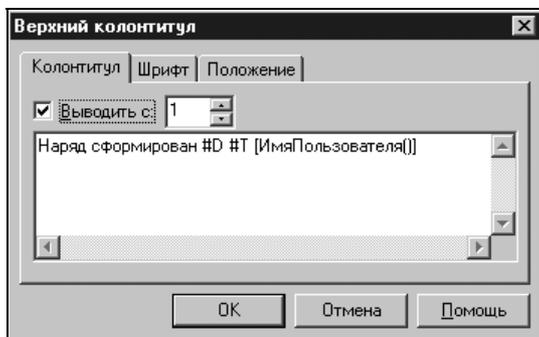


Рис. 4.63. Диалоговое окно **Верхний колонтитул** на вкладке **Колонтитул**

Вкладка **Колонтитул** позволяет задать текст и высоту колонтитула, а также управлять выводом колонтитула на печать.

Опция **Выводить** позволяет разрешить или запретить вывод колонтитула. Текст колонтитула и параметры его форматирования остаются без изменения.

Поле, расположенное справа от опции **Выводить**, позволяет установить границы вывода текста колонтитула в миллиметрах. В данном случае высота колонтитулов отсчитывается от границ зоны печати, установленных при настройке страницы.

Многострочное поле служит для ввода текста, который будет помещаться в колонтитул. Текст колонтитула может включать следующие управляющие конструкции:

- #T — при печати в данном месте будет вставлено время печати документа;
- #D — при печати в данном месте будет вставлена дата печати документа;
- #P — при печати в данном месте будет вставлен номер страницы документа.

В текст колонтитула может быть включено выражение на встроенном языке системы 1С:Предприятие, заключенное в квадратные скобки. Выражение вычисляется при формировании документа и его результат помещается в текст колонтитула. В нашем случае введем текст:

Наряд сформирован #D #T [ИмяПользователя()]

Для установки характеристик выравнивания текста колонтитула служат управляющие элементы вкладки **Положение**:

□ **По горизонтали:**

- **Лево** — выравнивание текста по левой границе колонтитула;
- **Право** — выравнивание текста по правой границе колонтитула;
- **Центр** — текст размещается вдоль вертикальной линии, проходящей через центр колонтитула;
- **По ширине** — текст выравнивается одновременно по правой и по левой границам колонтитула.

□ **По вертикали:**

- **Низ** — текст располагается вдоль нижней границы колонтитула;
- **Верх** — текст располагается вдоль верхней границы колонтитула;
- **Центр** — текст размещается вдоль горизонтальной линии, проходящей через центр колонтитула.

Зададим выравнивание по горизонтали — **Право** (рис. 4.64).

Для задания атрибутов шрифта для текста колонтитула служат управляющие элементы вкладки **Шрифт** (рис. 4.65).

Управляющие элементы позволяют выбрать гарнитуру (начертание) шрифта, его размер и стиль (жирный, курсив, подчеркнутый). Поле **Цвет** позволяет установить цвет, которым будет изображаться шрифт на экране и при печати. Если в этом поле вместо конкретного цвета выбрать пункт **Авто**, то цвет шрифта будет определяться установками операционной системы Windows.

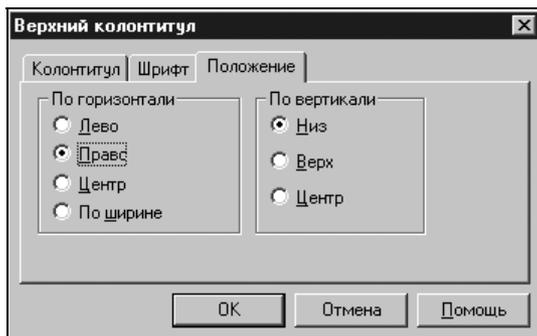


Рис. 4.64. Диалоговое окно **Верхний колонтитул** на вкладке **Положение**

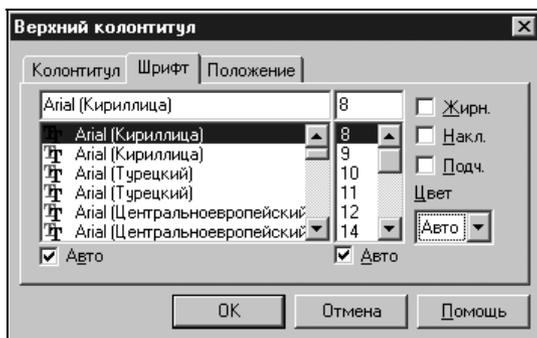


Рис. 4.65. Диалоговое окно **Верхний колонтитул** на вкладке **Шрифт**

Внимание!

Колонтитулы не отображаются ни в шаблоне отчета, ни в сформированном отчете на печать. Колонтитулы можно увидеть либо в режиме просмотра, либо в напечатанном отчете (рис. 4.66).

Меню редактора таблиц

Следующий список содержит специфические команды главного меню, доступные в режиме работы с редактором таблиц.

☐ Меню **Файл**:

- **Новый** — позволяет создать новый файл таблицы;
- **Открыть** — вызывает окно диалога **Открыть**, позволяя открыть существующий табличный файл;
- **Закрывать** — закрывает текущий активный табличный файл;
- **Сохранить** — записывает на диск текущий активный табличный файл;

- **Очистить** — очищает отмеченную часть таблицы;
- **Раздвинуть** — добавляет в таблицу часть, равноценную отмеченной части таблицы;
- **Удалить** — удаляет отмеченную часть таблицы;
- **Очистить текст** — очищает текст из ячеек в отмеченной части таблицы;
- **Поиск/Искать** — вызывает окно диалога **Поиск**. Он позволяет ввести с клавиатуры искомое слово или фразу, а затем найти его следующее появление вперед или назад по таблице от текущей позиции курсора;
- **Поиск/Искать вперед** — повторяет последний поиск вперед по таблице;
- **Поиск/Искать назад** — повторяет последний поиск назад по таблице;
- **Заменить** — вызывает окно диалога **Замена**, который позволяет вам заменить какой-либо участок текста. Вы можете проводить замену текста либо один раз, либо по отмеченной части таблицы, либо по всей таблице;
- **Свойства** — вызывает окно диалога **Свойства**.

☐ Меню **Таблица**:

- [**Включить в секцию** — включает в секцию выделенные ячейки;
-] **Исключить из секции** — исключает из секции выделенные ячейки;
- **Секции** — вызывает окно диалога **Секции**;
- **Высота строки** — вызывает окно диалога **Высота строки**, позволяющего установить высоту отмеченной строки;
- **Ширина столбца** — вызывает окно диалога **Ширина столбца**, позволяющего установить ширину отмеченного столбца;
- **Колонтитул** — вызывает окно диалога **Колонтитул**;
- **Вставить рисунок** — вызывает диалог выбора рисунка или объекта или задает простые графические элементы типа линии, прямоугольника или текста;
- **Положение рисунка** — задает положение рисунка на заднем или переднем плане;
- **Вставить перевод страницы** — вставляет в таблицу команду перевода страницы;
- **Убрать перевод страницы** — убирает из таблицы команду перевода страницы.

☐ Меню Вид:

- **Заголовки** — включить/выключить режим отображения заголовков строк и столбцов;
- **Сетка** — включить/выключить режим отображения сетки таблицы;
- **Секции** — включить/выключить режим отображения секций;
- **Идентификаторы** — включить/выключить режим отображения имен секций;
- **Только просмотр** — включить/выключить режим работы редактора таблиц "Только просмотр";
- **Зафиксировать (Расфиксировать)** — зафиксировать/расфиксировать панели окна.

Для проведения замены в диалоговом окне **Замена** (рис. 4.67) введите с клавиатуры слово или фразу, которые нужно заменить в поле ввода **Искать**. Введите с клавиатуры замещаемый текст в поле ввода **Заменить**. Укажите требуемые режимы замены: **Учитывать регистр**, **Ячейки целиком**. Нажмите кнопку **Искать**, если необходимо найти следующее появление искомого слова или фразы в тексте. Нажмите кнопку **Заменить**, когда нашли. Нажмите кнопку **Заменить все**, если необходимо заменить все вхождения по всему тексту. Окно диалога **Замена** не является модальным. Это значит, что это окно остается на экране, так что вы можете легко выходить и возвращаться в него для повторения операций замены.

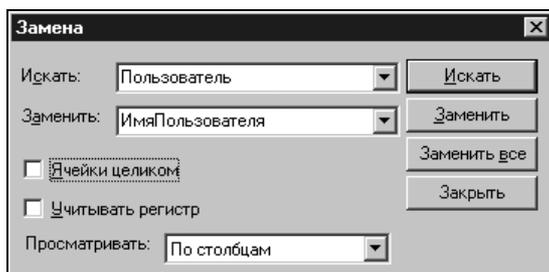


Рис. 4.67. Диалоговое окно **Замена**

Нажатие правой кнопки мыши в любой точке окна редактирования таблицы вызывает на экран контекстное меню.

Контекстное меню содержит следующие пункты:

- ☐ **Вырезать** (<Ctrl>+<X>) — удаляет отмеченную часть таблицы и помещает ее в обменный буфер;
- ☐ **Запомнить** (<Ctrl>+<C>) — помещает в обменный буфер копию отмеченной части таблицы;

- ❑ **Вставить** (<Ctrl>+<V>) — вставляет в текущую ячейку в точку курсора содержимое обменного буфера;
- ❑ **Очистить** — очищает отмеченную часть таблицы;
- ❑ **Раздвинуть** — добавляет в таблицу часть, равноценную отмеченной части таблицы;
- ❑ **Удалить** — удаляет отмеченную часть таблицы;
- ❑ **Очистить текст** — очищает текст из ячеек в отмеченной части таблицы;
- ❑ **Высота строки** — вызывает окно диалога **Высота строки**, позволяющего установить высоту отмеченной строки;
- ❑ **Ширина столбца** — вызывает окно диалога **Ширина столбца**, позволяющего установить ширину отмеченного столбца;
- ❑ **Свойства** — вызывает окно диалога **Свойства**.

Процедура

Формирование отчета запускается по нажатию кнопки, которой оно было назначено на вкладке **Дополнительно** (рис. 4.68).

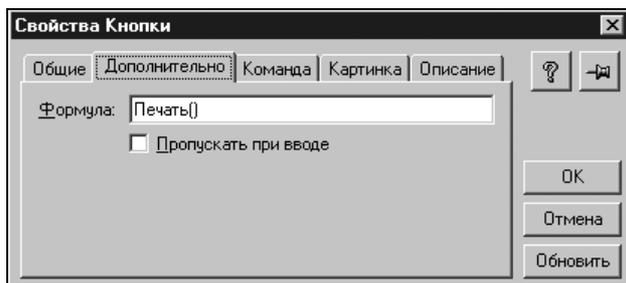


Рис. 4.68. Диалоговое окно **Свойства Кнопки**

Для нашего отчета необходимо организовать обращение к таблице и вывод соответствующих ее секций:

Процедура **Печать ()**

```
Таб=СоздатьОбъект ("Таблица") ;  
Таб.ИсходнаяТаблица ("Наряд") ;  
Таб.ВывестиСекцию ("Шапка") ;  
ВыбратьСтроки () ;  
КоэффициентЛьготы = Квартира.Льгота.Получить (ДатаДок) ;  
к=0 ;
```

```
Пока ПолучитьСтроку () =1 Цикл  
    к=к+1 ;  
    Таб.ВывестиСекцию ("Строка") ;
```


установить указатель на пункт **Отчеты** и в меню **Действия** выбрать пункт **Новый элемент**. Можно также воспользоваться выбором в контекстном меню пункта **Новый Отчет** (рис. 4.70).

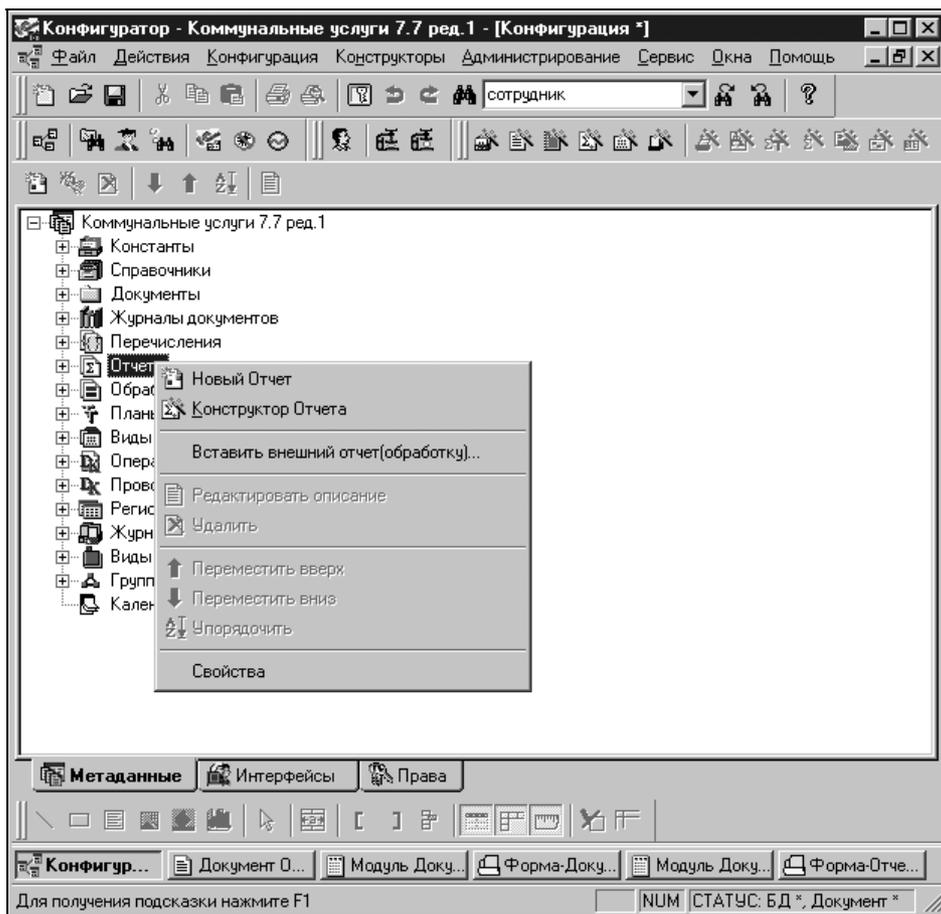


Рис. 4.70. Контекстное меню отчетов

Для задания основных свойств объекта метаданных предлагается заполнить реквизиты **Идентификатор**, **Синоним** и **Комментарий** на вкладке **Общие** диалогового окна **Свойства Отчета** (рис. 4.71).

Поскольку отчеты предназначены для формирования различных выборок данных, в форме отчета обычно задаются такие параметры отчета, как период выборки, группа или элемент справочника для детализации. В нашем отчете такими параметрами будут выступать месяц начисления платы за коммунальные услуги и группа квартир.

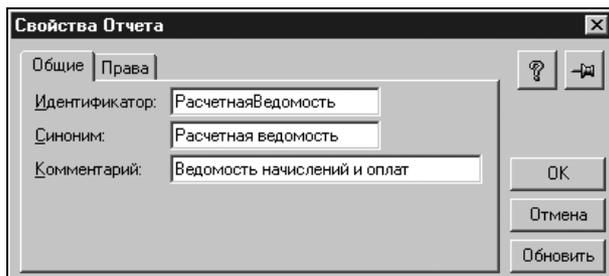


Рис. 4.71. Диалоговое окно **Свойства Отчета**

Рассмотрим несколько стандартных приемов задания периода выборки отчета. Задание периода выборки присутствует практически во всех расчетах, поэтому соответствующие функции выносятся в Глобальный модуль. Зададим периодичность задания отчета — месяц (рис. 4.72).

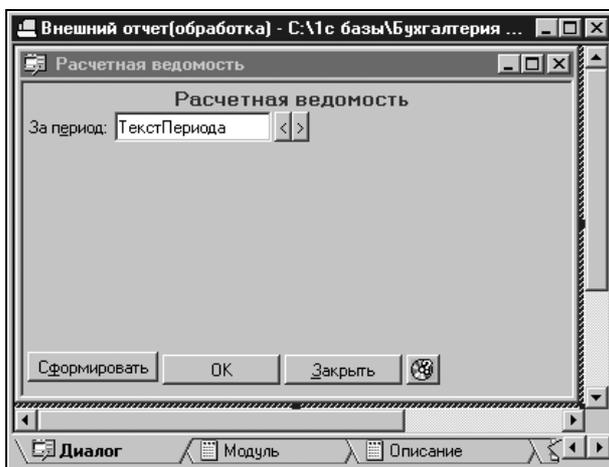


Рис. 4.72. Диалоговое окно отчета **Расчетная ведомость** на вкладке **Диалог** с заданием выбора периода

В нашем случае реализация задания периода выборки расчетов осуществляется вызовом функции Глобального модуля `глСменитьПериод()`, который располагается в диалоге свойств кнопок  и  на вкладке **Дополнительно** (рис. 4.73).

Для кнопки  уменьшения месяца:

`ТекстПериода = глСменитьПериод(ДатаАктуальности, -1)`

Для кнопки  увеличения месяца:

`ТекстПериода = глСменитьПериод(ДатаАктуальности, +1)`

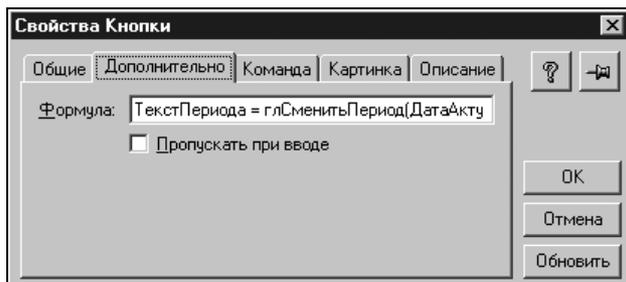


Рис. 4.73. Диалоговое окно **Свойства Кнопки** уменьшения месяца

Период даты актуальности выставляется вызовом функции Глобального модуля `глСменитьПериод(ДатаАктуальности)` из предопределенной процедуры `ПриОткрытии()`, которая располагается в теле модуля отчета:

функция `глСменитьПериод(ДатаАктуальности, Направление=0)` **Экспорт**

`ЖР = СоздатьОбъект("ЖурналРасчетов.Начисления");`

Если `ТипЗначения(ДатаАктуальности) <> 3` **Тогда**

`ДатаАктуальности = ЖР.КонецТекущегоПериода()`

КонецЕсли;

Если `Направление = 1` **Тогда**

`ДатаАктуальности = ЖР.КонецПериодаПоДате (`
`ЖР.КонецПериодаПоДате(ДатаАктуальности)+1)`

ИначеЕсли `Направление=-1` **Тогда**

`ДатаАктуальности = ЖР.НачалоПериодаПоДате(ДатаАктуальности)-1`

КонецЕсли;

`Возврат ЖР.ОписательПериода(ДатаАктуальности)`

Конецфункции // `глСменитьПериод`

Для задания произвольного периода выборки отчета существует системный вызов `ВвестиПериод(Дата1, Дата2)`. Он назначается стандартной кнопке  (рис. 4.74).

Назначим системный вызов `ВвестиПериод(Дата1, Дата2)` кнопке  (рис. 4.75).

Нажатие данной кнопки вызывает при выполнении стандартное диалоговое окно **Выбор периода** (рис. 4.76).

Инициация текущего периода производится в предопределенной процедуре `ПриОткрытии()` двумя операторами:

`Дата1 = НачалоПериодаБИ();`

`Дата2 = КонецПериодаБИ();`

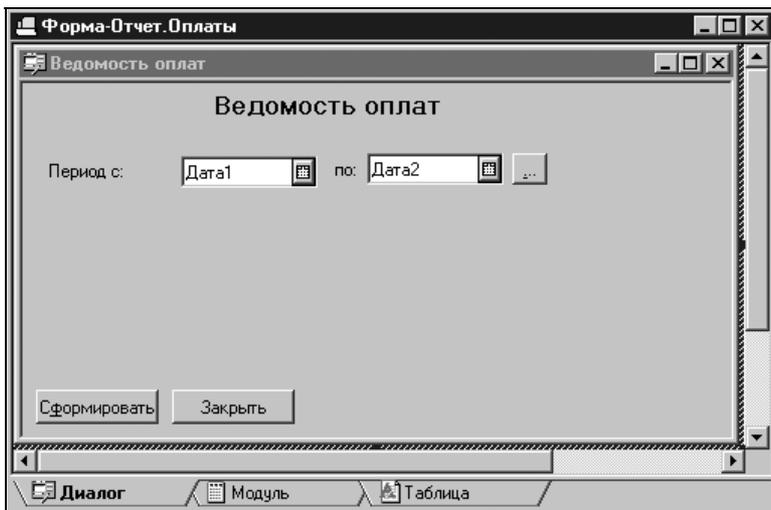


Рис. 4.74. Диалоговое окно отчета **Ведомость оплат**

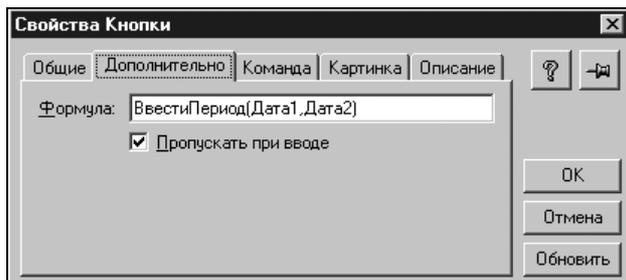


Рис. 4.75. Диалоговое окно **Свойства Кнопки** запроса периода

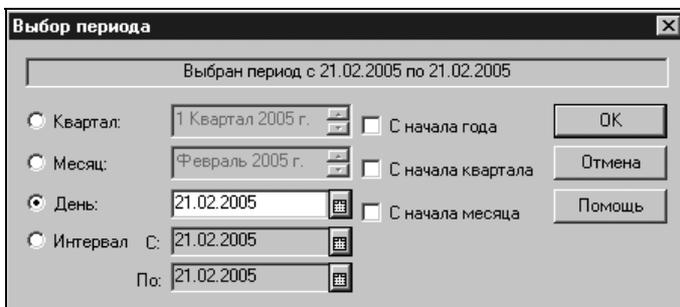


Рис. 4.76. Диалоговое окно **Выбор периода**

Внимание!

Функции `НачалоПериодаБИ()` и `КонецПериодаБИ()` берут свои значения из настройки параметров системы на вкладке **Бухгалтерские итоги** (рис. 4.77).

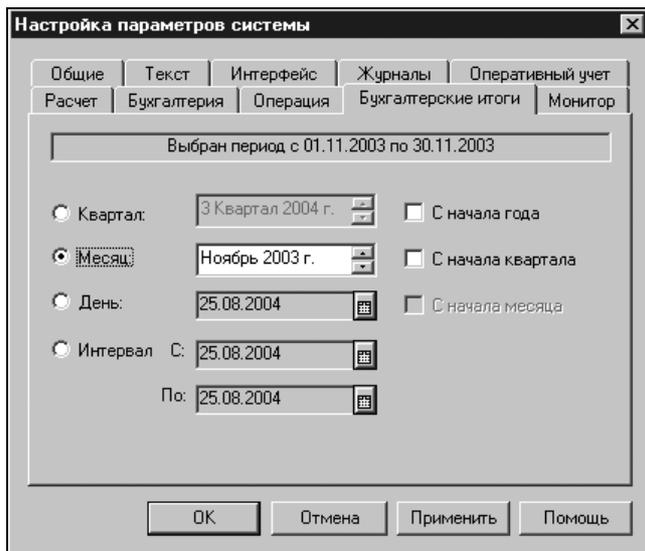


Рис. 4.77. Диалоговое окно **Настройка параметров системы**

Как правило, в настройке параметров требуется более одного варианта задания параметров выборки. В отчете **Расчетная ведомость** зададим следующие основные варианты задания выборки квартир (рис. 4.78):

- по всем — с упорядочиванием по квартирам;
- по дому;
- по списку квартир.

При задании варианта выборки **по дому** выдается диалоговое окно справочника "Дома" для выбора дома. Можно задать отчет с заданием произвольной выборки квартир. Для этого зададим список квартир и кнопки различных действий со списком (рис. 4.79).

Список квартир может быть большой, и формировать его подбором утомительно и долго. Следовательно, необходимо предусмотреть комбинацию основных вариантов задания выборки с заданием списка. Для этого добавим кнопку **Заполнить**, по нажатию которой будет создаваться список квартир по заданному в переключателе варианту. Созданный список квартир может по-

том редактироваться пользователем с помощью кнопок **Подбор**, **Удалить**, **Очистить**, **Сортировать** (рис. 4.80).

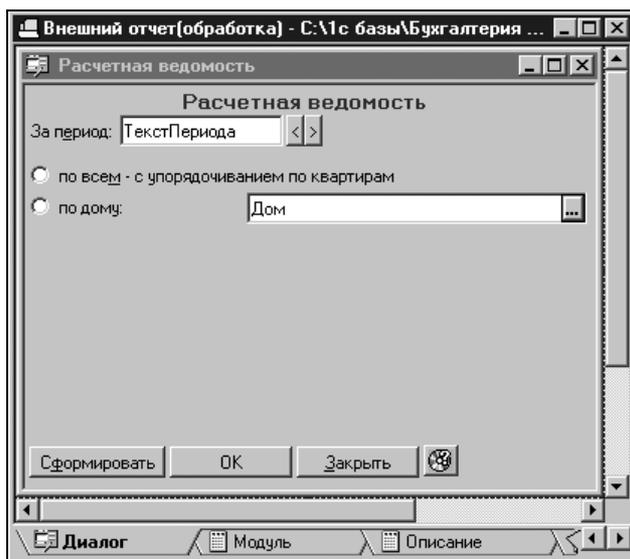


Рис. 4.78. Диалоговое окно отчета **Расчетная ведомость** на вкладке **Диалог** с заданием выбора квартир

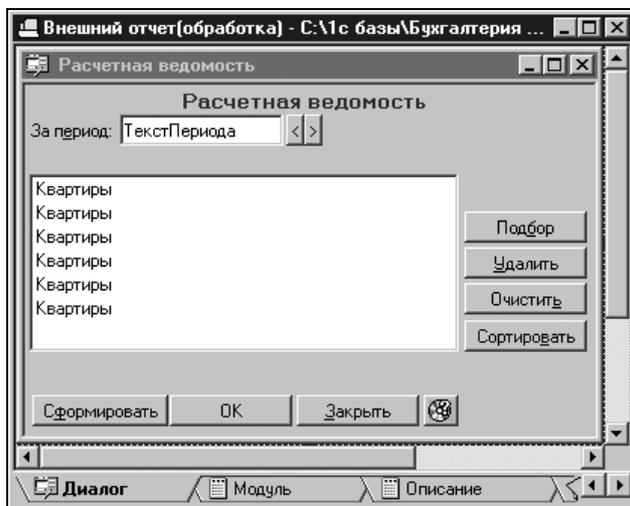


Рис. 4.79. Диалоговое окно отчета **Расчетная ведомость** на вкладке **Диалог** с заданием списка квартир

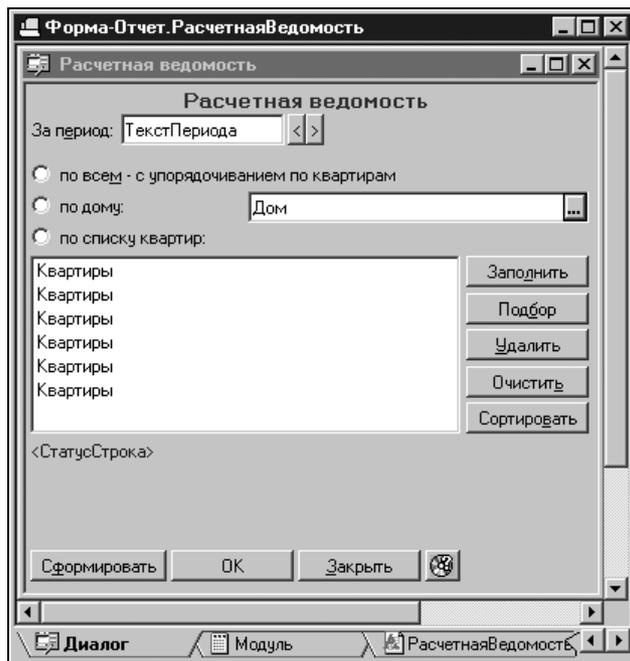


Рис. 4.80. Диалоговое окно отчета Расчетная ведомость на вкладке Диалог

Процедура заполнения списка просто реализуется с помощью механизма запросов:

Процедура Заполнить ()

Если Квартиры.РазмерСписка () > 0 **Тогда**

Если Вопрос ("Перед заполнением список квартир будет очищен."
"Продолжить?" , "Да+Нет")="Нет" **Тогда**

Возврат

КонецЕсли ;

КонецЕсли ;

ТаблицаКвартир = СоздатьОбъект ("ТаблицаЗначений") ;

ТаблицаКвартир.НоваяКолонка ("Квартира") ;

ПустаяГруппа = ПолучитьПустоеЗначение ("Справочник.Квартиры") ;

ПустойДокумент = ПолучитьПустоеЗначение ("Документ") ;

ТекстЗапроса =

"/ / { ЗАПРОС (Сформировать)

|Период с ДатаАктуальности по ДатаАктуальности;

|Квартира = Справочник.Квартиры.ТекущийЭлемент;

|КвРодитель = Справочник.Квартиры.Родитель;

|Группировка Квартира Упорядочить по Квартира.Наименование без групп;

```
| Условие (Квартира.ПометкаУдаления()=0) ;
```

```
| "///} } ЗАПРОС
```

```
;
```

Если ПустоеЗначение (Дом)=0 **Тогда**

```
ТекстЗапроса = ТекстЗапроса+"КвартДом = Справочник.Квартиры.Дом;"
"Условие (КвартДом в Дом) ;"
```

КонецЕсли ;

```
НачалоПериода = НачМесяца (ДатаАктуальности) ;
```

```
ОкончаниеПериода = КонМесяца (ДатаАктуальности) ;
```

```
// сформируем запрос и выгрузим результат в список квартир
```

```
Запрос = СоздатьОбъект ("Запрос") ;
```

Если Запрос.Выполнить (ТекстЗапроса)=1 **Тогда**

```
// выгрузим запрос в таблицу квартир
```

```
Запрос.Выгрузить (ТаблицаКвартир, , 0) ;
```

```
ТаблицаКвартир.Выгрузить (Квартиры) ;
```

КонецЕсли ;

КонецПроцедуры // Заполнить

Шаблон

Конструирование шаблона оперативного отчета ничем не отличается от конструирования шаблона встроенного в документ *отчета* (см. разд. "Встроенные отчеты" этой главы). Данный отчет интересен тем, что количество колонок, как и количество строк, переменные. Невозможно в шаблоне зафиксировать все присутствующие в отчете колонки. Динамическое формирование колонок и строк отчета производится средствами процедуры печати. В шаблоне отчета выделим отдельными секциями шаблон той строки (**ОТ**) и шаблон той колонки (**ВР**), которые будут повторяться при формировании отчета (рис. 4.81).

В шаблоне заданы переменные отчета:

- ИмяВР** — имя вида расчета (начисления);
- Квартира** — квартира, по которой ведется расчет;
- Запрос.СуммаПоВР** — сумма начисления по текущему виду расчета для текущей квартиры;
- ИтогоВР** — всего начислено по текущей квартире;
- СуммаПоВР** — сумма оплаты по текущей квартире;
- Итого** — конечное сальдо по текущей квартире;
- Итог** — начислено по текущему виду расчета по всем квартирам;
- ИтВсегоНач** — всего начислено;

- ИтВыплачено — всего оплачено;
- ИтСальдоКон — конечное сальдо.

	ЗаглВерт		ВР	ИтогВерт		
	1	2	3	4	5	6
1						
2	<Предприятие [Константа.НазваниеОрганизации]>					
3						
4	Расчетная ведомость					
5						
6	<за [ЖрнНачисления.ОписательПериода(ДатаАктуальности)]>					
7						
8	Номер п/п	Квартира	<ИмяВР>	Итого начислено	Выплачено через кассу/банк	Сальдо
9						
10	<сч>	<Квартира>	сс.СуммаПоВР>	<ИтогоВР>	<СуммаПоВР>	<Итого>
11	Итого по ведомости:		<Итого>	<ИтВсегоНач>	ИтВыплачено>	ИтСальдоКон>
12	Гл. бухгалтер					
13						
14						

Рис. 4.81. Диалоговое окно отчета Расчетная ведомость на вкладке Таблица

Процедура

Динамическое формирование колонок отчета реализуется посредством метода работы с таблицей `ПрисоединитьСекцию()`:

Процедура `Сформировать()`

Если `Квартиры.РазмерСписка() = 0` **Тогда**

`СпособВыборкиКвартир = 1;`

Иначе

`СпособВыборкиКвартир = 3;`

КонецЕсли;

`ЖрнНачисления = СоздатьОбъект("ЖурналРасчетов.Начисления");`

`НачПериода = ЖрнНачисления.НачалоПериодаПоДате(ДатаАктуальности);`

`КонПериода = ЖрнНачисления.КонецПериодаПоДате(ДатаАктуальности);`

`СписокКвартир = СоздатьОбъект("СписокЗначений");`

`Запрос = СоздатьОбъект("Запрос");`

`// Сформируем выборку информации в Запрос`

`ТекстЗапроса =`

`"/ / { ЗАПРОС (РасчВед)`

```

|Период с НачПериода по КонПериода;
|Квартира      = ЖурналРасчетов.Начисления.Объект;
|Дом           = ЖурналРасчетов.Начисления.Объект.Дом;
|Расчет        = ЖурналРасчетов.Начисления.ВидРасч;
|Результат     = ЖурналРасчетов.Начисления.Результат;
|ДатаНачала   = ЖурналРасчетов.Начисления.ДатаНачала;
|Функция ВсегоНачислено = Сумма (Результат) ;
|Функция СуммаПоВР = Сумма (Результат)  когда
                    (Расчет.ВходитВГруппу (ГруппаРасчетов.КвартПлата)=1
                    ) ;
|Функция Оплачено      = Сумма (Результат)  когда
                    (Расчет=ВидРасчета.Оплата) ;

```

```

|Группировка Квартира без групп;
|Группировка Расчет без групп;
|Условие ( (Результат<>0) ); "+

```

```

? (СпособВыборкиКвартир=3, "
|Условие (Квартира в СписокКвартир) ;",

```

```

? (СпособВыборкиКвартир=2, "
|Условие (Дом в Дом) ;", "" ) + "
|"//} }ЗАПРОС
;

```

```
// Если ошибка в запросе, то выход из процедуры
```

```
Если Запрос.Выполнить (ТекстЗапроса)=0 Тогда
```

```
    глСтатусСтрока (Контекст, "");
```

```
    Возврат;
```

```
КонецЕсли;
```

```
// Сформируем список видов расчетов (ВР) в том порядке, который нам
нужен в расчетной ведомости
```

```
СписокВР=СоздатьОбъект ("СписокЗначений");
```

```
ВременнаяТаблица = СоздатьОбъект ("ТаблицаЗначений");
```

```
ВременнаяТаблица1 = СоздатьОбъект ("ТаблицаЗначений");
```

```
Итоги = СоздатьОбъект ("ТаблицаЗначений");
```

```
Запрос.Выгрузить (ВременнаяТаблица, 0, 0);
```

```
ВременнаяТаблица.Свернуть ("Расчет", );
```

```
ВременнаяТаблица.Выгрузить (СписокВР, ,, "Расчет");
```

```
// Описание таблицы
```

```
Таблица = СоздатьОбъект ("Таблица");
```

```
Таблица.ИсходнаяТаблица ("РасчетнаяВедомость");
```

```
// Выведем стационарную часть шапки
```

```
Таблица.ВывестиСекцию ("ЗаглГориз | ЗаглВерт");
```

```
// Теперь пройдемся по списку ВР начислений и выведем первую часть
Размер= СписокВР.РазмерСписка ();
Для Сч=1 по Размер Цикл
    ИмяВР = СписокВР.ПолучитьЗначение (Сч);
    Если ИмяВР<>ВидРасчета.Оплата Тогда
// Нарастим шапку текущим видом расчета
    Таблица.ПрисоединитьСекцию ("ЗаглГориз|ВР");
    КонецЕсли;
КонецЦикла;
Если Размер > 0 тогда
// Присоединим секцию итогов в шапке
    Таблица.ПрисоединитьСекцию ("ЗаглГориз|ИтогВерт");
КонецЕсли;
// Запрос.Группировка ("Квартира");
Запрос.Выгрузить (ВременнаяТаблица1, 0, 0);
ВременнаяТаблица1.Свернуть ("Квартира", );
Запрос.Выгрузить (Итоги, 0, 2);
Итоги.Свернуть ("Расчет", "СуммаПоВР");
сч=0;
ВременнаяТаблица1.ВыбратьСтроки ();
Пока ВременнаяТаблица1.ПолучитьСтроку ()=1 цикл
    Квартира =ВременнаяТаблица1.Квартира;
    сч=сч+1;
// Выведем в строке секцию квартир
    Таблица.ВывестиСекцию ("ОТ|ЗаглВерт");
    Размер= СписокВР.РазмерСписка ();
    ИтогоВР = 0;
// Теперь пройдемся по списку ВР начислений в том порядке, как в шапке
    Для С=1 по Размер Цикл
        ИмяВР = СписокВР.ПолучитьЗначение (С);
// Найдем сумму начисления в запросе
        Если Запрос.Получить (Квартира,ИмяВР)=1 тогда
            СуммаПоВР = Запрос.СуммаПоВР;
        Иначе
            СуммаПоВР = 0;
        КонецЕсли;
// Оплату не выводим здесь
        Если ИмяВР<>ВидРасчета.Оплата тогда
            ИтогоВР=ИтогоВР+СуммаПоВР;
// Нарастим строку суммой по текущему виду расчета
    Таблица.ПрисоединитьСекцию ("ОТ|ВР");
    КонецЕсли;
КонецЦикла;
ИмяВР=ВидРасчета.Оплата;
```

```

// Найдем сумму оплаты
Если Запрос.Получить (Квартира,ИмяВР)=1 тогда
    СуммаПоВР = Запрос.СуммаПоВР;
Иначе
    СуммаПоВР = 0;
КонецЕсли;
Итого=СуммаПоВР-ИтогоВР;
// Нарастим строку оплатой и итогами
Таблица.ПрисоединитьСекцию ("ОТ|ИтогВерт" );
КонецЦикла ;
// Выведем стационарную часть итоговой строки
Таблица.ВывестиСекцию ("ИтогоПоВедомости|ЗаглВерт" );
Размер= СписокВР.РазмерСписка ();
ИтВсегоНач = 0;
// Сформируем итоговую строку
// Пройдемся по списку ВР начислений в том порядке, как в шапке
Для С=1 по Размер Цикл
    ИмяВР = СписокВР.ПолучитьЗначение (С) ;
    Итоги.ВыбратьСтроки ();
// Найдем итоги по текущему виду расчета
Пока Итоги.ПолучитьСтроку ()=1 цикл
    Если Итоги.Расчет= ИмяВР тогда
        Если ИмяВР=ВидРасчета.Оплата тогда
            ИтВыплачено = Итоги.СуммаПоВР;
        Иначе
            Итог = Итоги.СуммаПоВР;
            ИтВсегоНач = ИтВсегоНач + Итог;
// Нарастим итоговую строку итогом по текущему виду расчета
Таблица.ПрисоединитьСекцию ("ИтогоПоВедомости|ВР" );
КонецЕсли ;
Иначе
    Продолжить ;
КонецЕсли ;
КонецЦикла ;
КонецЦикла ;
ИтСальдоКон = ИтВыплачено - ИтВсегоНач;
// Нарастим итоговую строку оплатой и сальдо
Таблица.ПрисоединитьСекцию ("ИтогоПоВедомости|ИтогВерт" );
Таблица.ТолькоПросмотр (1) ;
Таблица.Опции (0, 0, 0, 0, "" );
Таблица.ПараметрыСтраницы (,,,,,,1,);
Таблица.Показать ("Расчетная ведомость за "+ТекстПериода) ;
КонецПроцедуры

```

Расчетная ведомость (Т-51) за Ноябрь 2003 г. *

Предприятие МУП "Жилзаказчик"

Расчетная ведомость

за Ноябрь 2003 г.

Номер пл/л	Квартира	КвартПлата	Горячая вода по норме	Обслуживание лифта	Хол. вода по водомеру	Итого начислено	Выплачено через кассу/банк	Сальдо
1	Дзержинского 120, к.60	96	99	99	0	294	0	-294
2	Дзержинского 129, к.60	256	118.8	118.8	300	793.6	1000	206.4
Итого по ведомости:		352	217.8	217.8	300	1087.6	0	-87.6

Рис. 4.82. Сформированный отчет "Расчетная ведомость"

В результате работы процедуры сформирован отчет "Расчетная ведомость" (рис. 4.82).

Внешние отчеты

К внешним отчетам относятся отчеты, которые хранятся в отдельных файлах, а не в конфигурации. По функциональным возможностям внешние отчеты ничем не отличаются от отчетов, определенных в конфигурации. Возможности же переносимости от конфигурации к конфигурации шире. Их можно запускать в любой конфигурации при условии, что объекты метаданных, используемые в отчете, были определены в данной конфигурации с необходимыми атрибутами и реквизитами. В Конфигураторе предусмотрена возможность сохранения внешних отчетов в отдельных файлах и встраивания внешних отчетов в конфигурацию. Эта функция позволяет отлаживать отчеты без записи изменений в конфигурации, т. е. без перезапуска программы.

Внимание!

Работа по отладке и запуску внешнего отчета не требует монопольного режима работы с конфигурацией! Это одно из самых важных преимуществ внешних отчетов.

Создадим внешний отчет "Ведомость должников". Для создания нового внешнего отчета необходимо в меню **Файл** выбрать пункт **Новый** или на панели инструментов нажать кнопку  или <Ctrl>+<N>) (рис. 4.83).

Данная функция меню предназначена для создания текстовых файлов, таблиц в формате mxl и внешних отчетов. Для выбора конкретного вида объекта предлагается диалоговое окно **Выбор вида документа** (рис. 4.84).

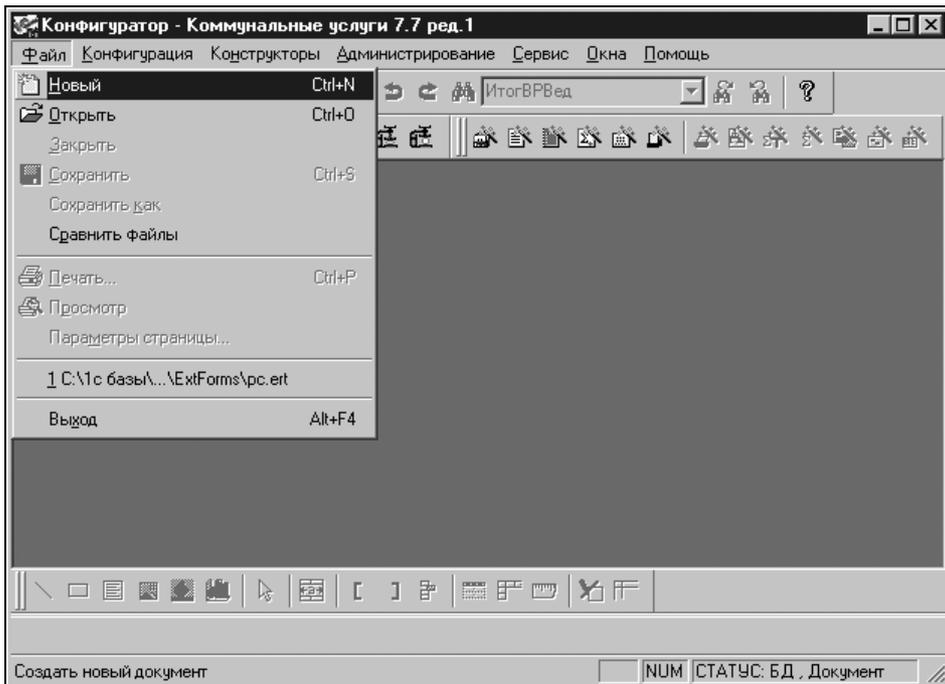


Рис. 4.83. Меню Конфигуратора Файл

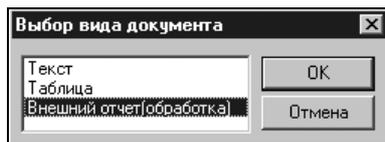


Рис. 4.84. Диалоговое окно Выбор вида документа

После выбора пункта **Внешний отчет (обработка)** будет предложен стандартный пустой отчет, как и в случае обычного отчета (рис. 4.85). Следует обратить особое внимание на то, что основные реквизиты объекта метаданных для внешних отчетов не задаются. Имя файла отчета задается при записи отчета.

Запись отчета производится несколькими способами:

- по запросу при закрытии диалогового окна;
- при выборе пункта **Сохранить** в меню **Файл** (кнопка на панели инструментов  или <Ctrl>+<S>) (рис. 4.83);
- при выборе пункта меню **Сохранить как** в меню **Файл** (см. рис. 4.83).



Рис. 4.85. Диалоговое окно внешнего отчета на вкладке **Диалог**

При первом закрытии диалогового окна отчета на экран выдается запрос на сохранение отчета (рис. 4.86).

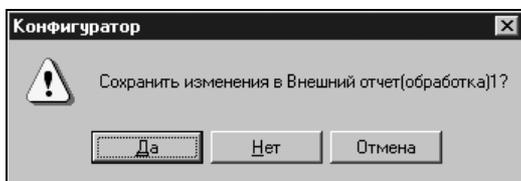


Рис. 4.86. Диалоговое окно внешнего отчета на вкладке **Диалог**

При нажатии на кнопку **Да** в предложенном диалоговом окне **Сохранить как...** необходимо задать имя файла отчета и путь к нему, затем нажать кнопку **Сохранить**. По умолчанию в диалоговом окне предлагается каталог Ext Forms текущей базы данных (рис. 4.87).

В дальнейшем после любого изменения, внесенного в отчет, при закрытии будет выдаваться аналогичный запрос на сохранение, только с уже заданным именем файла и путем к нему.

Для того чтобы сохранить файл отчета под другим именем, необходимо выбрать пункт меню **Сохранить как** в меню **Файл** и задать другое имя и путь, если это необходимо.

Открыть файл отчета на редактирование можно с помощью пункта **Открыть** меню **Файл** (см. рис. 4.83). Открыть файл отчета можно также выбором из списка последних используемых отчетов в меню **Файл** (рис. 4.88).

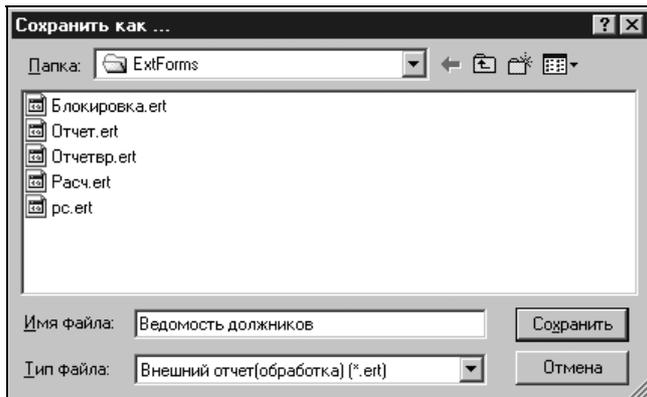


Рис. 4.87. Диалоговое окно Сохранить как..

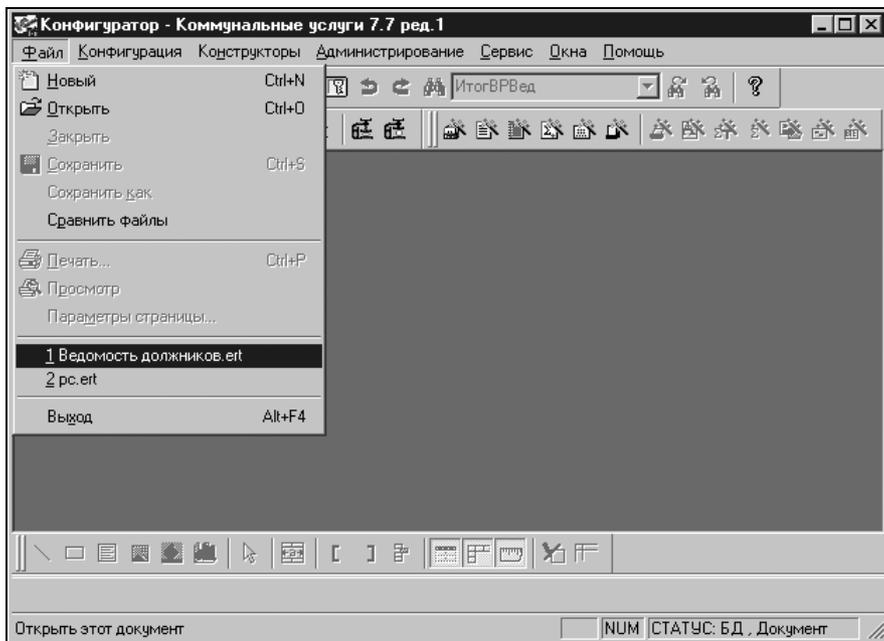


Рис. 4.88. Меню Файл Конфигуратора со списком отчетов

Отчет "Ведомость должников" информационно похож на отчет "Расчетная ведомость", только необходимо в процедуре добавить проверку на отрицательное сальдо и изменить шаблон. Для отладки нового отчета во внешнем файле необходимо выполнить следующие действия:

1. Установить указатель на отчет, который необходимо сохранить в файле.
2. В меню **Действия** выбрать пункт **Внешний отчет**, подпункт **Сохранить как внешний отчет (обработку)...** (рис. 4.89) или в контекстном меню выбрать пункт **Сохранить как внешний отчет (обработку)....**
3. В диалоговом окне **Сохранить как...** задать имя файла сохраняемого отчета и путь к нему (см. рис. 4.87).

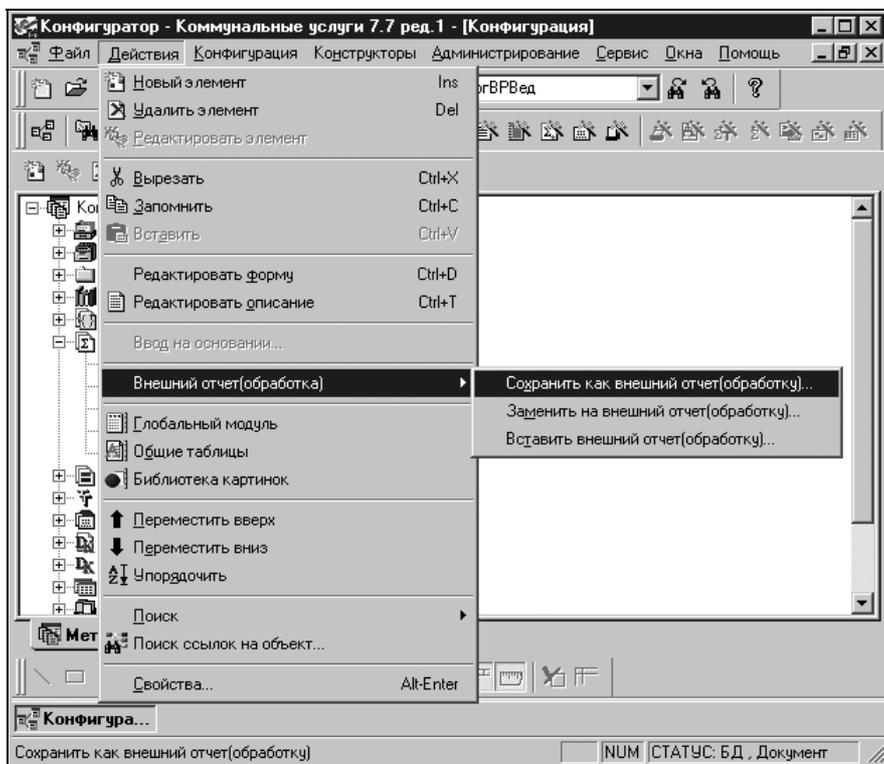


Рис. 4.89. Меню Действия Конфигуратора для отчетов

Сохраним таким образом нашу расчетную ведомость в файл с именем **Ведомость должников.ert**. Откроем данный файл с помощью пункта **Открыть** меню **Файл** и откорректируем его согласно требованиям к ведомости должников. В форме отчета изменим только заголовок на **Ведомость должников** (рис. 4.90).

В ведомости должников нет детализации по видам расчетов, следовательно, разбиение на вертикальные секции не нужно. Оставим в ведомости только колонки **Квартира** и **Сальдо** и исключим их из секций (рис. 4.91).

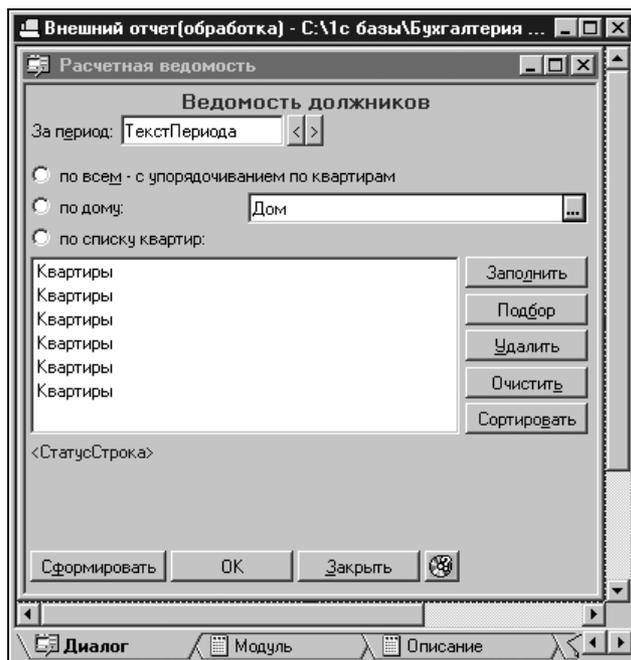


Рис. 4.90. Диалоговое окно отчета **Ведомость должников** на вкладке **Диалог**

	1	2	3	4
Шапка	1			
	2	<Предприятие [Константа.НазваниеОрганизации]>		
	3			
	4	Ведомость должников		
	5	<за [ЖрнНачисления.ОписательПериода(ДатаАктуальности)]>		
	6			
	7			
	8	Номер п/п	Квартира	Сальдо
	9			
Строка	10	<сч> <Квартира>		<Итого>
ИтогоПоВе	11	Итого по ведомости:		<ИтогоВед>
	12	Гл. бухгалтер		
	13			
	14			
	15			

Рис. 4.91. Диалоговое окно отчета **Ведомость должников** на вкладке **ВедомостьДолжников**

Процедура формирования отчета значительно упростится, но отлаженный в расчетной ведомости запрос не изменится:

Процедура Сформировать ()

Если Квартиры.РазмерСписка () = 0 **тогда**

СпособВыборкиКвартир = 1;

Иначе

СпособВыборкиКвартир = 3;

КонецЕсли;

ЖрнНачисления = СоздатьОбъект ("ЖурналРасчетов.Начисления");

НачПериода = ЖрнНачисления.НачалоПериодаПоДате (ДатаАктуальности);

КонПериода = ЖрнНачисления.КонецПериодаПоДате (ДатаАктуальности);

СписокКвартир = СоздатьОбъект ("СписокЗначений");

Запрос = СоздатьОбъект ("Запрос");

// Сформируем выборку информации в Запрос

ТекстЗапроса =

"//{ {ЗАПРОС (РасчВед)

|Период с НачПериода по КонПериода;

|Квартира = ЖурналРасчетов.Начисления.Объект;

|Дом = ЖурналРасчетов.Начисления.Объект.Дом;

|Расчет = ЖурналРасчетов.Начисления.ВидРасч;

|Результат = ЖурналРасчетов.Начисления.Результат;

|ДатаНачала = ЖурналРасчетов.Начисления.ДатаНачала;

|Функция ВсегоНачислено = Сумма (Результат);

|Функция СуммаПоВР = Сумма (Результат) когда
(Расчет.ВходитВГруппу (ГруппаРасчетов.КвартПлата)=1);

|Функция Оплачено = Сумма (Результат) когда
(Расчет=ВидРасчета.Оплата);

|Группировка Квартира без групп;

|Группировка Расчет без групп;

|Условие ((Результат<>0));"+

? (СпособВыборкиКвартир=3, "

|Условие (Квартира в СписокКвартир);",

? (СпособВыборкиКвартир=2, "

|Условие (Дом в Дом);", "")+"

| " // } ЗАПРОС

;

// Если ошибка в запросе, то выход из процедуры

Если Запрос.Выполнить (ТекстЗапроса)=0 **Тогда**

глСтатусСтрока (Контекст, "");

Возврат;

КонецЕсли;

```

ВременнаяТаблица = СоздатьОбъект ("ТаблицаЗначений");
// Описание таблицы
Таблица = СоздатьОбъект ("Таблица");
Таблица.ИсходнаяТаблица ("ВедомостьДолжников");
Таблица.ВывестиСекцию ("Шапка");
Запрос.Выгрузить (ВременнаяТаблица, 0, 0);
ВременнаяТаблица.Свернуть ("Квартира", "СуммаПоВР, Оплачено");
сч=0;
ИтогоВед= 0;
ВременнаяТаблица.ВыбратьСтроки();
Пока ВременнаяТаблица.ПолучитьСтроку()=1 цикл
    Квартира =ВременнаяТаблица.Квартира;
    сч=сч+1;
    Итого = ВременнаяТаблица.Оплачено - ВременнаяТаблица.СуммаПоВР;
    Если Итого < -Константа.ПределСуммыДолга.Получить (КонПериода) тогда
        ИтогоВед = ИтогоВед+Итого;
        Таблица.ВывестиСекцию ("Строка");
    КонецЕсли;
КонецЦикла;
Таблица.ВывестиСекцию ("ИтогоПоВедомости");
Таблица.ТолькоПросмотр (1);
Таблица.Опции (0, 0, 0, 0, "");
Таблица.ПараметрыСтраницы (,,,,,, 1);
Таблица.Показать ("Ведомость должников за "+ТекстПериода);

```

КонецПроцедуры

В результате работы процедуры сформирован отчет "Ведомость должников" (рис. 4.92).

Номер плп	Квартира	Сальдо
1	Дзержинского 120, к.60	-294
Итого по ведомости:		-294

Рис. 4.92. Сформированный отчет "Ведомость должников"

После отладки необходимо встроить отчет в конфигурацию. Для включения внешнего отчета в конфигурацию необходимо выполнить следующие действия:

1. Установить указатель на пункт **Отчеты** дерева метаданных.
2. В меню **Действия** выбрать пункт **Внешний отчет**, подпункт **Вставить внешний отчет (обработку)...** (см. рис. 4.89) или в контекстном меню выбрать пункт **Вставить внешний отчет (обработку)...**
3. В диалоговом окне **Открыть** выбрать файл отчета.
4. В меню **Действия** выбрать пункт **Свойства**.
5. В диалоговом окне **Свойства отчета** задать основные реквизиты отчета:
 - Идентификатор;
 - Синоним;
 - Комментарий.

Внимание!

Прежде чем приступить к вставке внешнего отчета, убедитесь в том, что вставляемый внешний отчет не открыт в Конфигураторе. В противном случае вставки отчета не произойдет!

Регламентированные отчеты

В конфигурации "1С:Предприятие. Бухгалтерский учет" в составе отчетов присутствует отчет "Регламентированные отчеты". Данный отчет предназначен для загрузки и формирования регламентированной бухгалтерской и налоговой отчетности (рис. 4.93).

Комплект отчетов, которые присутствуют в меню **Регламентированные отчеты**, поставляется ежеквартально фирмой "1С". В последние релизы конфигурации добавлена возможность выгрузки отчетов в налоговые органы и различные фонды. С точки зрения конструирования интересен механизм доступа к формам и шаблонам отчетов из комплекта. Регламентированные отчеты загружаются в каталог ExtForms текущей базы данных, в подкаталог с именем RpNNqM.grp, где NN — две последние цифры года, M — номер квартала. Например, отчетность за IV квартал 2004 года будет загружена в каталог ExtForms\ Rp04q4.grp. Структура меню и связь пунктов меню с файлами каталога задается в файле RPLIST.txt каталога отчетов. Структура файла RPLIST.txt следующая.

- Наименование периода в списке **ГруппыОтчетов**.
- Перевод строки.

- Имя файла отчета в каталоге отчетов.
- Начало группы в меню.
- Имя пункта меню в таблице **ТаблОтчетов**.
- Описание текущего отчета в тексте **Описание**.
- Перевод строки.



Рис. 4.93. Диалоговое окно отчета "Регламентированные отчеты"

Следует обратить особое внимание на количество символов (;) в начале строки. Количество символов (;), следующих подряд, означает уровень вложенности группы. Рассмотрим фрагмент файла RPLIST.txt:

```

Отчетность за IV квартал 2004 года
info.ert;Информация для пользователей;Дополнительная информация
о комплекте отчетности
lsbinstr.txt;Инструкция по составлению отчетности
RpUnload.ert;Выгрузка данных отчетности (формат версии 3.00);Подготовка
представления отчетности в электронном виде в формате 3.00

```

RpUnld20.ert;Выгрузка данных отчетности (формат версии 2.01/2.00);Подготовка представления отчетности в электронном виде в формате 2.01 и 2.00

;Бухгалтерская отчетность

frm1.ert;Бухгалтерский баланс (форма №1);Утвержден приказом Минфина России от 22.07.2003 N 67н

frm2.ert;Отчет о прибылях и убытках (форма №2);Утвержден приказом Минфина России от 22.07.2003 N 67н

frm3.ert;Отчет об изменениях капитала (форма №3);Утвержден приказом Минфина России от 22.07.2003 N 67н

frm4.ert;Отчет о движении денежных средств (форма №4);Утвержден приказом Минфина России от 22.07.2003 N 67н

frm5.ert;Приложение к балансу (форма №5);Утверждено приказом Минфина России от 22.07.2003 N 67н

frm6.ert;Целевое использование полученных средств (форма №6);Утвержден приказом Минфина России от 22.07.2003 N 67н

aktv.ert;Расчет чистых активов акционерного общества;Утвержден приказом Минфина России и ФКЦБ России от 29 января 2003 г. №10н/03-б/пз

;Отчеты в фонды

FSS_nspz.ert;Использование сумм страховых взносов (НСИПЗ);Приложение N 1 к Постановлению Фонда социального страхования Российской Федерации от 27.03.2003 N 33

stpkvot.ert;Форма 1-КВОТИРОВАНИЕ (Москва);Введена Комитетом общественных связей города Москвы в соответствии с постановлением Правительства Москвы от 04.03.03г. № 125-ПП

;Страховые взносы в ФСС по временной нетрудоспособности (форма-4а ФСС);Утверждена постановлением Фонда социального страхования Российской Федерации от 25.04.2003 №46

fss4a_01.ert;Форма-4а ФСС: титульный лист

fss4a_02.ert;Форма-4а ФСС - Раздел I. Таблицы 1,2

fss4a_03.ert;Форма-4а ФСС - Раздел II. Таблицы 3,4,5

fss4a_04.ert;Форма-4а ФСС - Раздел II. Таблица 6

;Расчетная ведомость по средствам ФСС РФ (форма 4-ФСС);Утверждена постановлением Фонда социального страхования Российской Федерации от 01.12.2003 № 130

fss4_01.ert;Ведомость в ФСС: Титульный лист

fss4_02.ert;Ведомость в ФСС: Раздел I. Таблицы 1, 2

fss4_03.ert;Ведомость в ФСС: Раздел I. Таблица 3

fss4_04.ert;Ведомость в ФСС: Раздел II. Таблицы 4, 5

fss4_05.ert;Ведомость в ФСС: Раздел II. Таблица 6

fss4_06.ert;Ведомость в ФСС: Раздел II. Таблица 7

fss4_07.ert;Ведомость в ФСС: Раздел II. Таблица 8

fss4_08.ert;Ведомость в ФСС: Раздел III. Таблицы 9, 10

fss4_09.ert;Ведомость в ФСС: Раздел III. Таблицы 11, 12

Планы счетов

План счетов — это, по сути, справочник. Ввиду того, что из базы данных любой конфигурации производится выгрузка проводок в конфигурацию "Бухгалтерский учет", план счетов присутствует почти в любой конфигурации. Главным его отличием от других справочников является количество базовых реквизитов — атрибутов справочника. Кроме кода и наименования, как в других справочниках, в плане счетов присутствуют такие базовые реквизиты, как различные признаки счета и несколько уровней субконто.

Поскольку план счетов жестко не регламентирован правительством, а всего лишь рекомендован, в конфигурации предусмотрена возможность работы с несколькими планами счетов. Для вызова окна редактирования **Планы Счетов** дважды щелкните левой кнопкой мыши на фразе **Планы Счетов** в дереве метаданных. Окно редактирования **Планы Счетов** служит для редактирования свойств планов счетов как объектов метаданных. Фактически, в окне редактирования **Планы Счетов** задаются параметры бухгалтерского учета в системе 1С:Предприятие (рис. 4.94).

Поле **Максимальная длина кода счета** задает максимальную длину кода счета, который может быть указан в плане счетов. Код счета в общем случае представляет собой символьную строку вида:

<Код счета>.<Код субсчета>.<Код субсчета>...

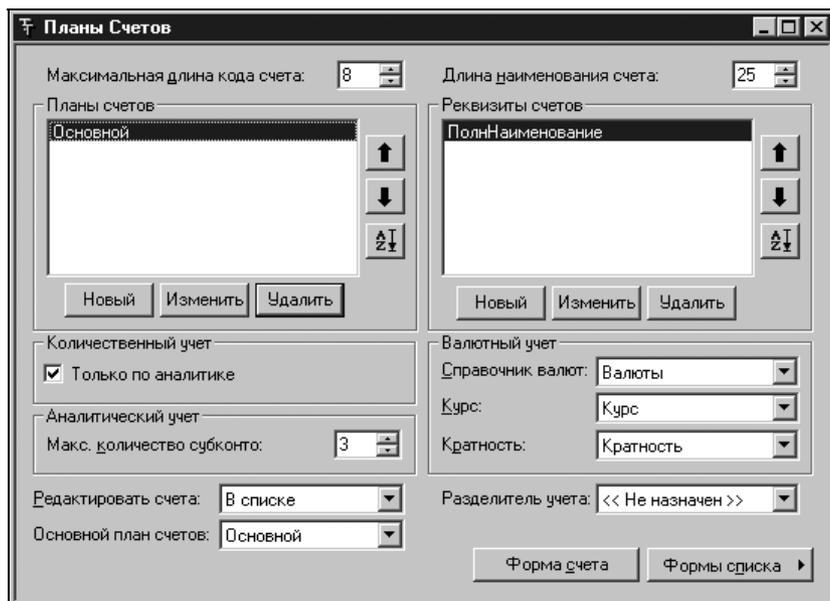


Рис. 4.94. Диалоговое окно Планы Счетов

Общая длина кода счета в системе 1С:Предприятие ограничена 255 символами. В это значение входят: длина кода счета первого порядка, длины кодов счетов всех нижележащих порядков и разделители номеров счетов (точка).

Поле **Длина наименования счета** задает максимальную длину наименования счета.

Управляющие элементы группы **Планы счетов** предназначены для создания, редактирования и удаления планов счетов. Использование этих управляющих элементов в основном совпадает с общими приемами редактирования списков объектов метаданных. Единственное отличие: при нажатии кнопки **Изменить** будет вызвано окно редактирования **План счетов** для редактирования свойств плана счетов и редактирования бухгалтерских счетов.

Управляющие элементы группы **Реквизиты счетов** предназначены для создания, редактирования и удаления дополнительных реквизитов счетов. Использование этих управляющих элементов в основном совпадает с общими приемами редактирования списков реквизитов обычных справочников (см. разд. "Справочники" данной главы).

Признак **Количественный учет** — **только по аналитике** задает способ ведения количественного учета. На практике количественный учет не имеет смысла без аналитического учета: например, нельзя вести количественный учет материалов "вообще" — обычно учитывают количество конкретного материала. Однако если предполагается вести аналитический учет с использованием субсчетов, а не субконто, может потребоваться указать ведение количественного учета на таком аналитическом субсчете. Чтобы предоставить пользователю такую возможность, служит опция **Количественный учет** — **только по аналитике**. Если эта опция выключена, то при редактировании планов счетов пользователь может устанавливать признак ведения количественного учета для тех счетов, на которых не ведется аналитический учет по субконто.

Поле **Основной план счетов** задает план счетов "по умолчанию" для конструкций встроенного языка, в которых можно указывать конкретный план счетов. Основной план счетов выбирается из списка планов счетов, существующих в конфигурации.

Поле **Максимальное количество субконто** задает максимальное количество видов субконто, которое может быть "прикреплено" к одному счету или субсчету. Может принимать числовое значение от 0 до 5.

Управляющие элементы, объединенные в группу **Валютный учет**, служат для настройки справочника валют. В поле **Справочник валют** необходимо выбрать идентификатор одного из существующих в конфигурации справочников. Этот справочник будет использоваться компонентой "Бухгалтерский

учет" в качестве справочника валют. В поле **Курс** выдается список числовых реквизитов справочника, выбранного в поле **Справочник валют**. Из этого списка необходимо выбрать идентификатор реквизита, в котором будут храниться курсы используемых валют. Если у справочника валют нет ни одного числового реквизита, такой реквизит необходимо создать. При вводе проводок значение из этого реквизита будет использоваться для расчета суммы в рублях, если указана валютная сумма проводки. Поле **Кратность** позволяет указать, в каком реквизите справочника валют будет храниться кратность валюты. Так же, как и для курса валюты, это должен быть реквизит типа "число". Если такой реквизит в поле **Кратность** указан, то при работе с системой 1С:Предприятие в этом реквизите необходимо поставить коэффициенты для всех используемых валют. Для валют, не имеющих кратности, следует поставить 1. Вычисление рублевой суммы проводки по валютной сумме будет выполняться системой с использованием кратности.

Компонента "Бухгалтерский учет" системы 1С:Предприятие позволяет вести в одной информационной базе бухгалтерский учет одновременно для нескольких предприятий. Для реализации такой возможности в объекте метаданных *Проводка* необходимо предусмотреть дополнительный реквизит (лучше, если этот реквизит будет иметь тип значения "Справочник"). При работе с системой 1С:Предприятие в этот реквизит необходимо вводить то или иное значение, которое позволит разделять проводки, принадлежащие разным предприятиям. Идентификатор этого реквизита необходимо выбрать в поле **Разделитель учета**. Система 1С:Предприятие автоматически организует систему хранения бухгалтерских итогов, разделяя их по значению указанного реквизита. С помощью встроенного языка информацию из итогов можно извлекать и по конкретному предприятию, и в целом — по всем предприятиям.

Поле **Редактировать счета** устанавливает способ редактирования конкретного счета в плане счетов выбором из списка:

- в списке** — редактирование кода, наименования и прочих характеристик счета будет проводиться непосредственно в ячейках плана счетов;
- в диалоге** — для редактирования счета будет вызываться специально разработанный диалог, куда будет помещаться информация из текущей строки плана счетов;
- обоими способами** — выбор конкретного способа редактирования будет производиться конечным пользователем при работе с системой 1С:Предприятие.

Нажатие кнопки **Форма счета** вызывает редактор форм для редактирования формы счета, которая будет использоваться, если в поле **Редактировать счета** выбрана установка **в диалоге**.

Нажатие кнопки **Формы списка** вызывает диалог **Формы списков** или редактор форм для редактирования уже существующих форм плана счетов.

Редактирование плана счетов

Редактирование плана счетов может выполняться как в Конфигураторе, так и при работе с системой 1С:Предприятие. При редактировании плана счетов в Конфигураторе доступны все операции в пределах заданных для планов счетов свойств:

- ввод новых счетов и субсчетов;
- удаление счетов и субсчетов;
- включение валютного и количественного учета на счете (субсчете) путем задания соответствующих признаков;
- редактирование признака "**Активный** — **Пассивный**";
- настройка аналитического учета.

При редактировании плана счетов в сеансе работы с системой 1С:Предприятие пользователь не может редактировать счета и субсчета, введенные в план счетов в Конфигураторе, — ему доступно только редактирование дополнительных реквизитов. Однако пользователь может вводить новые счета и субсчета и уже для них устанавливать признаки ведения валютного и количественного учета и другие характеристики, о которых говорилось ранее. Разделение места редактирования плана счетов позволяет, с одной стороны, разрабатывать конфигурацию, ориентированную на заранее известную структуру счетов и субсчетов; с другой стороны, дает возможность пользователю в своих целях добавлять в планы счетов новые счета и субсчета и использовать их для ввода информации и получения отчетных документов.

Внимание!

Такое разделение пугает разработчика от удаления или изменения тех свойств счетов, которые он использует в модулях проведения документов. Не рекомендуется использовать в модулях проведения документов и других обработках те счета, которые не были заданы в Конфигураторе.

Каждый план счетов обладает собственным набором свойств. Для редактирования этих свойств можно использовать палитру свойств или окно редактирования **План счетов**, который можно вызвать двойным щелчком мыши на идентификаторе плана счетов в дереве метаданных (рис. 4.95).

Шаблон кода представляет собой символьную строку, состоящую из символов (#) и (.). Шаблон кода схематически иллюстрирует структуру полного кода счета. Фактически шаблон кода задает количество уровней субсчетов и косвенно — максимальное количество субсчетов, которое может быть открыто к счету.

План счетов Основной

Идентификатор: Основной Синоним: Основной план

Комментарий: Основной

Шаблон кода: ###.###

Код	Наименование	В...	К...	З...	А...	Субконто1	Субконто2	Субконто3
Уг 00	Вспомогательный				АП			
Уг 01	Основные средства				А			
Уг 01.1	ОС в организации				А			
Уг 01.2	Выбытие ОС				А			
Уг 02	Амортизация ОС				П			
Уг 02.1	Аморт. ОС, уч. на сч.01.1				П			
Уг 02.2	Аморт. ОС, уч. на сч.03				П			
Уг 03	Доходные вложения в МЦ				А			
Уг 03.1	МЦ в организации				А			
Уг 03.2	МЦ, перед. во врем.влад.				А			
Уг 03.3	МЦ, перед. во врем.польз.				А			
Уг 03.4	Прочие доходные вложения				А			
Уг 03.5	Выбытие МЦ				А			
Уг 04	Нематериальные активы				А			
Уг 04.1	Нематериальные активы				А			
Уг 04.2	Расходы на НИОКР				А			
Уг 05	Амортизация НМА				П			
Уг 07	Оборудование к установке			+	А			
Уг 08	Влож. во внеоборотн. активы				А			

Рис. 4.95. Диалоговое окно План счетов Основной

Например, символьная строка вида

##.#.###

означает, что код счета первого уровня может состоять максимум из двух символов. К счету могут быть открыты 2 уровня субсчетов. Код субсчета первого уровня содержит 1 символ, код субсчета второго уровня — до 3-х символов. В системе 1С:Предприятие количество уровней вложенности субсчетов в плане счетов не ограничивается. В качестве "ограничителя" выступает величина реквизита **Максимальная длина кода счета** в окне редактирования **Планы счетов** (см. рис. 4.94). В пределах указанной длины можно задавать произвольное количество уровней вложенности, комбинируя символы (#) и (.) в шаблоне счета. **Шаблон счета** задается для каждого конкретного плана отдельно. Иерархия субсчетов строится системой 1С:Предприятие автоматически, основываясь на шаблоне счета. Шаблон кода для плана счетов может быть не задан. В этом случае допускается ввод кодов счетов в произвольном формате, с использованием букв, цифр и символов точки. Исходя из указанного кода счета, система 1С:Предприятие автоматически введет счета вышестоящих уровней.

План счетов в окне редактирования **План счетов** представляет собой таблицу, содержащую список счетов и субсчетов. Эта таблица имеет набор колонок для ввода параметров счета. Количество колонок для указания видов субконто может, в общем случае, быть различным и задается установкой реквизита **Макс. количество субконто** в окне редактирования **Планы счетов** (см. рис. 4.94).

В крайней левой колонке плана счетов различными значками обозначается тип текущей строки.

Для просмотра плана счетов можно использовать клавиши управления курсором или линейки прокрутки. Редактирование плана счетов заключается в добавлении новых счетов и субсчетов и корректировании их свойств.

Ввод нового счета (субсчета)

Для ввода нового счета (субсчета) в план счетов следует нажать клавишу <Ins> или выбрать пункт **Новая строка** в контекстном меню плана счетов. В появившейся новой строке необходимо заполнить колонку **Код** и нажать клавишу <Enter>. Колонка **Код** будет содержать маску кода счета в соответствии с шаблоном, указанным в поле **Шаблон кода**. В колонку можно сразу ввести код субсчета (кода второго порядка), тогда после завершения ввода нового субсчета Конфигуратор автоматически введет строки, соответствующие субсчетам вышестоящих уровней. Если для редактируемого плана счетов не задан шаблон счета, то в колонку **код** можно ввести произвольный код счета, комбинируя буквы, цифры и символы точки в пределах разрешенной длины кода счета. Конфигуратор также автоматически введет субсчета вышестоящих уровней.

После ввода кода счета и нажатия клавиши <Enter> курсор автоматически переместится в колонку **Наименование**. Наименование счета представляет собой произвольную строку символов. Ее длина ограничивается величиной, заданной в реквизите **Длина наименования счета** окна редактирования **Планы счетов** (см. рис. 4.94). После редактирования наименования счета следует нажать клавишу <Enter>.

Если в план счетов вводится новый счет не последнего уровня, будет выдан запрос о существовании субсчета данного счета (рис. 4.96).

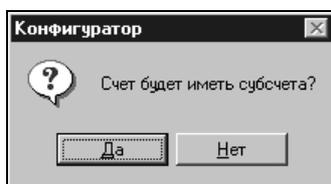


Рис. 4.96. Диалоговое окно запроса

Поясним смысл этого запроса. Счета в **Плане счетов** делятся на 2 вида — группы счетов и собственно счета. Различие между ними простое: группы счетов имеют субсчета, а "собственно счета" не имеют субсчетов.

Если счет будет иметь субсчета (т. е. вводится группа счетов), то такой счет нельзя будет указать в качестве корреспондирующего при вводе проводок —

необходимо будет указывать субсчета этого счета. И наоборот, если счет не будет иметь субсчетов, его можно указывать при вводе проводок. Однако если для счета, для которого было указано отсутствие субсчетов, все-таки введен субсчет, то система 1С:Предприятие выполнит следующее. В план счетов для этого счета автоматически будет введен субсчет с условным кодом 0. Во всех проводках, которые были сделаны по этому счету, этот счет также автоматически будет заменен субсчетом с кодом 0. В дальнейшем код субсчета 0 можно будет изменить на другой, в этом случае будет также проведено соответствующее изменение номеров субсчетов в проводках.

Для редактирования счета необходимо установить курсор в колонку плана счетов, значение которой нужно отредактировать, и нажать клавишу <Enter> или дважды щелкнуть на ней мышью. Ячейка, на которой находится курсор, переключится в режим редактирования; в ней появится курсор в виде мигающей вертикальной полоски. При редактировании ячейки можно перемещаться в другие колонки текущей строки, используя клавиши <Tab> и <Shift>+<Tab>. Для окончания редактирования ячейки следует нажать клавишу <Enter>. Чтобы отказаться от внесения изменений в значение ячейки, следует нажать клавишу <Esc>. Все изменения, выполненные в редактируемой ячейке, будут отменены.

Помимо кода и наименования, для любого счета (субсчета) в плане счетов можно задать ряд свойств:

- признак ведения на счете валютного учета;
- признак ведения на счете количественного учета;
- признак забалансового счета;
- признак **"Активный — Пассивный"**.

Кроме этого, для любого счета в плане счетов можно включить ведение аналитического учета по субконто: для этого к счету могут быть "прикреплены" виды субконто.

Путем установки свойств счета фактически настраивается система учета в целом: включается ведение валютного и количественного учета, настраиваются "разрезы", по которым будет вестись аналитический учет.

Редактирование перечисленных свойств выполняется в соответствующих колонках плана счетов. Для включения признака на счете (субсчете) выполните следующие действия:

1. Переведите ячейку в режим редактирования, нажав клавишу <Enter>.
2. Нажмите клавишу <F4> или кнопку в ячейке. Включенный признак обозначается знаком (+).
3. Еще раз нажмите <Enter> для выхода из режима редактирования ячейки.

Для снятия признака необходимо повторно выполнить описанные действия.

Внимание!

Свойства, задаваемые для счета, автоматически задаются и для всех субсчетов этого счета. Если признак устанавливается для счета, имеющего субсчет, то признак автоматически устанавливается также для всех субсчетов. Если к счету, имеющему признак, открывается новый субсчет, для субсчета автоматически ставится этот признак. Если у счета не установлен признак, то признаки у субсчетов можно устанавливать произвольно.

Колонка **Вал.** ("Валютный") служит для включения признака ведения валютного учета на выбранном счете (субсчете). Эта колонка появляется в окне редактирования **План счетов**, если заполнены реквизиты в группе **Валютный учет** окна редактирования **Планы счетов** (см. рис. 4.94): как минимум, должно быть указано, какой справочник будет использоваться в качестве справочника валют.

Внимание!

Если для счета (субсчета) включен признак валютного учета, необходимо обеспечить возможность ввода необходимой информации при работе с системой 1С:Предприятие: следует вставить соответствующие колонки в формы журнала проводок.

Колонка **Кол.** ("Количественный") служит для включения признака ведения количественного учета на выбранном счете (субсчете).

Колонка **Заб.** ("Забалансовый") служит для включения признака забалансового счета. Для забалансовых счетов система 1С:Предприятие не позволяет корреспондировать их в проводках с балансовыми счетами.

Колонка **Акт.** служит для установки признака "**Активный — Пассивный**". Для установки этого признака выполните следующие действия:

1. Переведите ячейку в режим редактирования, нажав клавишу <Enter>.
2. В ячейке нажмите клавишу <F4> или кнопку, расположенную справа от ячейки.
3. Из выпадающего списка выберите одно из значений:
 - **А** ("Активный");
 - **П** ("Пассивный");
 - **АП** ("Активно-пассивный").
4. Еще раз нажмите <Enter> для выхода из режима редактирования ячейки.

Признак "**Активный — Пассивный**" имеет следующий смысл. Счет (субсчет), обозначенный в плане счетов как "Активный", не должен иметь кредитового остатка.

Для такого счета:

- в случае дебетового остатка на счете (это нормальная ситуация) методы встроенного языка `СНД()` и `СКД()` будут возвращать величину остатка со знаком (+);
- в случае кредитового остатка на счете методы `СНД()` и `СКД()` будут возвращать величину остатка со знаком (-). Такая ситуация должна рассматриваться как недопустимая;
- методы `СНК()` и `СКК()` всегда будут возвращать 0.

Счет (субсчет), обозначенный в плане счетов как "Пассивный", не должен иметь дебетового остатка. Для такого счета:

- в случае кредитового остатка на счете (это нормальная ситуация) методы встроенного языка `СНК()` и `СКК()` будут возвращать величину остатка со знаком (+);
- в случае дебетового остатка на счете методы `СНК()` и `СКК()` будут возвращать величину остатка со знаком (-). Такая ситуация должна рассматриваться как недопустимая;
- методы `СНД()` и `СКД()` всегда будут возвращать 0.

Счет (субсчет), обозначенный в плане счетов как "Активно-пассивный", может иметь как дебетовый, так и кредитовый остаток. Для такого счета:

- в случае дебетового остатка на счете методы встроенного языка `СНД()` и `СКД()` будут возвращать величину остатка со знаком (+), а методы `СНК()` и `СКК()` будут возвращать 0;
- в случае кредитового остатка на счете методы `СНК()` и `СКК()` будут возвращать величину остатка со знаком (+), а методы `СНД()` и `СКД()` будут возвращать 0.

Для получения развернутого сальдо по счетам, помеченным в плане счетов как активно-пассивные, следует организовать на таких счетах ведение аналитического учета. Аналитический учет может быть организован по субсчетам счета — в этом случае к счету должно быть открыто достаточное количество субсчетов, или по субконто.

Колонки **Субконто1...Субконто5** служат для указания наименований видов субконто, прикрепленных к счету (субсчету). Количество колонок зависит от установки реквизита **Макс. количество субконто** в окне редактирования **Планы счетов** (см. рис. 4.94).

Для выбора вида субконто необходимо выполнить следующие действия:

1. Поместите курсор в одну из колонок **Субконто** плана счетов.
2. Переведите ячейку в режим редактирования, нажав клавишу <Enter> или дважды щелкнув в ячейке мышью.

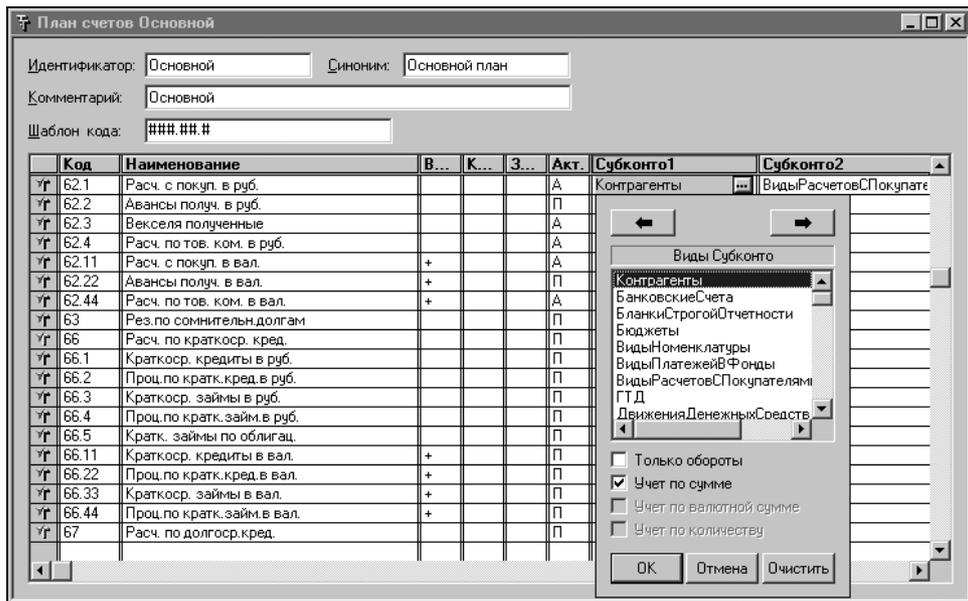


Рис. 4.97. Диалоговое окно **План счетов** со списком для выбора вида субконто

3. В открывшемся списке с наименованиями видов субконто выберите нужный вид субконто. Если список видов субконто не открылся автоматически — нажмите клавишу <F4> или кнопку в ячейке (рис. 4.97).
4. Для ввода наименования вида субконто в ячейку плана счетов дважды щелкните наименование левой кнопкой мыши или нажмите кнопку **ОК**.
5. Для отказа от выбора вида субконто необходимо нажать кнопку **Отмена** в списке видов субконто. Ячейка сохранит то состояние, которое было до редактирования. Чтобы очистить ячейку и таким образом отказаться от ведения аналитического учета по этом виду субконто, следует нажать кнопку **Очистить**.

Кнопки  и  позволяют перемещать наименование вида субконто из одной колонки **Субконто** в другую. Порядок наименований видов субконто в плане счетов влияет на последовательность выбора субконто при вводе проводок в системе 1С:Предприятие, а также на скорость расчета итогов в разных разрезах. Включение (или отключение) аналитического учета по счету может выполняться в любой момент эксплуатации системы 1С:Предприятие.

Внимание!

Следует обратить особое внимание на то, что, если к счету, по которому не велось аналитического учета, прикрепляется один (или несколько) вид субконто,

то все проводки по этому счету, существующие в информационной базе, при реструктуризации будут отнесены на "пустое" субконто. В дальнейшем при редактировании проводок пустое субконто можно заменить на конкретный объект аналитического учета. Наоборот, если у счета отключается ведение аналитического учета по субконто, то при реструктуризации информационной базы из всех проводок по этому счету будут удалены ссылки на субконто.

Если установить признак **Только обороты**, то по этому субконто не будут разворачиваться остатки счета, а будут разворачиваться только обороты.

Признаки **Учет по сумме**, **Учет по валютной сумме**, **Учет по количеству** позволяют регулировать использование данного субконто с точки зрения суммового, валютного и количественного учета.

Операция

Объект метаданных "Операция" предназначен для настройки структуры заголовка операции, т. е. информации, описывающей всю операцию. Параметры и структура проводок описываются в объекте метаданных "Проводка" (см. разд. "Проводка" данной главы). Редактирование свойств операции производится в диалоговом окне **Операция** (рис. 4.98).

Поле **Длина содержания** устанавливает длину содержания операции.

Управляющие элементы, объединенные в группу **Сумма операции**, управляют размерностью суммы операции и ее представлением.

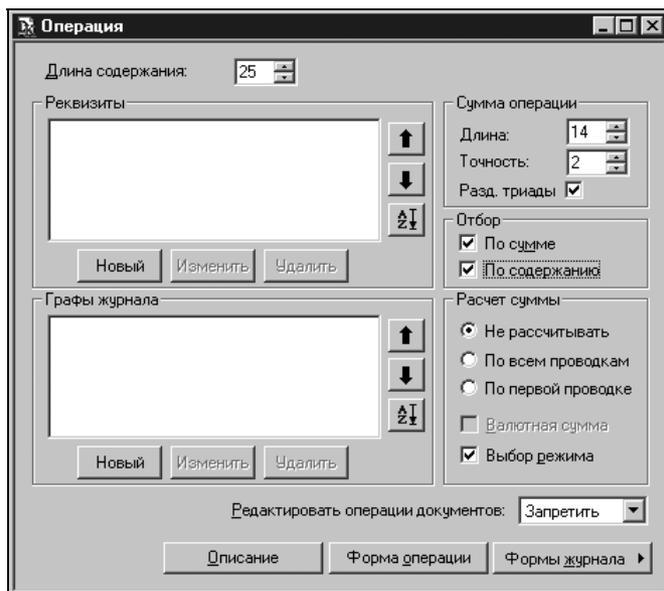


Рис. 4.98. Диалоговое окно **Операция**

Отбором называется механизм, дающий пользователю возможность работать в журналах с операциями, отобранными по какому-либо значению. Включением соответствующих опций группы **Отбор** можно задавать выполнение отбора операций по сумме или содержанию операции при работе с системой 1С:Предприятие. При задании признаков отбора пользователем в журнале операций в список **Виды отбора** будут дополнительно включены те виды отбора, которые указаны в опциях группы **Отбор** в Конфигураторе (рис. 4.99).

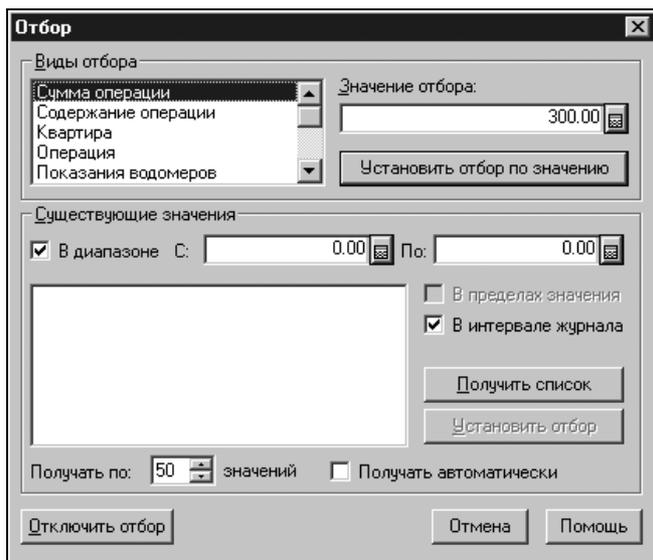


Рис. 4.99. Диалоговое окно **Отбор** для задания признаков отбора в журнале операций

Управляющие элементы группы **Расчет суммы** устанавливают порядок расчета суммы операции:

- Не рассчитывать** — сумма операции вводится вручную в соответствующий реквизит диалога;
- По всем проводкам** — сумма операции вычисляется как сумма всех проводок операции;
- По первой проводке** — сумма операции принимается равной сумме первой проводки операции;
- Валютная сумма** — если опция включена, для вычислений будет использоваться валютная сумма проводок операции. Этот управляющий элемент становится доступным, если выбран какой-либо способ вычисления суммы операции — по всем проводкам или по первой проводке.

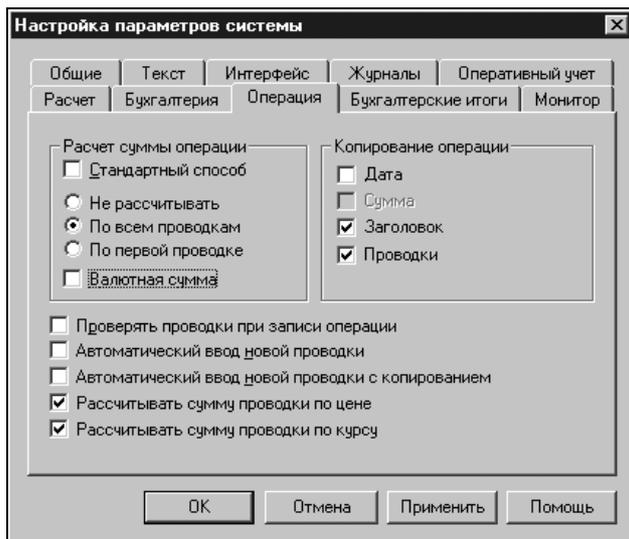


Рис. 4.100. Диалоговое окно **Настройка параметров системы** для пользователя

Если включена опция **Выбор режима**, то при работе с системой 1С:Предприятие пользователь сам может устанавливать порядок расчета суммы операции. Такой выбор будет выполняться в режиме настройки параметров системы 1С:Предприятие (рис. 4.100).

Опция **Редактировать операции документов** (см. рис. 4.98) может принимать одно из трех значений:

- Разрешить** — разрешить пользователю редактировать операции, сформированные документами;
- Запретить** — запретить пользователю редактировать операции, сформированные документами;
- Выборочно** — возможность редактирования операции, сформированной документами, устанавливается в Конфигураторе для каждого вида документов индивидуально.

Если выбран режим **Редактировать операции документов - выборочно**, в окне редактирования документа становится доступен флажок **Редактировать операцию**. Установка или снятие этого флажка, соответственно, разрешает или запрещает пользователю редактировать вручную операции, сформированные документами этого вида.

Внимание!

При перепроведении документов операции формируются заново, следовательно, отредактированные пользователем операции будут удалены и заменены на автоматически сформированные.

Управляющие элементы, объединенные в группу **Реквизиты**, позволяют создавать, удалять и редактировать дополнительные реквизиты операции. Приемы работы с элементами этой группы совпадают с общими приемами управления списками объектов метаданных. Дополнительные реквизиты операции позволяют включать в операцию любую необходимую информацию помимо даты, номера, суммы и содержания операции. Дополнительные реквизиты могут быть размещены в форме операции.

Группа **Графы журнала** объединяет управляющие элементы, предназначенные для ведения списка дополнительных колонок журнала операций и определения их свойств.

Нажатие кнопки **Форма операции** вызывает редактор форм для редактирования формы операции.

Нажатие кнопки **Формы журнала** вызывает редактор форм для редактирования формы журнала операций.

Проводка

Окно редактирования **Проводка** служит для редактирования свойств объекта метаданных "Проводка" (рис. 4.101). Это окно вызывается двойным щелчком мыши на ключевом слове **Проводка** в дереве метаданных.

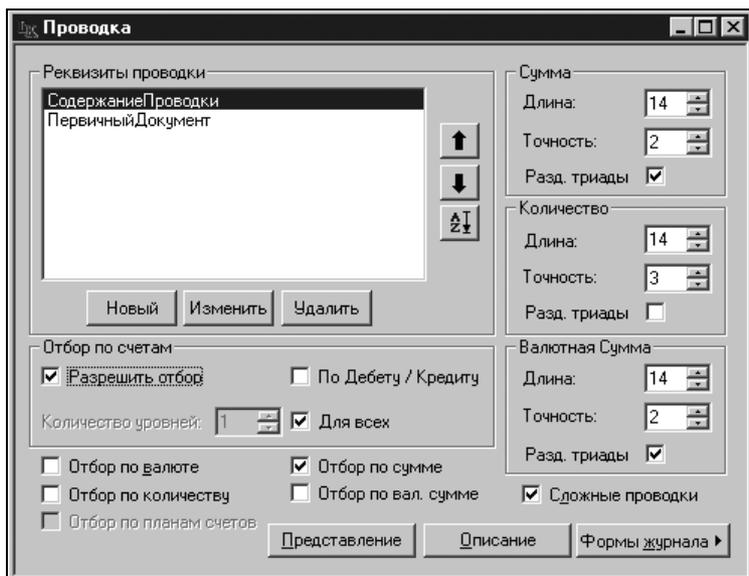


Рис. 4.101. Диалоговое окно Проводка

Управляющие элементы, объединенные в группы **Сумма**, **Количество**, **Валютная Сумма**:

- управляют размерностью, соответственно, суммы, количества и валютной суммы проводок;
- управляют изображением этих реквизитов при работе с системой 1С:Предприятие;
- включают возможность отбора проводок в журнале проводок по одному из этих критериев.

Управляющие элементы группы **Отбор по счетам** позволяют настроить отбор проводок по счетам:

- **Разрешить отбор** — включает возможность отбора проводок в журнале проводок по номерам счетов и субсчетов. Наличие отбора по счетам позволяет просматривать проводки в журнале проводок, отобранные по какому-либо одному счету. Кроме этого, наличие такого отбора ускоряет получение отчетов с отбором по счетам.
- **По Дебету / Кредиту** — управляет отбором проводок по счетам в журнале проводок. Если опция включена, при работе с журналом проводок можно будет отбирать проводки, у которых указанный счет стоит или только в дебете, или только в кредите. Если опция **По Дебету / Кредиту** отключена, то остается возможность отбирать все проводки, в которых присутствует указанный счет — либо в дебете, либо в кредите.
- **Количество уровней** — позволяет указать, до какого уровня вложенности будет выполняться отбор по счетам (субсчетам). Если включена опция **Для всех**, отбор будет выполняться по указанному счету и всем субсчетам этого счета. Если опция **Для всех** отключена, то для определения "глубины" отбора будет использоваться число, указанное в поле **Количество уровней**.

Опции **Отбор по...** управляют возможностью выполнять отбор проводок в журнале проводок по различным условиям. Включение той или иной опции разрешает выполнять соответствующий отбор проводок в журнале при работе с системой 1С:Предприятие. При задании признаков отбора пользователем в журнале проводок в список **Виды отбора** дополнительно будут включены те виды отбора, которые указаны в опциях отбора в Конфигураторе (рис. 4.102).

Включение опции **Сложные проводки** (см. рис. 4.101) разрешает ввод сложных проводок при редактировании операции. Сложная проводка представляет собой совокупность корреспонденций, первая из которых устанавливает счет дебета (кредита) проводки и общую сумму проводки, а остальные —

корреспондирующие счета кредита (дебета) и суммы, из которых складывается общая сумма проводки.

Управляющие элементы группы **Реквизиты проводки** служат для ведения списка дополнительных реквизитов проводки. Приемы работы с элементами этой группы совпадают с общими приемами управления списками объектов метаданных. Дополнительные реквизиты позволяют хранить любую необходимую информацию в дополнение к той, которая содержится в стандартных реквизитах проводки (корреспондирующие счета, суммы, количество и др.).

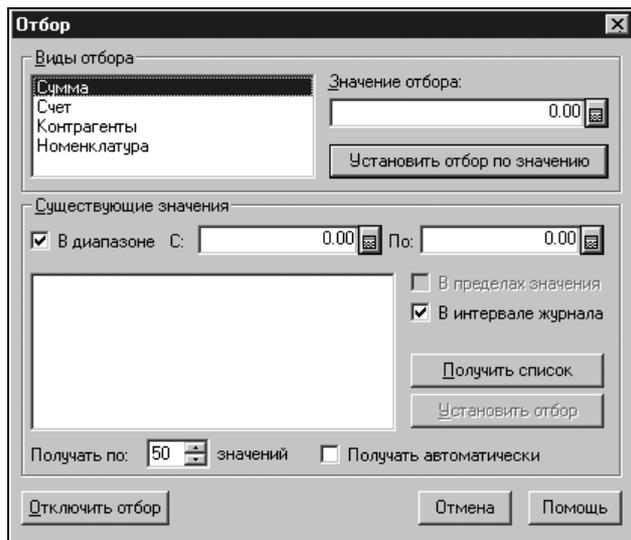


Рис. 4.102. Диалоговое окно **Отбор** для задания признаков отбора в журнале проводок

Нажатие кнопки **Представление** вызывает диалог для редактирования представления проводки.

Нажатие кнопки **Формы журнала** вызывает редактор форм и позволяет редактировать, соответственно, формы журнала проводок.

Регистры

Регистры служат для создания регистра, определения разрезов и вида хранимой итоговой информации. Пустой диалог автоматически вызывается на экран при создании нового регистра. Создадим регистр **Заявки** для оперативного отслеживания накопления и выполнения заявок на ремонт (рис. 4.103).

Для редактирования свойств уже существующего регистра диалог можно вызвать на экран одним из следующих способов:

- ❑ установите указатель мыши на наименовании нужного регистра в окне **Метаданные** и два раза щелкните левой кнопкой мыши;
- ❑ установите указатель мыши на наименовании нужного регистра и нажати-ем правой кнопки мыши вызовите контекстное меню; из этого меню сле-дует выбрать функцию **Редактировать**;
- ❑ выберите название нужного регистра мышью или клавишами; в меню **Действия** выберите пункт **Редактировать элемент**.



Рис. 4.103. Диалоговое окно **Регистр Заявки**

Поле **Идентификатор** — имя регистра. Представляет собой строку длиной не более 128 символов, состоящую из букв, цифр и знаков подчеркивания. Идентификатор не должен начинаться с цифры и содержать пробелы. Идентификатор позволяет обращаться к информации, хранящейся в регистре, из модулей.

Поле **Комментарий** — полное наименование регистра, служит для расшифровки идентификатора.

При создании нового регистра необходимо задать, в каких разрезах и какой набор показателей итоговой информации необходимо сохранять. Для этого служат управляющие элементы групп **Измерения** и **Ресурсы**. С функциональной точки зрения элементы этих групп одинаковы.

Для регистров определены следующие атрибуты:

- Приход;**
- Расход;**
- Измерение;**
- Ресурс;**
- Реквизит.**

Из приведенного перечня видно, что атрибуты **Приход** и **Расход** — базовые атрибуты. **Измерение** — обобщенное название атрибутов регистра, которые задаются пользователем в Конфигураторе в группе **Измерения**. **Ресурс** — обобщенное название атрибутов регистра, которые задаются пользователем в Конфигураторе в группе **Ресурсы**. **Реквизит** — обобщенное название атрибутов регистра, которые задаются пользователем в Конфигураторе в группе **Реквизиты**. В группе **Измерения** перечислены все атрибуты учетных позиций (см. рис. 4.103):

- Услуга;**
- Квартира;**
- ДатаИсполнения;**
- Заявка.**

В группе **Ресурсы** перечислены все количественные и суммовые показатели учетных позиций (см. рис. 4.103):

- Количество;**
- Долг.**

В группе **Реквизиты** перечисляются все реквизиты, по которым необходим отбор (см. рис. 4.101).

Задание значений атрибутов регистров обычно применяется в модулях документов в предопределенной процедуре *ОбработкаПроведения* (см. разд. "Проведение документа" данной главы).

Кнопка **Новый** вызывает на экран пустой диалог для задания свойств нового измерения или ресурса — краткого и полного наименования, типа значения, длины, других характеристик.

Кнопка **Изменить** открывает заполненный диалог для изменения свойств измерения или ресурса.

Кнопка **Удалить** удаляет наименование измерения или ресурса из соответствующего списка.

Кнопка  (сдвинуть вверх) — сдвигает наименование измерения или ресурса в списке на одну позицию вверх.

Кнопка  (сдвинуть вниз) — сдвигает наименование измерения или ресурса в списке на одну позицию вниз.

Кнопка  (отсортировать список) — сортирует список измерений или ресурсов по алфавиту.

Управляющие элементы группы **Тип регистра** задают тип регистра — оборотный или регистр остатков. В зависимости от типа регистра накапливается информация в разных разрезах. Если выбрана опция **Обороты**, необходимо задать значение поля **Периодичность** выбором из списка:

- Декада;
- Месяц;
- Квартал;
- Год.

Периодичность задает уровень детализации итоговых таблиц.

Кнопка **Описание** вызывает текстовый редактор для создания пользовательского описания регистра.

Виды расчетов

Виды расчета предназначены для выполнения расчетов по определенным алгоритмам через журнал расчетов. Для задания свойств и модуля расчета служит диалоговое окно **Вид расчета**. Создадим вид расчета **ХолоднаяВодаПоВодомеру** для расчета коммунальной услуги (рис. 4.104).

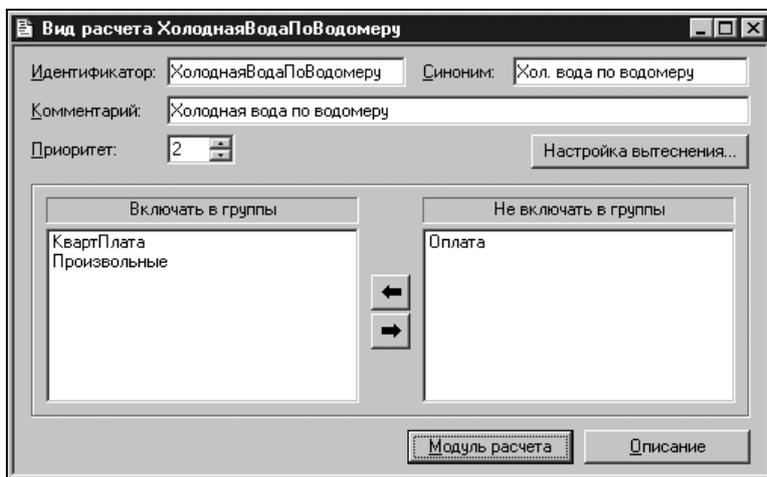


Рис. 4.104. Диалоговое окно вида расчета ХолоднаяВодаПоВодомеру

Идентификатор — наименование вида расчета. Представляет собой строку длиной не более 16 символов, состоящую из букв, цифр и знаков подчеркивания. Не должен начинаться с цифры и содержать пробелы. Идентификатор позволяет обращаться к реквизитам документа из модулей.

Комментарий — полное наименование вида расчета, служит для расшифровки идентификатора.

Синоним — наименование вида расчета, которое при работе с системой 1С:Предприятие будет выдаваться вместо идентификатора во всех диалогах и списках.

Приоритет задает очередность, в которой располагаются расчеты в журнале расчетов и, следовательно, рассчитываются.

Настройка вытеснения предназначена для определения тех видов расчета, которые вытесняются данным видом расчета, и для определения тех видов расчета, которые сами вытесняют данный вид расчета. Если один вид расчета вытесняет другой, то он влияет на интервал действия вытесняемого вида расчета, уменьшая его. При этом совокупное время действия вытесняющего и вытесняемого видов расчета будет составлять расчетный период. Так, например, если вид расчета, вычисляющий дни болезни сотрудника, определен в настройке как вытесняющий вид расчета "Оплата по окладу", то при расчете зарплаты интервал действия вида расчета "Оплата по окладу" будет меньше расчетного периода (в данном случае, месяца) на количество дней болезни. Один вид расчета может быть как вытесняемым, так и вытесняющим. Так, например, вид расчета "Оплата по тарифу" может вытесняться видом расчета "Больничный лист", а с другой стороны, сам может быть вытесняющим по отношению к виду расчета "Оплата по окладу". По нажатию кнопки **Настройка вытеснения** вызывается диалоговое окно **Порядок вытеснения** (рис. 4.105).

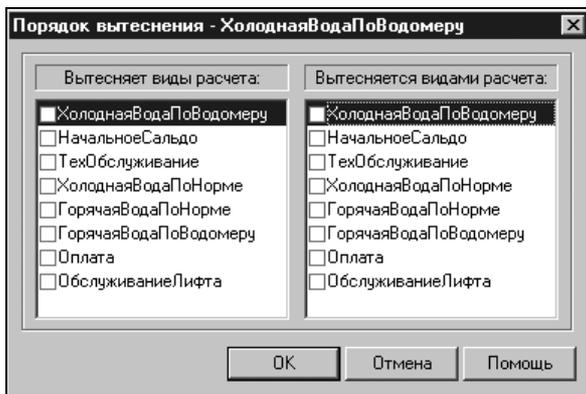


Рис. 4.105. Диалоговое окно Порядок вытеснения

В этом окне все виды расчета, включая определяемый, помещены в два списка:

- Вытесняет виды расчета** — список видов расчета, которые вытесняются данным видом расчета;
- Вытесняется видами расчета** — список видов расчета, которые вытесняют данный вид расчета.

Включать в группы (см. рис. 4.101) — список групп видов расчета, в которые входит данный вид расчета. Первоначально при создании нового вида расчета, вид расчета не входит ни в одну группу, т. е. окно **Включать в группы** пустое.

Не включать в группы — список групп видов расчета, в которые не входит данный вид расчета. Первоначально при создании нового вида расчета он не входит ни в одну группу, т. е. окно **Не включать в группы** содержит полный перечень имеющихся групп видов расчета.

В нашем случае предварительно были созданы группы:

- КвартПлата;**
- Произвольные;**
- Оплата.**

Списки групп формируются следующим образом:

1. Отметить одну или несколько групп в окне.
2. Перенести выделенные группы в окно **Включать в группы** или **Не включать в группы** кнопками  и .

Нажатие кнопки **Описание** вызывает текстовый редактор для создания пользовательского комментария для каждого конкретного вида расчета.

Нажатие кнопки **Модуль расчета** вызывает текстовый редактор для создания алгоритма, основным назначением которого является вычисление и запись реквизита **Результат** в журнал расчетов. В нашем журнале расчетов объектом является **Квартира** — элемент справочника "Квартиры", в котором реквизит **ПоказаниеХолВоды** — периодический. С учетом этого модуль расчета следующий:

Процедура ПровестиРасчет()

// Процедура выполняется при проведении расчета

ПоказаниеПредьдущее = Объект.ПоказаниеХолВоды.Получить (ДатаНачала);

ПоказаниеТекущее = Объект.ПоказаниеХолВоды.Получить (ДатаОкончания);

Документ.ПолучитьСтрокуПоНомеру (НомерСтрокиДокумента);

Результат = (ПоказаниеТекущее -ПоказаниеПредьдущее) *Документ.Цена *

Объект.Льгота.Получить (ДатаОкончания);

КонецПроцедуры

Группы расчетов

Группа расчетов служит для задания условий отбора по группе в запросах. Диалог **Группа расчетов КвартПлата** предназначен для редактирования свойств группы расчетов (рис. 4.106).

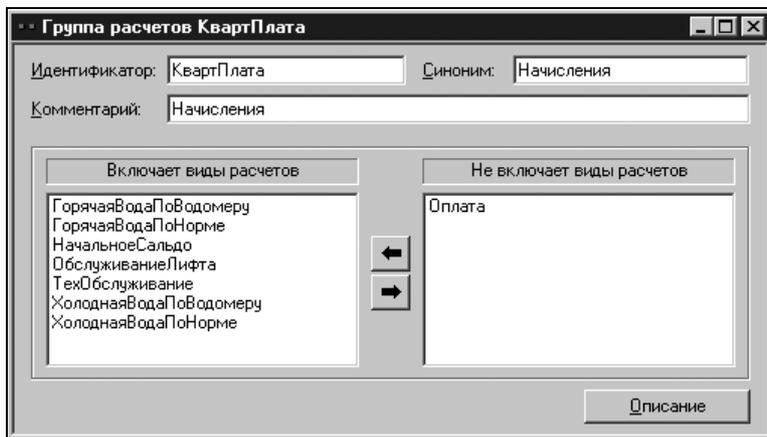


Рис. 4.106. Диалоговое окно **Группа расчетов КвартПлата**

Для вызова диалога на экран необходимо дважды щелкнуть мышью идентификатор группы расчетов в окне **Метаданные**.

Поле **Идентификатор** — краткое наименование группы расчетов. Идентификатор используется для обозначения группы расчетов в списках и диалогах, а также используется в модулях для обращения к расчетам, входящим в группу. Идентификатор представляет собой строку длиной не более 128 символов, состоящую из букв, цифр и знаков подчеркивания. Идентификатор не должен начинаться с цифры и содержать пробелы. В идентификаторе допускается смешивать прописные и строчные буквы.

Поле **Комментарий** — полное наименование группы расчетов, служит для расшифровки идентификатора. Комментарий представляет собой произвольную строку символов и может содержать любой текст, раскрывающий смысл идентификатора. Каких-либо ограничений на длину комментария не накладывается, здесь следует руководствоваться принципом разумной достаточности.

Поле **Синоним** — наименование группы видов расчета, которое при работе с системой 1С:Предприятие будет выдаваться вместо идентификатора во всех диалогах и списках.

Включает виды расчетов — список видов расчета, которые входят в данную группу. Вновь созданная группа не содержит ни одного вида расчета.

Не включает виды расчетов — список видов расчета, которые не входят в данную группу. Для вновь созданной группы окно содержит полный перечень существующих в конфигурации видов расчета.

Нажатие кнопки  включает в группу виды расчета, выделенные в списке **Не включает виды расчетов**.

Нажатие кнопки  исключает из группы виды расчетов, выделенные в списке **Включает виды расчетов**.

Нажатие кнопки **Описание** вызывает текстовый редактор для создания пользовательского описания группы расчетов.

Правила пересчета

"Правила перерасчета" являются вспомогательными объектами метаданных, предназначенными для автоматического отслеживания актуальности результатов проведенных расчетов при вводе новых записей журнала расчетов, удалении существующих или ручном исправлении результата расчета. При создании правила перерасчета в Конфигураторе определяются *ведущие* виды расчетов, т. е. такие, при редактировании которых правило перерасчета "срабатывает", и *зависимые* виды расчетов — те, которые должны быть перерасчитаны при срабатывании данного правила. При описании правила пересчета выбирается тип его пересчета. Правило перерасчета может быть трех типов (рис. 4.107):

- перерасчет "по текущему периоду";
- пересчет по периоду ведущего вида расчета;
- по периоду расчета "будущих периодов".

Поле **Идентификатор** — краткое наименование правила пересчета. Идентификатор используется для обозначения правила пересчета в модулях. Представляет собой строку длиной не более 128 символов, состоящую из букв, цифр и знаков подчеркивания. Не должен начинаться с цифры и содержать пробелы.

Поле **Комментарий** — полное наименование правила пересчета, служит для расшифровки идентификатора.

Поле **Синоним** — наименование правила пересчета, которое при работе с системой 1С:Предприятие будет выдаваться вместо идентификатора во всех диалогах и списках.

При вводе вида расчета... — список ведущих видов расчета, на основании которых выполняется конкретное правило перерасчета.

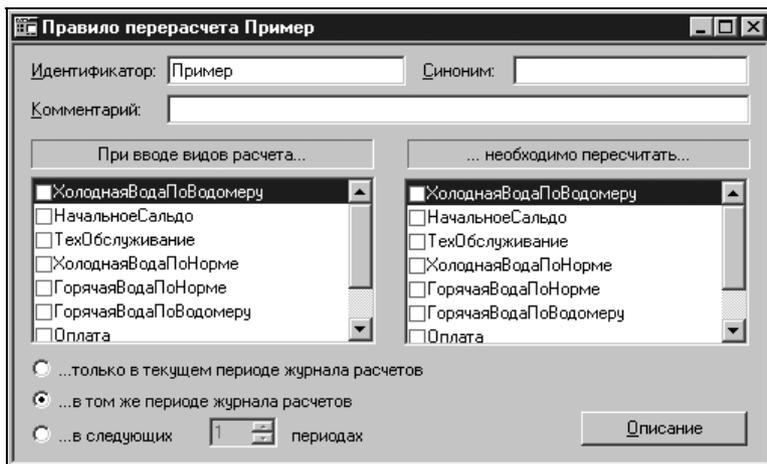


Рис. 4.107. Диалоговое окно Правило перерасчета

...необходимо пересчитать... — список зависимых видов расчета, которые должны быть перерассчитаны при выполнении конкретного правила.

...только в текущем периоде журнала расчета — тип пересчета, при котором пересчитываются зависимые виды расчетов текущего периода действия журнала расчетов.

...в том же периоде журнала расчета — тип пересчета, при котором пересчитываются зависимые виды расчетов, период действия которых совпадает с периодом действия ведущих видов расчета.

...в следующих ... периодах — тип расчета, позволяющий пересчитать зависимые виды расчетов будущих периодов, следующих за периодом ведущего вида расчета. При данном типе пересчета указывается количество будущих периодов, которые должны быть учтены при пересчете зависимых видов расчета.

Нажатие кнопки **Описание** вызывает текстовый редактор для создания пользовательского описания группы расчетов.

Журналы расчетов

Журналы расчетов предназначены для учета расчетных действий по тем или иным объектам расчета. Объектом расчета может служить любой имеющийся в системе справочник. Журнал расчетов характеризуется не только тем справочником, по которому он создан, но также периодичностью его ведения и датой начала ведения журнала. Все расчеты в нем выполняются в пределах определенного временного интервала. Каждая запись журнала расчетов, со-

ответствующая одному акту расчета, также имеет временное протяжение (т. е. имеет дату начала и дату окончания). Ввод записей в журнал расчетов возможен через ввод документов расчета.

Создадим журнал расчетов "Начисления" (рис. 4.108).

Любой журнал расчетов в момент создания уже имеет набор обязательных реквизитов, которые необходимы для правильного ведения журнала. Обязательными реквизитами являются:

- Объект** — ссылка на элемент справочника объектов расчета;
- Вид расчета** — ссылка на вид расчета, используемый в конкретной записи журнала расчетов;
- Результат** — результат расчета;
- Дата начала** — дата начала действия конкретного расчета;
- Дата окончания** — дата окончания действия конкретного расчета;
- Документ** — документ — основание текущей записи журнала;
- Родительский документ** — ссылка на документ, который ввел конкретную запись журнала расчетов;
- Первичная запись** — ссылка на запись журнала расчетов, перерасчетом которой является текущая запись.

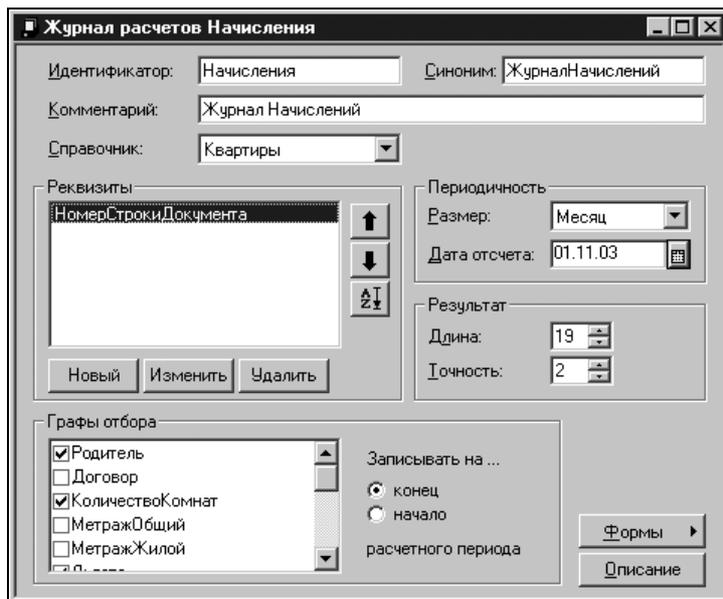


Рис. 4.108. Диалоговое окно Журнал расчетов Начисления

Кроме того, для каждого журнала можно задать дополнительные реквизиты (рис. 4.109).

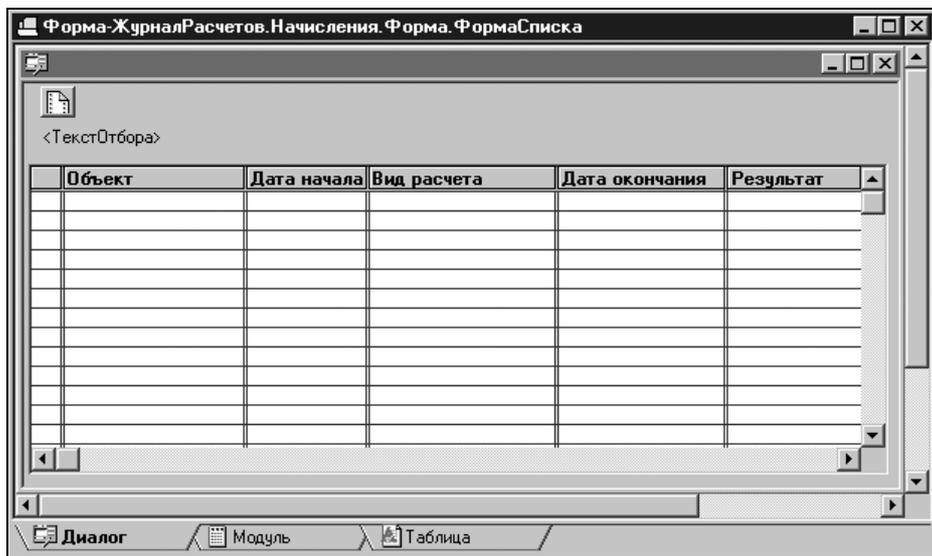


Рис. 4.109. Форма Журнал расчетов

Поле **Идентификатор** — имя журнала расчетов. Представляет собой строку длиной не более 128 символов, состоящую из букв, цифр и знаков подчеркивания. Не должна начинаться с цифры и содержать пробелы. Идентификатор позволяет из модулей обращаться к записям, хранящимся в журнале.

Поле **Синоним** — произвольная последовательность символов. Если синоним указан, он будет выдаваться во всех диалогах и списках вместо идентификатора.

Поле **Комментарий** — полное наименование журнала расчетов. Служит для расшифровки идентификатора.

Поле **Справочник** — идентификатор справочника, элементы которого будут являться объектами для расчетов.

Внимание!

Система не позволяет изменить эту характеристику журнала расчетов в том случае, если журнал расчетов уже содержит записи.

Поле **Размер** — период проведения расчетов. Возможные значения: день, неделя, месяц, квартал, год. Для нашего журнала, в котором проводится расчет начислений за коммунальные услуги, период расчета равен месяцу.

Поле **Дата отсчета** — дата, начиная с которой начинается ведение журнала, а также и отсчет первого периода. Например, если введена дата 01.11.03, а периодичность равна месяцу, то начальный период журнала длится с 01.11.03 по 01.12.03.

Поле **Результат** — результат вычисления вида расчета. Характеризуется длиной и точностью.

Реквизиты — перечень реквизитов журнала расчета, которые вводятся дополнительно к уже имеющимся обязательным реквизитам. Используя управляющие элементы группы **Реквизиты**, можно создавать и удалять дополнительные реквизиты журнала, а также редактировать их свойства.

Нажатие кнопки **Новый** создает новый реквизит журнала расчетов и вызывает палитру свойств для редактирования свойств реквизита.

Нажатие кнопки **Изменить** открывает палитру свойств для редактирования свойств журнала расчетов.

Нажатие кнопки **Удалить** удаляет наименование реквизитов из списка.

Порядок следования наименований реквизитов в списке сверху вниз соответствует расположению дополнительных колонок в журнале слева направо, если иной порядок не определен при создании формы журнала.

Нажатие кнопки  (**Переместить вверх**) сдвигает наименование реквизита в списке на одну позицию вверх.

Нажатие кнопки  (**Переместить вниз**) сдвигает наименование реквизита в списке на одну позицию вниз.

Нажатие кнопки  (**Упорядочить**) сортирует список реквизитов по алфавиту.

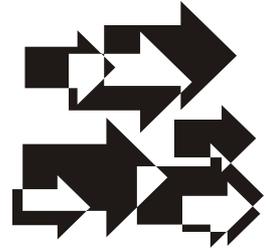
Графы отбора — графы, по которым будет выполняться отбор строк в журнале расчетов. Графами отбора в журнале расчетов могут служить реквизиты справочника объектов расчета, назначенного данному журналу. Кроме того, в качестве граф отбора могут использоваться значения **Родитель** и **Владелец** этого справочника. Использование тех или иных реквизитов справочника в качестве граф отбора указывается знаком у наименования реквизита.

Переключатель **Записывать на начало/конец расчетного периода** задает правила заполнения граф отбора для периодических реквизитов справочника. Если переключатель установлен в положение "**конец**", то отбор по периодическому реквизиту будет проводиться по его значению на конец расчетного периода журнала расчетов.

Нажатие кнопки **Формы** вызывает диалог **Формы списков** для создания экранных и печатных форм журнала расчетов.

Нажатие кнопки **Описание** вызывает текстовый редактор для создания пользовательского описания журнала расчетов.

ГЛАВА 5



Пользователи

В *главе 1* уже обсуждался вопрос взаимосвязи трех составных частей конфигурации **Метаданные**, **Интерфейсы** и **Права** с точки зрения привязки их к пользователю. Кроме формирования удобного интерфейса, не перегруженного лишними функциями и меню, для конкретного пользователя, есть еще ряд серьезных причин для организации связки "метаданные — права — интерфейс".

Механизм контроля ссылочной целостности в системе 1С:Предприятие требует монопольного режима запуска. Для многопользовательской напряженной работы перерыв — непозволительная роскошь. А ведь есть еще и перерывы на обновление и сохранение, пересчет итогов, восстановление последовательности документов и т. д. и т. п. Неплохо организовать "эксклюзивный" вариант прав с режимом непосредственного удаления объектов "эксклюзивному" пользователю.

Конструкторы, конечно, позволяют "включать команды в пользовательское меню", но после десяти справочников при "хорошем" разрешении экрана для подслеповатого бухгалтера меню поползет вправо или влево. Разбиение на подменю возможно только при создании специального очень полезного интерфейса.

В меню **Операция** присутствуют все самые важные и самые критичные режимы работы, что легко может свести на нет все усилия по контролю над действиями пользователей. В специальном интерфейсе можно отключить меню **Операция**.

Для расследования некоторых ситуаций, возникающих в процессе работы, ведется журнал регистраций подобно журналу регистраций Windows. Из журнала регистраций всегда можно определить виновника искомого события, если в конфигурации установлена авторизация.

Вся работа по созданию списка пользователей, присвоению паролей, закреплению за пользователями прав и интерфейса ведется в окне **Пользователи** (рис. 5.1).

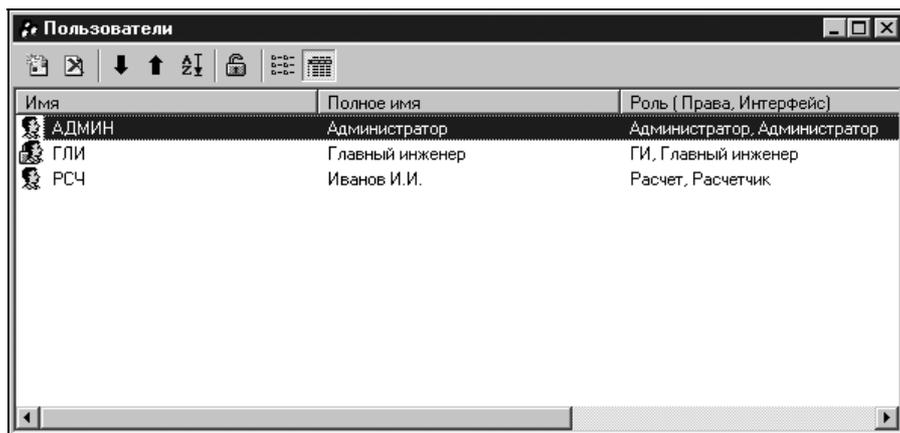


Рис. 5.1. Диалоговое окно **Пользователи**

Это окно вызывается на экран выбором пункта **Пользователи** из меню **Администрирование** главного меню программы. Окно со списком пользователей имеет 3 колонки. В колонке **Имя** выводится список пользователей, зарегистрированных для работы с системой 1С:Предприятие. В колонке **Полное имя** выдаются расшифровки имен пользователей. В колонке **Роль** выводятся (через запятую) наименование набора прав, присвоенных этому пользователю, и наименование интерфейса, который будет использоваться при работе этого пользователя. Если какое-то из этих свойств пользователя указано с ошибкой, в этой колонке будет выдано соответствующее сообщение.

Чтобы создать нового пользователя, необходимо выбрать пункт **Новый** в меню **Действия** главного меню конфигуратора. На экран будет выдан запрос для ввода имени нового пользователя (рис. 5.2).

Имя пользователя представляет собой произвольную строку длиной не более 24 символов и не должно содержать пробелы и специальные символы.

Внимание!

Имя нового пользователя не должно совпадать с именами, уже существующими в списке. В противном случае будет выдано предупреждение на экран и запрос повторится (рис. 5.3).

Диалог **Свойства пользователя** предназначен для редактирования характеристик входа и работы пользователя в системе 1С:Предприятие. Для вызова

диалога на экран необходимо дважды щелкнуть левой кнопкой мыши на имени нужного пользователя в окне **Пользователи**.

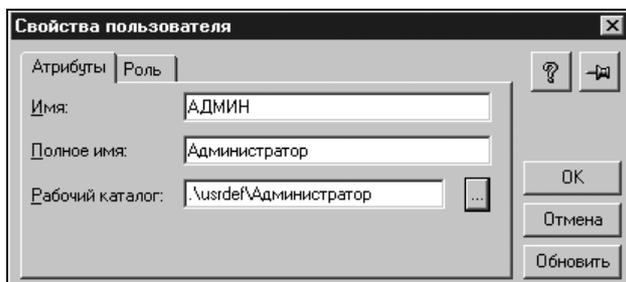


Рис. 5.2. Диалоговое окно **Свойства пользователя** на вкладке **Атрибуты**

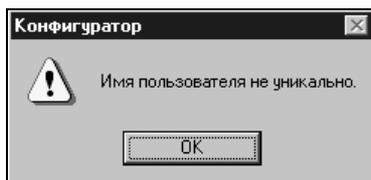


Рис. 5.3. Предупреждение о дублировании имени пользователя

Рабочий каталог

В *главе 1* говорилось о том, что при задании рабочего каталога пользователя рекомендуется в каталоге <каталог БД>/USRDEF создать папку с именем пользователя и выбрать ее в качестве рабочей. Рекомендуется это отнюдь не для соблюдения порядка, а по двум довольно серьезным причинам.

- Каталог пользователя используется системой для организации разделенного доступа к данным. При попытке запуска пользователем, у которого одинаковый рабочий каталог с уже присутствующим пользователем, программа не пустит данного пользователя в систему с предварительным предупреждением (рис. 5.4).

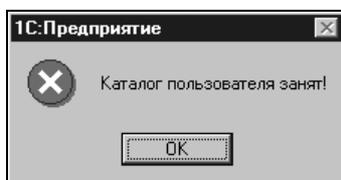


Рис. 5.4. Предупреждение о дублировании рабочего каталога пользователя

- При восстановлении конфигурации из архива каталогов, указанных в качестве рабочих для каждого пользователя, может не оказаться. При попытке запуска программы пользователем, рабочий каталог которого отсутствует, программа не пустит данного пользователя в систему с предварительным предупреждением (рис. 5.5).

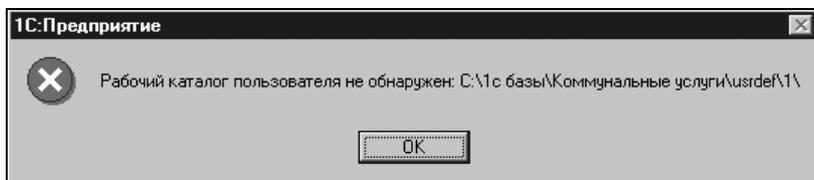


Рис. 5.5. Предупреждение об отсутствии рабочего каталога пользователя

Во избежание неприятностей при восстановлении из архива, в программе предусмотрено построение пути к рабочему каталогу относительно каталога базы данных. Если при задании рабочего каталога был выбран какой-либо каталог базы данных, программа предложит пользователю запрос на данное построение (рис. 5.6).

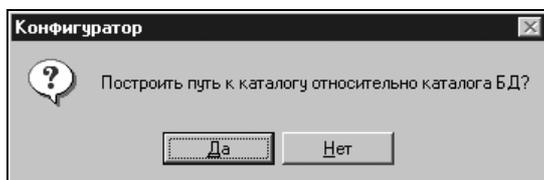


Рис. 5.6. Запрос на построение относительного пути к рабочему каталогу

Программа сама построит относительный путь к рабочему каталогу пользователя (рис. 5.7). При запуске программа определит рабочий каталог в каталоге базы данных.

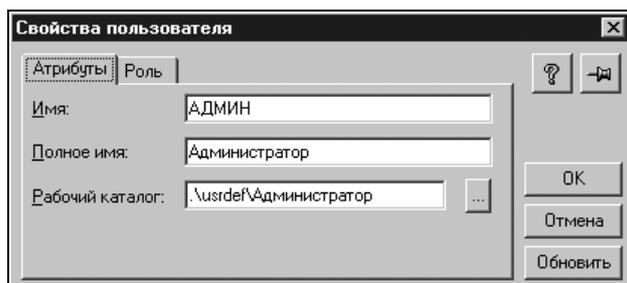


Рис. 5.7. Диалоговое окно **Свойства пользователя** с относительным путем к рабочему каталогу пользователя

На вкладке **Роль** окна **Свойства пользователя** предлагается назначить данному пользователю интерфейс выбором из списка существующих интерфейсов и набор прав из списка созданных наборов (рис. 5.8).

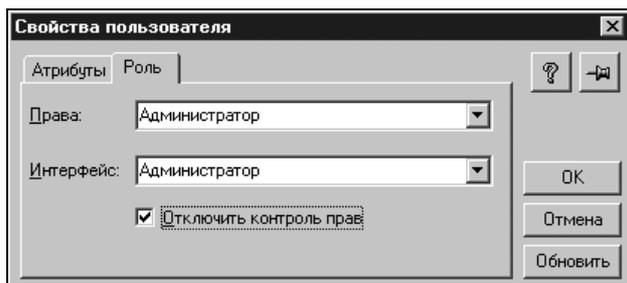


Рис. 5.8. Диалоговое окно **Свойства пользователя** на вкладке **Роль**

Внимание!

После удаления интерфейса, назначенного пользователю, необходимо переназначить интерфейс данному пользователю. В противном случае программа запустится с минимальным интерфейсом, предусмотренным системой, предварительно выдав предупреждение на экран (рис. 5.9).

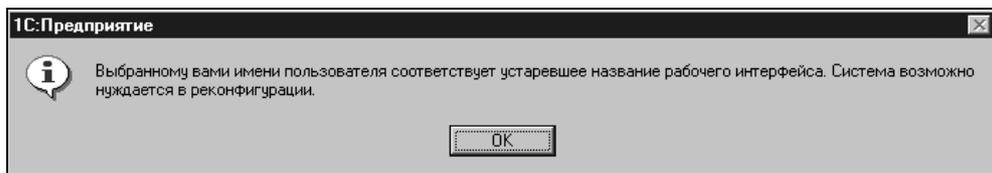


Рис. 5.9. Предупреждение об отсутствии назначенного интерфейса

Как правило, создавать нового пользователя с нуля утомительно. Гораздо проще скопировать уже существующий и отредактировать его. Чтобы создать нового пользователя по образцу существующего, необходимо выполнить следующие действия:

1. Выбрать в списке имя пользователя, который будет служить образцом.
2. Выбрать пункт **Копировать** в меню **Действия** главного меню Конфигуратора.
3. Выбрать пункт **Вставить** в меню **Действия** главного меню Конфигуратора.

В список будет внесено имя нового пользователя. Имя пользователя может быть преобразовано с целью соблюдения уникальности. Остальные свойства

нового пользователя будут такие же, как и у пользователя, выбранного в качестве образца.

Чтобы удалить пользователя из списка, необходимо выполнить следующие действия:

1. Выбрать в списке имя пользователя, которого необходимо удалить.
2. Выбрать пункт **Удалить** в меню **Действия** главного меню Конфигуратора.

Внимание!

После выбора пункта **Удалить** указанный пользователь будет сразу удален!

Если необходимо восстановить случайно удаленного пользователя, следует закрыть окно со списком пользователей, отказавшись от сохранения списка.

Для упорядочивания списка пользователей необходимо использовать кнопки:

-  ("**Переместить вниз**");
-  ("**Переместить вверх**");
-  ("**Упорядочить**") на панели инструментов окна **Пользователи**.

Чтобы назначить пользователю пароль, необходимо выполнить следующие действия:

1. Выбрать в списке имя пользователя, которому необходимо присвоить пароль.
2. Выбрать пункт **Изменить пароль** в меню **Действия** главного меню Конфигуратора.
3. В запросе **Смена пароля** ввести пароль пользователя. Пароль представляет собой символьную строку длиной не более 10 символов и не должен включать пробелы и специальные символы. Вводимый пароль на экране не появляется, вместо него выводятся символы (*).
4. Нажать кнопку **ОК**.
5. Запрос **Смена пароля** будет выдан на экран еще раз. Необходимо повторить ввод пароля.
6. Нажать кнопку **ОК**.

Пароль будет присвоен пользователю (рис. 5.10).

Для снятия пароля следует оба раза в запросе **Смена пароля** оставить поле ввода пустым.

Внимание!

Если пользователь забыл свой пароль, необходимо задать ему новый.

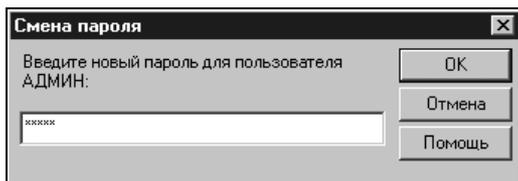


Рис. 5.10. Диалоговое окно **Смена пароля**

Интерфейсы

Средства администрирования, содержащиеся в Конфигураторе системы 1С:Предприятие, позволяют создавать индивидуальные пользовательские интерфейсы для различных категорий пользователей. Такой индивидуальный интерфейс расширяет стандартный интерфейс системы 1С:Предприятия, предоставляя пользователю системы дополнительные панели инструментов и добавляя новые пункты в главное меню программы.

Для создания, редактирования и настройки меню индивидуального интерфейса в Конфигураторе используется специализированный редактор пользовательских меню.

Редактор меню предназначен для создания, редактирования и настройки меню индивидуального пользовательского интерфейса. Редактор меню можно вызвать из окна **Конфигурация** на вкладке **Интерфейсы**.

Главное меню системы 1С:Предприятие может содержать команды двух типов: системные и пользовательские. Системные команды предоставляют доступ к набору функций, которые поддерживаются исполняемым модулем системы 1С:Предприятие. Этот набор постоянный и пользователем меняться не может (рис. 5.11). А вот подменю системных функций пользователь может расширять.

Создадим новый пользовательский интерфейс. Чтобы создать новый пользовательский интерфейс, необходимо выбрать пункт **Новый** в меню **Действия** главного меню конфигуратора на вкладке **Интерфейсы**. На экран будет выдан запрос для ввода идентификатора нового пользовательского интерфейса (рис. 5.12).

Идентификатор пользовательского интерфейса представляет собой произвольную строку символов.

Внимание!

Идентификатор нового пользовательского интерфейса не должен совпадать с идентификаторами, уже существующими в списке. Если введенный идентификатор совпадает с уже существующим, будет выдано сообщение "Результат не уникален!", тогда идентификатор необходимо исправить.

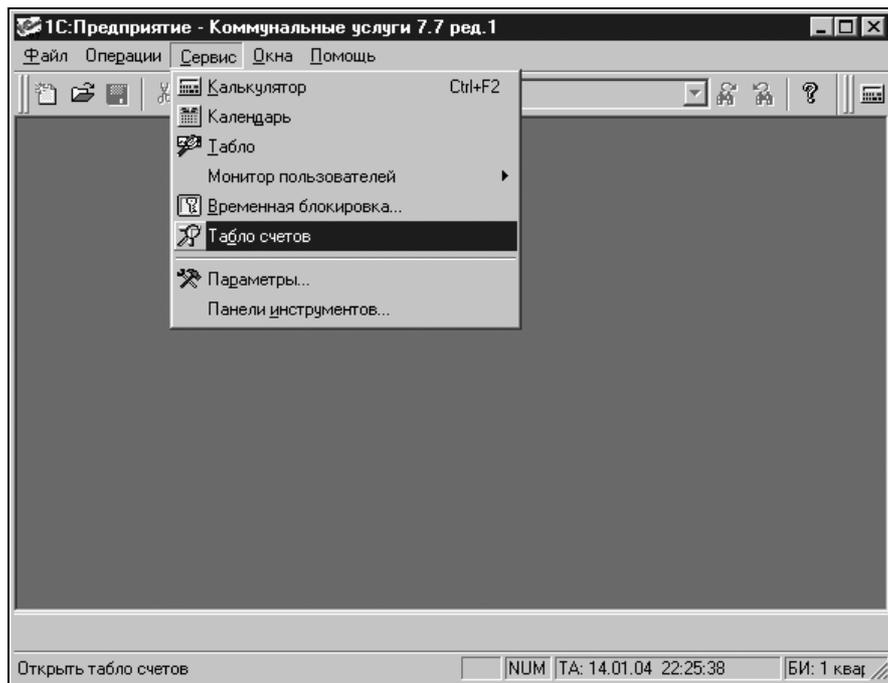


Рис. 5.11. Главное меню программы без задания интерфейса

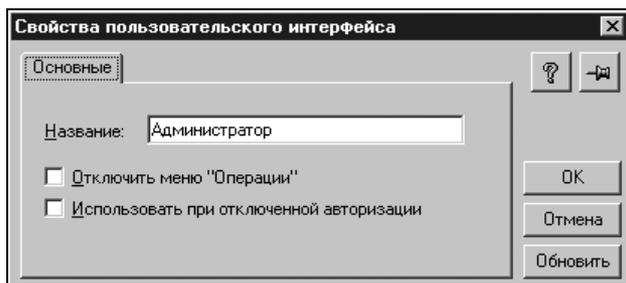


Рис. 5.12. Диалоговое окно Свойства пользовательского интерфейса

Если список пользовательских интерфейсов пуст, то при вызове на экран окна **Пользовательские интерфейсы** будет выдан запрос для ввода идентификатора нового пользовательского интерфейса.

Следует обратить особое внимание на флажок **Использовать при отключенной авторизации**. Данный флажок работает тогда, когда список пользователей пуст. При наличии нескольких интерфейсов запускается тот, у которого включен флажок **Использовать при отключенной авторизации**.

В списке интерфейс с включенным флажком отмечен красным значком. На рис. 5.13 интерфейс **Расчетчик** будет запущен, если пользователи не определены.

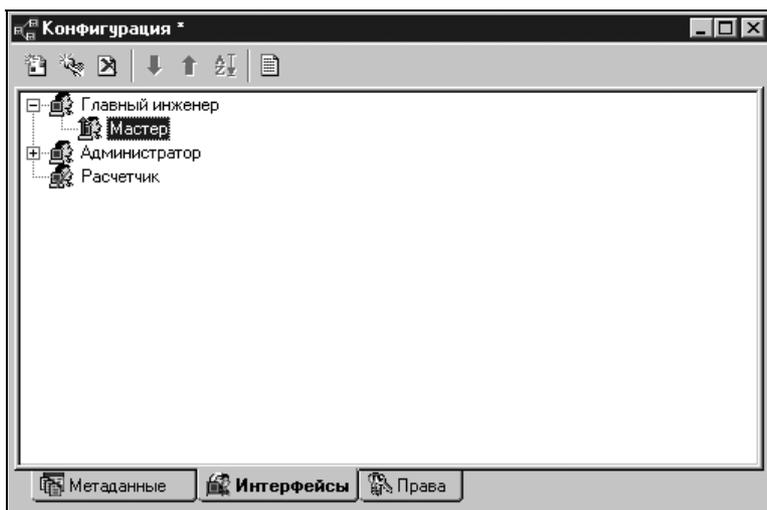


Рис. 5.13. Диалоговое окно **Конфигурация** на вкладке **Интерфейсы**

Внимание!

В случае, когда список пользователей пуст и флажок **Использовать при отключенной авторизации** не установлен ни в одном интерфейсе, запускается минимальный интерфейс.

Процесс создания меню заключается в построении структуры будущего меню и определении свойств входящих в него элементов. Сделать это можно автоматически, используя **Конструктор меню**, или вручную — отдельно создавая каждый элемент будущего меню и описывая его свойства. Редактор меню вызывается только при работе со списком пользовательских интерфейсов для редактирования пользовательского меню конкретного интерфейса. Для вызова редактора меню необходимо выполнить следующие действия:

1. Открыть окно **Конфигурация** и перейти к вкладке **Интерфейсы**.
2. В списке индивидуальных пользовательских интерфейсов выбрать название интерфейса.
3. Нажатием правой кнопки мыши открыть контекстное меню.
4. Выбрать пункт **Редактировать меню**.

Окно редактора меню содержит структуру пользовательского меню, представленную в виде дерева. Для удобства просмотра и работы дерево имеет

способность раскрываться и сжиматься. Знак (+) в узле ветви указывает, что эту ветвь можно раскрыть, для этого достаточно щелкнуть знак (+) мышью. Ветвь откроет для просмотра следующий уровень, а знак (+) изменится на (-).

На первом уровне дерева находятся меню. В редакторе меню они обозначены рисунком . Названия меню появляются в главном меню системы 1С:Предприятие. Как правило, названия меню первого уровня обозначают основные функциональные блоки конфигурации. Первоначально в новом интерфейсе системных пунктов нет, хотя в главном меню системы 1С:Предприятие они присутствуют. На последующих уровнях могут располагаться элементы меню трех типов: команды, подменю и разделители. Команда, непосредственно связанная с выполнением каких-либо действий, в редакторе меню обозначается рисунком .

Добавим в меню **Сервис** подменю **Обработка документов**. Естественно, вначале необходимо создать саму колонку **Сервис**. Для этого в строке <новая колонка...> дважды щелкнем левой кнопкой мыши. В окне **Свойства элемента меню** (рис. 5.14) установим флажок **Системная колонка**. В поле **Название** выбором из списка зададим пункт **Сервис**. Нажмем кнопку **ОК**, иначе заданные свойства не сохранятся.

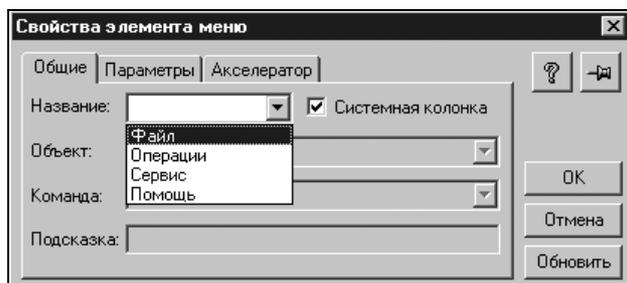


Рис. 5.14. Диалоговое окно **Свойства элемента меню** для системной колонки

Откроем в дереве меню узел **Сервис** и создадим подменю **Обработка документов**. Для этого в строке <новый...> дважды щелкнем левой кнопкой мыши. В окне **Свойства элемента меню** (рис. 5.15) в поле **Название** введем имя пункта **Обработка документов**.

В поле **Тип** зададим тип **Команда** выбором из списка:

- Команда;**
- Разделитель;**
- Подменю.**

В поле **Объект** выбором из списка всех существующих объектов метаданных зададим объект **Обработка.ОбработкаДокументов**. В поле **Подсказка**

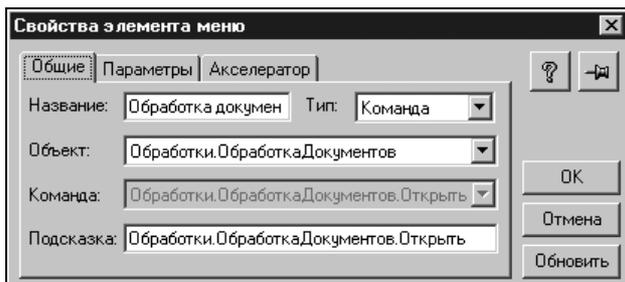


Рис. 5.15. Диалоговое окно Свойства элемента меню для обычного пункта меню

можно ввести текст подсказки, которая будет появляться в нижней строке окна программы при наведении указателя мыши на данный пункт. Следует обратить особое внимание на объект **Задача**. При задании объекта **Задача** в поле **Команда** предлагается выбрать системную обработку из списка:

- **Задача.УправлениеОперативнымииИтогами;**
- **Задача.УправлениеБухгалтерскимииИтогами;**

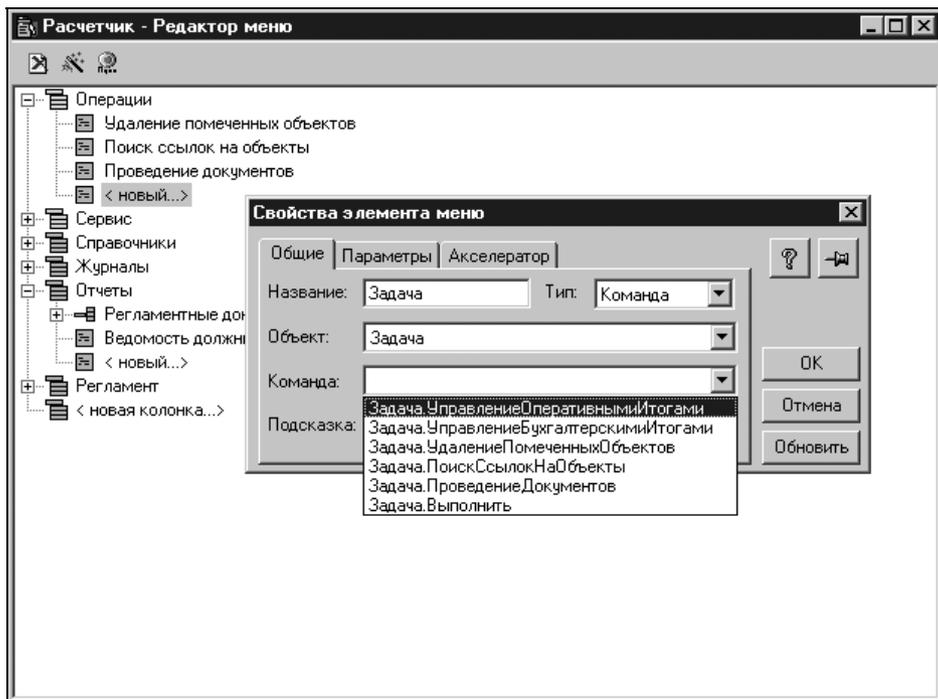


Рис. 5.16. Диалоговое окно Свойства элемента меню с объектом Задача

- **Задача.УдалениеПомеченныхОбъектов;**
- **Задача.ПоискСсылокНаОбъекты;**
- **Задача.ПроведениеДокументов;**
- **Задача.Выполнить.**

Как правило, данные обработки включают в меню **Операция** (см. рис. 5.16).

Подменю служит для дополнительной группировки элементов меню. Выбор подменю непосредственного выполнения каких-либо операций не вызывает, а открывает список, который, в свою очередь, также может состоять из команд, подменю и разделителей. В редакторе меню подменю обозначается рисунком  (рис. 5.17).

Названия команд и подменю, которые появляются в меню при его работе, также называют пунктами меню.

Разделитель не несет функциональной нагрузки и служит только для визуальной группировки логически связанных пунктов меню. В меню разделитель выглядит как горизонтальная черта между двумя пунктами меню.

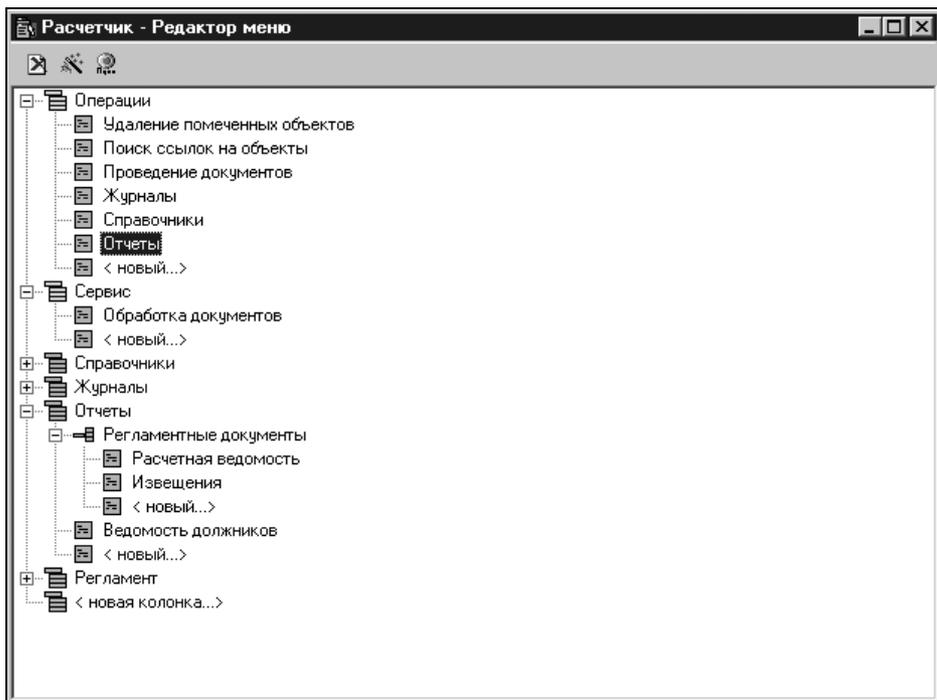


Рис. 5.17. Диалоговое окно Редактор меню

Конструктор меню

После автономной отладки возникает необходимость создавать интерфейс для большого количества объектов метаданных. Для ускорения процесса создания меню предназначен конструктор. Конструктор позволяет быстро построить пользовательское меню, содержащее необходимый набор команд. Для вызова на экран окна конструктора существует два способа:

- выбрать в окне редактора меню любой элемент созданного меню, нажав правой кнопки мыши вызвать на экран контекстное меню и выбрать пункт **Конструктор**;
- выбрать в окне редактора меню любой элемент созданного меню и выбрать пункт **Конструктор** в меню **Действия** главного меню Конфигуратора.

Меню, созданное Конструктором, полностью заменяет уже существующее меню, на экран будет выдано предупреждение об этом. Можно отказаться от использования Конструктора, нажав кнопку **Отмена**. При нажатии кнопки **ОК** на экран будет выдано окно Конструктора меню (рис. 5.18).

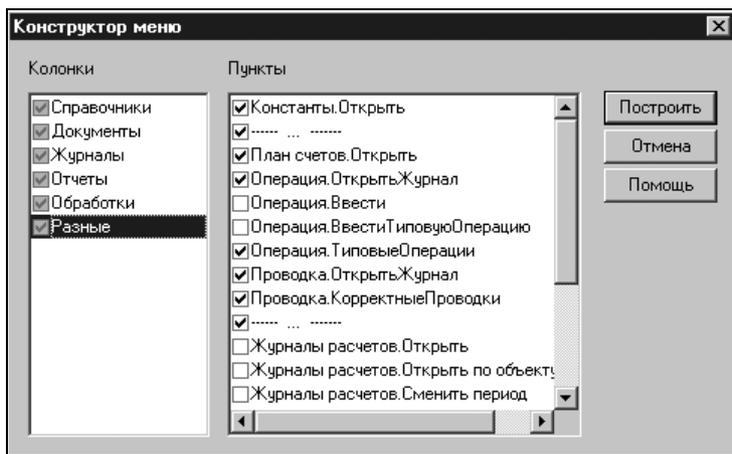


Рис. 5.18. Диалоговое окно **Конструктор меню**

Панель **Колонки** содержит список типов данных, с которыми может производиться какая-либо работа. В пользовательском меню названия типов данных будут являться названиями пунктов главного меню. Наименования типов данных можно выбирать мышью или клавишами.

Панель **Пункты** содержит список predetermined команд для работы с объектами данных выбранного типа. В пользовательском меню полные имена команд будут являться пунктами меню.

Процесс создания меню при помощи Конструктора заключается в расстановке отметок у команд, которые необходимо включить в меню. Чтобы включить название колонки в главное меню, необходимо щелкнуть левой кнопкой мыши на пустой рамке слева от наименования типа данных. Чтобы исключить название колонки из главного меню, необходимо щелкнуть левой кнопкой мыши на рамке с отметкой слева от наименования типа данных. Чтобы включить команду в меню, необходимо выполнить следующие действия:

1. Выбрать наименование типа данных в панели **Колонки**.
2. Щелкнуть левой кнопкой мыши на пустой рамке слева от имени команды.

Чтобы исключить команду из меню, необходимо выполнить следующие действия:

1. Выбрать наименование типа данных в панели **Колонки**.
2. Щелкнуть левой кнопкой мыши на рамке с отметкой слева от имени команды.

Чтобы сохранить результат работы **Конструктора меню**, необходимо нажать кнопку **Построить**.

Подчиненный интерфейс

Подчиненный интерфейс — это интерфейс, созданный по образцу и подобию интерфейса-родителя. Но в отличие от родителя, его можно редактировать только в части уменьшения пунктов меню. Чтобы создать новый подчиненный интерфейс по образцу существующего, необходимо выполнить следующие действия:

1. Выбрать интерфейс, по образцу которого будет создаваться новый интерфейс.
2. Выбрать пункт **Новый подчиненный интерфейс** в меню **Действия** главного меню конфигулятора на вкладке **Интерфейсы**.

На экран будет выдан запрос для ввода идентификатора нового пользовательского интерфейса. В списке интерфейсов подчиненный интерфейс будет отмечен красной стрелкой, указывающей на интерфейс-родитель (см. рис. 5.13).

При попытке редактирования меню подчиненного интерфейса предлагается не обычное дерево меню, а список, в котором можно только включать или отключать признаки выполнения команд меню (рис. 5.19).

С другой стороны, все изменения, сделанные в интерфейс-родителе, отражаются в подчиненном интерфейсе.

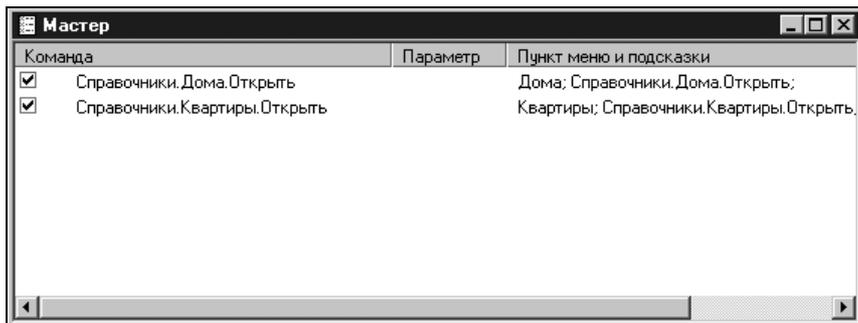


Рис. 5.19. Диалоговое окно редактирования подчиненного интерфейса

Панели инструментов интерфейса

Одной из составных частей пользовательского интерфейса являются пользовательские панели инструментов. Как правило, их настраивают на работу только с той информацией, доступ к которой разрешен набором прав пользователя. Для каждой категории пользователей можно создать свой набор панелей инструментов. Редактирование панели инструментов заключается в определении необходимого набора кнопок панели и задании свойств каждой кнопки. В процессе редактирования кнопке присваивается команда, которая будет выполняться при нажатии на эту кнопку. На кнопке может быть помещен образ — условное обозначение кнопки. Редактор панелей инструментов позволяет выбирать между графическим и текстовым образом. Для кнопки может быть заведена подсказка, кратко описывающая назначение кнопки. Помимо кнопок, в панели инструментов могут располагаться разделители. Разделители позволяют объединять кнопки на панели инструментов в логически связанные группы. В готовой панели инструментов разделитель изображается вертикальной чертой и увеличенным расстоянием между кнопками. Панели инструментов, созданные в редакторе, не могут редактироваться пользователем при работе с системой 1С:Предприятие. Пользователь только может изменить распределение кнопок по существующим панелям.

Вся работа по редактированию панелей инструментов выполняется в редакторе панелей инструментов (рис. 5.20).

Редактор панелей инструментов предназначен для создания, редактирования и настройки панелей инструментов индивидуального пользовательского интерфейса. Редактор панелей инструментов вызывается только при работе со списком пользовательских интерфейсов для редактирования панелей конкретного интерфейса. Редактор панелей инструментов позволяет создавать новые панели и удалять ненужные, а также изменять название у существующих панелей.

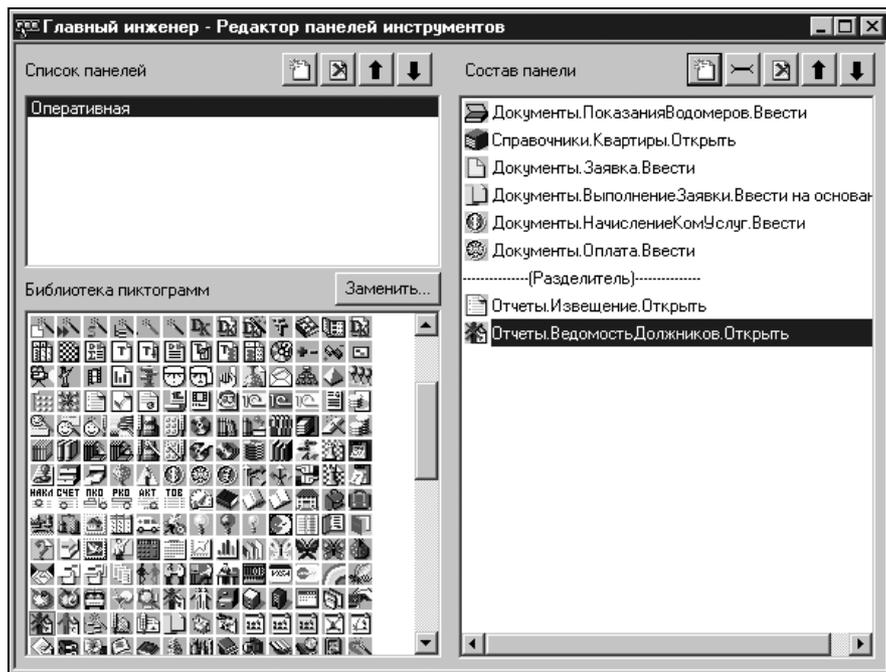


Рис. 5.20. Диалоговое окно Редактор панелей инструментов

Для вызова редактора панелей инструментов необходимо выполнить следующие действия:

1. Открыть окно **Конфигурация** на вкладке **Интерфейсы**.
2. В списке индивидуальных пользовательских интерфейсов выбрать название интерфейса.
3. Нажатием правой кнопки мыши открыть контекстное меню.
4. Выбрать пункт **Редактировать панели инструментов**.

Для создания новой панели инструментов необходимо нажать кнопку  (**Добавить**) над списком панелей и в запросе ввести название новой панели инструментов (рис. 5.21).

В поле **Положение** выбором из списка устанавливается положение панели в окне программы при первом запуске.

В поле **Показывать** устанавливается признак отображения панели при первом запуске.

Внимание!

Созданная панель инструментов наравне с другими штатными панелями инструментов подвергается манипулированию со стороны пользователя.

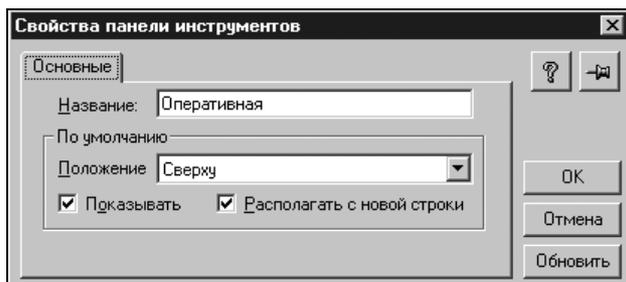


Рис. 5.21. Диалоговое окно **Свойства панели инструментов**

Двойным щелчком левой кнопки мыши на наименовании панели в списке вызывается окно свойств панели инструментов для редактирования.

Для удаления панели инструментов необходимо выполнить следующие действия:

1. В списке панелей выбрать панель инструментов, которую необходимо удалить.
2. Нажать кнопку  над списком панелей.
3. Подтвердить удаление панели инструментов.

Для создания новой кнопки необходимо выполнить следующие действия:

1. В списке панелей выделить название панели инструментов, для которой будет создаваться кнопка.
2. Нажать кнопку  над списком **Состав панели**.
3. В поле **Объект** палитры свойств выбрать объект метаданных, над которым будут производиться действия при нажатии на кнопку.
4. В поле **Команда** палитры свойств выберите команду, которая будет выполняться при нажатии на кнопку (рис. 5.22).

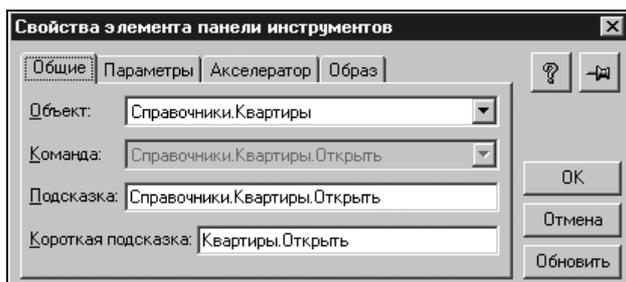


Рис. 5.22. Диалоговое окно **Свойства элемента панели инструментов**

5. Отредактировать остальные свойства созданной кнопки.
6. Нажать кнопку **ОК**, чтобы отразить внесенные изменения в списке **Состав панели**.
7. Используя кнопки  и , переместите созданную кнопку в требуемую позицию списка **Состав панели**.

Для изменения пиктограммы кнопки необходимо перетащить мышью рисунок из библиотеки пиктограмм на кнопку или дважды щелкнуть мышью на нужной пиктограмме.

Для создания разделителя необходимо выполнить следующие действия:

1. В списке панелей выделить название панели инструментов, для которой будет создаваться разделитель.
2. Нажать кнопку  (**Разделитель**) в списке **Состав панели**.
3. Используя кнопки  и , переместить созданный разделитель в требуемую позицию списка **Состав панели**.

Если перед нажатием кнопки выделить название кнопки в списке **Состав панели**, разделитель будет создан под выбранной кнопкой.

Для удаления кнопки необходимо выполнить следующие действия:

1. В списке панелей выделить название панели инструментов, из которой необходимо удалить кнопку.
2. В списке **Состав панели** выделить кнопку или разделитель, который необходимо удалить.
3. Нажать кнопку  над списком **Состав панели**.

Внимание!

Удаление выполняется без дополнительного предупреждения.

Слева внизу расположено окно с библиотекой пиктограмм — рисунков, которые могут быть размещены на кнопках создаваемой панели инструментов. Кнопка **Заменить** предназначена для смены библиотеки пиктограмм.

Права

С целью интеграции и разделения доступа пользователей к информации при работе с системой 1С:Предприятие в сети персональных компьютеров Конфигуратор системы позволяет установить для каждого пользователя права на работу с информацией, обрабатываемой системой. Права могут быть заданы

в достаточно широких пределах — от возможности только просмотра некоторых видов документов до полного набора прав по вводу, просмотру, корректировке и удалению любых видов данных.

Назначение пользователю прав доступа производится в 2 этапа. На первом этапе создаются типовые наборы прав на работу с информацией, отличающиеся, как правило, шириной предоставляемых возможностей доступа. На втором этапе пользователю ставится в соответствие один из таких типовых наборов прав.

Вся работа по созданию типовых наборов прав производится на вкладке **Права** окна **Конфигурация**. Это окно содержит список типовых наборов прав (рис. 5.23).

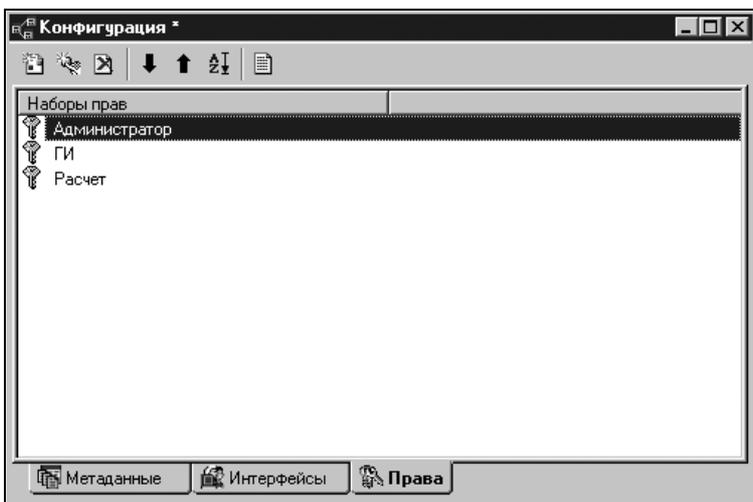


Рис. 5.23. Диалоговое окно **Конфигурация** на вкладке **Права**

Чтобы создать новый набор прав, необходимо выбрать пункт **Новый** в меню **Действия** главного меню Конфигуратора. На экран будет выдан запрос для ввода идентификатора нового набора прав (рис. 5.24).

Идентификатор набора прав представляет собой произвольную строку символов.

Внимание!

Идентификатор нового набора прав не должен совпадать с идентификаторами, уже существующими в списке.

Для ввода идентификатора нового набора прав в список необходимо нажать кнопку **ОК**. Если введенный идентификатор совпадает с уже существующим

щим — будет выдано сообщение "Результат не уникален!", тогда идентификатор необходимо будет исправить.

Внимание!

Если список наборов прав пуст, то при вызове на экран окна **Наборы прав** будет выдан запрос для ввода идентификатора нового набора прав.

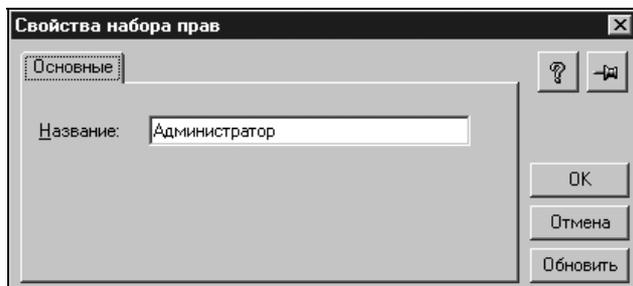


Рис. 5.24. Диалоговое окно **Свойства набора прав**

Чтобы создать новый набор прав по образцу существующего, необходимо выполнить следующие действия:

1. Выбрать в списке идентификатор набора прав, который будет служить образцом.
2. Выбрать пункт **Скопировать** в меню **Действия** главного меню конфигуризатора.
3. В выданном на экран запросе задать идентификатор нового набора прав и нажать кнопку **ОК**.

В список будет внесен идентификатор нового набора прав. Вновь созданному набору не устанавливаются никакие права.

Чтобы удалить набор прав, необходимо выполнить следующие действия:

1. Выбрать в списке идентификатор набора прав, который необходимо удалить.
2. Выбрать пункт **Удалить** в меню **Действия** главного меню конфигуризатора.

Внимание!

После выбора пункта **Удалить** указанный набор прав будет сразу удален!

Если необходимо восстановить случайно удаленный набор прав, следует закрыть окно со списком наборов прав, отказавшись от сохранения списка.

Для упорядочивания списка набора прав необходимо использовать кнопки:

-  ("Переместить вниз");
-  ("Переместить вверх");
-  ("Сортировать").

Задание набора прав, по сути, состоит в установке опций на все объекты метаданных и системные функции.

Чтобы установить все права, необходимо выполнить следующие действия:

1. Выбрать в списке идентификатор набора прав, для которого необходимо разрешить доступ ко всей информации.
2. Выбрать пункт **Установить все права** в меню **Действия** главного меню конфигулятора.

Чтобы снять все права, необходимо выполнить следующие действия:

1. Выбрать в списке идентификатор набора прав, для которого необходимо запретить доступ ко всей информации.
2. Выбрать пункт **Снять все права** в меню **Действия** главного меню конфигулятора.

Эти функции удобно использовать при создании нового набора прав — например, установить все права, затем вызвать редактор пользовательских прав и отредактировать доступ к отдельным видам информации.

Редактор пользовательских прав предназначен для создания набора прав доступа к информации, обрабатываемой системой 1С:Предприятие. Окно редактора можно вызвать на экран одним из следующих способов:

- В окне **Наборы прав** дважды щелкните левой кнопкой мыши на нужном наборе прав.
- В окне **Наборы прав** установите указатель мыши на идентификаторе набора прав, нажатием правой кнопки мыши вызовите на экран контекстное меню, в этом меню выберите функцию **Редактировать**.
- В окне **Наборы прав** установите указатель мыши на идентификаторе набора прав, в меню **Действия** главного меню конфигулятора выберите функцию **Редактировать**.

Окно редактора прав содержит представленную в виде дерева структуру данных, созданную для текущей конфигурации (рис. 5.25).

Для удобства просмотра и работы дерево имеет способность раскрываться и сжиматься. Знак (+) в узле ветви указывает, что эту ветвь можно раскрыть, для этого достаточно нажать мышью на знак. Ветвь откроет для просмотра следующий уровень, а знак (+) изменится на (-). В отличие от окна **Мета-**

данные, дерево данных в окне редактора пользовательских прав не содержит ветви **Перечисления**, т. к. этот тип данных в большей степени — служебный и используется в основном при вводе экземпляров данных других типов.

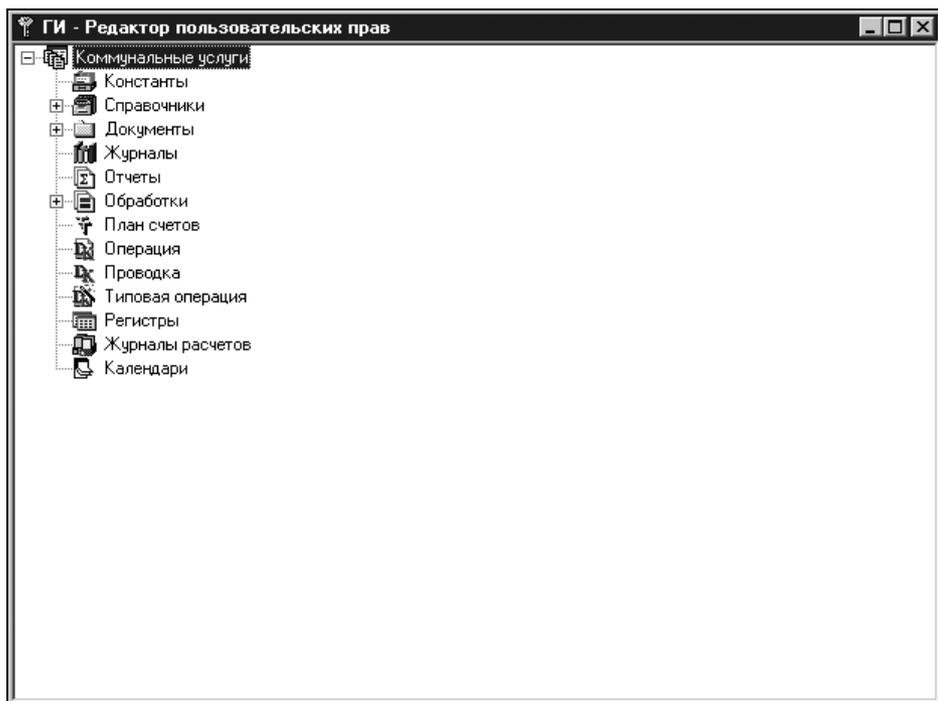


Рис. 5.25. Диалоговое окно Редактор пользовательских прав

Редактор пользовательских прав предоставляет возможность детального управления правами на работу как с типами данных в целом — документами, справочниками, журналами и т. д., так и с отдельными элементами данных этих типов.

Установка прав производится в диалоге **Свойства элемента прав**. Для вызова этого диалога можно воспользоваться одним из следующих способов:

- в окне редактора пользовательских прав установить указатель мыши на наименовании элемента данных и дважды щелкнуть левой кнопкой мыши;
- в окне редактора пользовательских прав установить указатель мыши на идентификаторе набора прав, в меню **Действия** главного меню конфигуратора выбрать пункт **Свойства**.

Диалог **Редактор пользовательских прав** организован в виде картотеки, имеющей только одну вкладку — **Доступ**. Вкладка содержит окно со спи-

ском операций, которые можно осуществлять над выбранным элементом данных, и управляющие элементы для разрешения и запрещения прав (рис. 5.26).

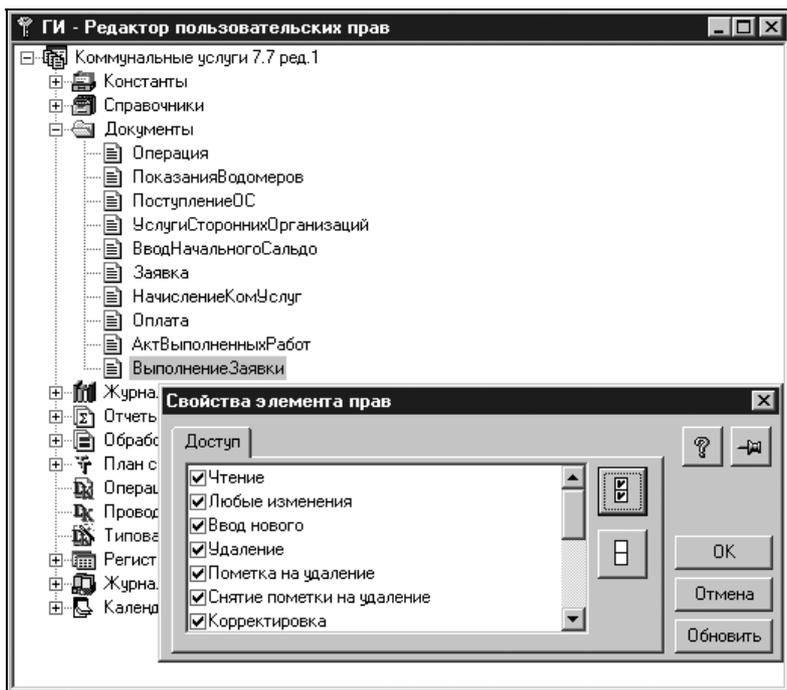


Рис. 5.26. Диалоговое окно Свойства элемента прав

Чтобы разрешить выполнение операции над элементом данных, необходимо щелкнуть левой кнопкой мыши на пустой рамке слева от наименования операции.

Чтобы запретить выполнение операции над элементом данных, необходимо щелкнуть левой кнопкой мыши на рамке с отметкой слева от наименования операции.

Разрешение той или иной операции может автоматически включать разрешение других логически связанных с ней операций, и наоборот, — выключение какой-либо операции вызывает выключение и некоторых других. Например, при включении операции **Любые изменения** автоматически включаются и все остальные операции, включение операции **Корректировка** автоматически включает и операцию **Чтение** и тому подобное. Характер подобных связей зависит от набора операций, которые можно установить для элемента данных.

Чтобы разрешить выполнение всех операций над элементом данных, необходимо нажать кнопку справа от списка операций .

Чтобы запретить выполнение всех операций над элементом данных, необходимо нажать кнопку справа от списка операций .

Внимание!

Установка прав для типа данных автоматически устанавливает такие же права и для всех экземпляров этого типа данных: ранее сделанные установки при этом теряются.

Следует обратить особое внимание на права, которые носят системный характер. Данный набор прав задается для элемента **Конфигурация** (рис. 5.27).

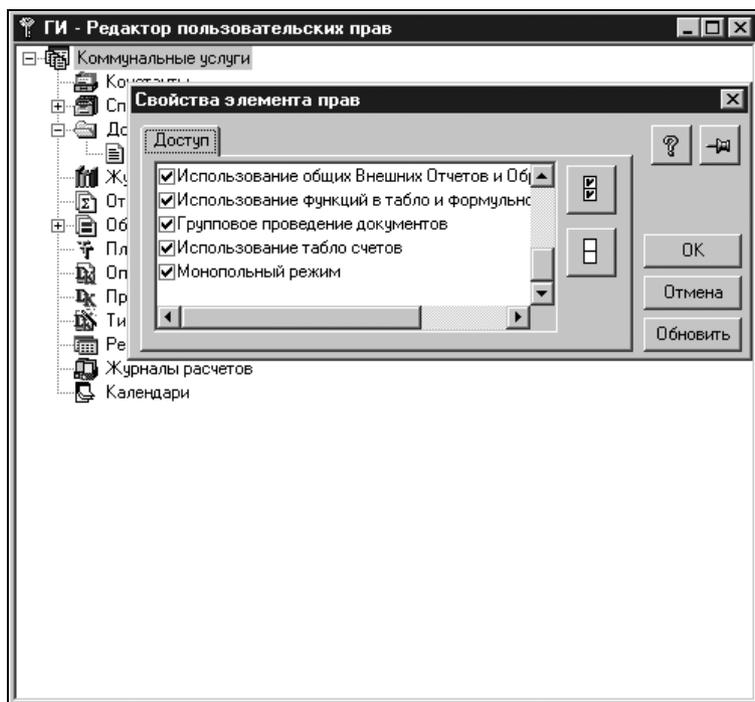


Рис. 5.27. Диалоговое окно **Свойства элемента прав** для элемента **Конфигурация**

В табл. 5.1 приведена сводка по всем правам, которые можно установить для объектов метаданных.

Таблица 5.1. Операции с объектами метаданных

Объект метаданных	Операция	Содержание операции
Конфигурация	Административные функции	Доступ к функциям Конфигуратора: изменение структуры метаданных, управление списком пользователей, назначение прав доступа, редактирование пользовательских интерфейсов
	Сохранение/Выгрузка данных	Создание архивной копии и выгрузка данных в файл переноса данных
	Управление оперативными итогами	Смена точки актуальности итогов и открытие нового периода
	Управление бухгалтерскими итогами	Расчет итогов
	Монитор	Работа с монитором пользователей
	Использование в качестве OLE Automation сервера	Доступ к данным системы 1С:Предприятие из внешних программ
	Удаление помеченных объектов	Выполнение операции физического удаления данных из информационной базы
	Поиск ссылок на объект	Поиск перекрестных ссылок между объектами данных
	Использование любых Внешних Отчетов и Обработок	Использование внешних отчетов (обработок)
	Использование общих Внешних Отчетов и Обработок	Использование внешних отчетов (обработок), расположенных в подкаталоге ExtForms каталога информационной базы
	Использование функций в табло и формульном калькуляторе	Использование в табло и формульном калькуляторе функций встроенного языка системы 1С:Предприятие
	Групповое проведение документов	Использование режима группового проведения документов и восстановления последовательностей документов
	Использование табло счетов	Использование табло счетов
	Монопольный запуск	Запуск системы 1С:Предприятие в монопольном режиме

Таблица 5.1 (продолжение)

Объект метаданных	Операция	Содержание операции
	Автообмен распределенной ИБ	Работа с режимом автообмена
Константы	Чтение	Просмотр значения константы
	Корректировка	Изменение значения константы
Справочники	Чтение	Просмотр справочника
	Любые изменения	Все операции
	Ввод нового элемента справочника	Ввод новых элементов и групп справочника
	Удаление	Удаление элементов и групп справочника
	Пометка на удаление	Пометка элементов и групп справочника на удаление
	Снятие пометки на удаление	Снятие пометки на удаление с элементов и групп справочника
	Корректировка	Редактирование элементов и групп справочника
Документы	Чтение	Просмотр документов
	Любые изменения	Все операции
	Ввод нового документа	Ввод нового документа
	Удаление	Удаление документа
	Пометка на удаление	Пометка документов на удаление
	Снятие пометки на удаление	Снятие пометки на удаление с документов
	Корректировка	Редактирование документа
	Выбор	Выбор документа
	Просмотр подчиненных документов	Просмотр документов, имеющих в основе текущий документ
	Проведение документа	Проведение документов
Изменения проведенных документов	Редактирование документов после проведения	
Изменение документов без перепроведения	Редактирование проведенных документов без повторного проведения	

Таблица 5.1 (продолжение)

Объект метаданных	Операция	Содержание операции
	Проведение документов "задним числом"	Проведение документов до точки актуальности итогов
	Редактирование архивного документа расчета	Редактирование архивного документа расчета
	Проведение архивного документа расчета	Проведение архивного документа расчета
	Редактирование операции документа	Редактирование бухгалтерской операции, сформированной документом
Журналы	Чтение	Просмотр журнала
	Просмотр полного журнала	Просмотр полного журнала
	Просмотр общего журнала	Просмотр общего журнала
Отчеты, обработки	Использование	Формирование отчета
Планы счетов	Чтение	Просмотр планов счетов
	Любые изменения	Все операции
	Ввод нового счета	Ввод нового счета
	Удаление	Удаление счета
	Пометка на удаление	Пометка счетов на удаление
	Снятие пометки на удаление	Снятие пометки на удаление со счетов
	Корректировка	Редактирование характеристик счета
Операция	Чтение	Просмотр операций
	Просмотр списка	Просмотр журнала операций
	Ввод операции без проверки проводок	Возможность отключения режима проверки корректности проводок
	Включение проводок операции	Сделать проводки операции активными
	Выключение проводок операции	Сделать проводки операции неактивными
Проводки	Просмотр списка	Просмотр журнала проводок
	Просмотр списка корректных проводок	Возможность просмотра списка корректных проводок

Таблица 5.1 (окончание)

Объект метаданных	Операция	Содержание операции
	Редактирование списка корректных проводок	Возможность редактирования списка корректных проводок
Типовые операции	Чтение	Просмотр типовых операций
	Любые изменения	Все действия
	Ввод нового	Создание типовых операций
	Удаление	Удаление типовых операций
	Корректировка	Редактирование типовых операций
	Использование	Ввод типовых операций
Регистры	Чтение	Просмотр движений регистров
Журналы расчетов	Чтение	Просмотр журнала расчетов
	Любые изменения	Редактирование журнала расчетов
	Расчет	Выполнение расчета записей журнала расчетов
	Изменение расчетного периода	Смена расчетного периода журнала расчетов
Календари	Чтение	Просмотр календаря
	Любые изменения	Редактирование календаря
	Редактировать праздники	Редактирование календаря праздничных дней

Права Администратора

Набор прав Администратора требует особого рассмотрения. Система защищена от "рассеянного" разработчика. Сохранить наборы прав не удастся, если не задать хотя бы в одном наборе прав административные функции. Под административными функциями понимается право на корректировку наборов прав. Без такой возможности система в дальнейшем не сможет определять права (заморозится в смысле раздачи прав в будущем).

Дата запрета редактирования

Следует обратить внимание также на установку права редактирования константы `ДатаЗапретаРедактирования`. Ряд пользователей, которые отвечают за корректность последовательности документов, вынуждены возвращаться в

прошлые периоды. Учитывая "продвинутость" и здоровое детское любопытство некоторых пользователей, административные функции желательно делать эксклюзивными, для того чтобы права не менялись некомпетентными пользователями.

Панели инструментов Конфигуратора

В Конфигураторе существует возможность настройки панелей инструментов для работы в нем самом. Чтобы вызвать окно редактора панелей инструментов, необходимо в меню **Сервис** выбрать пункт **Панели инструментов**.

Окно редактора панелей инструментов состоит из трех основных частей (рис. 5.28).

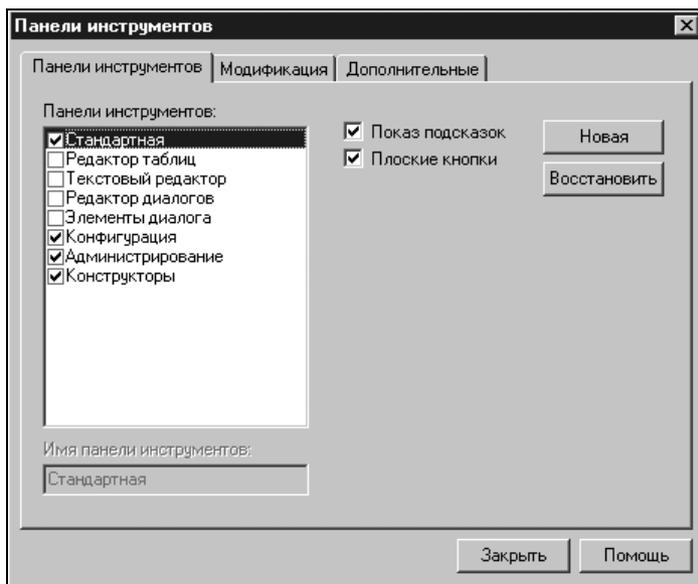


Рис. 5.28. Диалоговое окно **Панели инструментов** на вкладке **Панели инструментов**

Список панелей инструментов позволяет выбрать и настроить используемые при работе с программой инструментальные панели. Пользователь может отредактировать список существующих панелей, определить их состав и список доступных панелей, отображаемых на экране. Список панелей содержит как системные панели, так и панели, определенные пользователем. Системные панели невозможно удалить из списка, однако пользователь может отредактировать их состав.

Список **Панели инструментов** содержит все присутствующие в программе инструментальные панели.

Поле **Имя панели инструментов** позволяет изменить название панели. Для изменения названия панели выберите в списке нужную панель и введите новое название.

Нажатие кнопки **Новая** вызывает диалог для задания имени новой панели.

Нажатие кнопки **Удалить** удаляет выбранную панель из списка.

Внимание!

Кнопка **Удалить** действует только для пользовательских панелей инструментов.

Нажатие кнопки **Восстановить** для выбранной панели восстанавливает исходный состав панели до редактирования.

Внимание!

Кнопка **Восстановить** действует только для системных панелей инструментов.

Вкладка **Модификация** диалогового окна **Панели инструментов** предназначена для редактирования состава панелей (рис. 5.29).

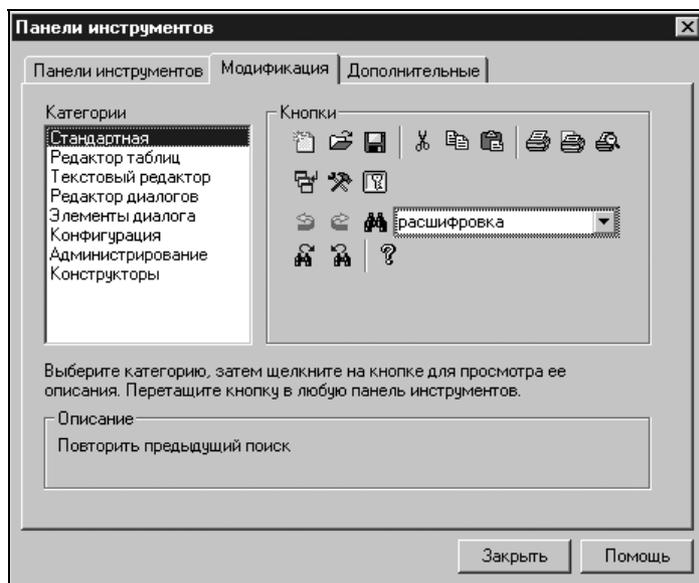


Рис. 5.29. Диалоговое окно **Панели инструментов** на вкладке **Модификация**

После выбора **Категории** в группе **Кнопки** отображаются кнопки, входящие в выбранную категорию. Для получения описания кнопки щелкните на ней

мышью. Описание выбранной кнопки отобразится в информационной строке

Описание.

Для создания новой панели необходимо выполнить следующие действия:

1. Выбрать категорию в списке **Категории**.
2. Перенести мышью нужную кнопку в любое место окна программы.

Будет автоматически создана новая панель инструментов с условным именем **UserToolBar1** (цифра может изменяться в зависимости от количества подобных панелей инструментов). Автоматически присвоенные имена можно впоследствии изменить, используя вкладку **Панели инструментов**.

Чтобы добавить кнопку на панель инструментов, необходимо перетащить ее мышью на нужную панель инструментов. Можно добавлять кнопки, перетаскивая их между панелями инструментов.

Для удаления кнопки с панели инструментов, необходимо перетащить ее мышью в любое место окна программы.

Между двумя любыми кнопками панели инструментов можно вставить разделитель. Для этого нужно "ухватить" мышью одну из двух кнопок, между которыми нужно вставить разделитель, немного отодвинуть ее от другой кнопки (примерно на четверть ширины) и отпустить. Между кнопками будет помещен разделитель. Для удаления разделителя достаточно "придвинуть" мышью одну из кнопок, между которыми есть разделитель, к другой, и отпустить ее — разделитель будет удален.

Вкладка **Дополнительные** диалога **Панели инструментов** позволяет управлять показом панелей инструментов (рис. 5.30).

Список **Инструментальные панели окон** позволяет указать расположение "оконных" панелей инструментов. Можно выбрать расположение панелей вдоль одной из границ окна или совсем отключить эти панели.

Опция **Единая настройка для всех режимов работы** отключает смену состава панелей инструментов при смене режима работы программы. Набор панелей инструментов, присутствующих на экране, постоянно изменяется в зависимости от режима работы. Например, при работе с редактором текстов на экране появится панель инструментов **Редактор текстов**. При работе с табличным редактором панель **Редактор текстов**, естественно, будет убрана, а на экран будут выданы панели **Редактора таблиц**. Первоначальный состав панелей инструментов, выдаваемых на экран в том или ином режиме работы, определяется на уровне системы.

При помощи контекстного меню панели инструментов (рис. 5.31) или вкладки **Панели инструментов** можно изменить состав присутствующих на

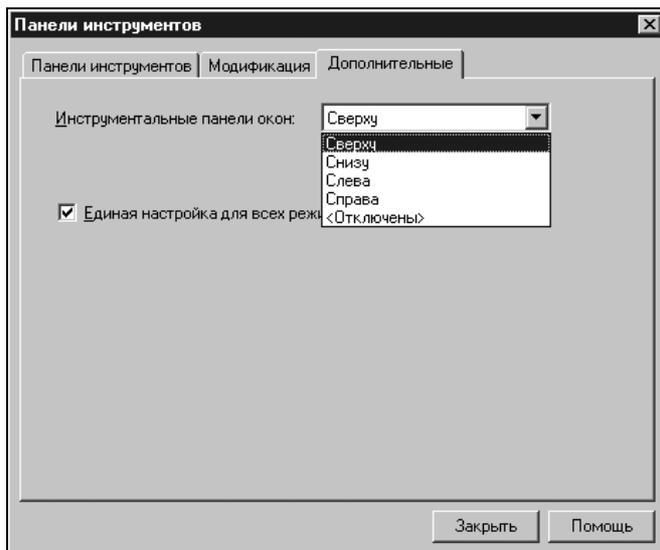


Рис. 5.30. Диалоговое окно Панели инструментов на вкладке **Дополнительные**

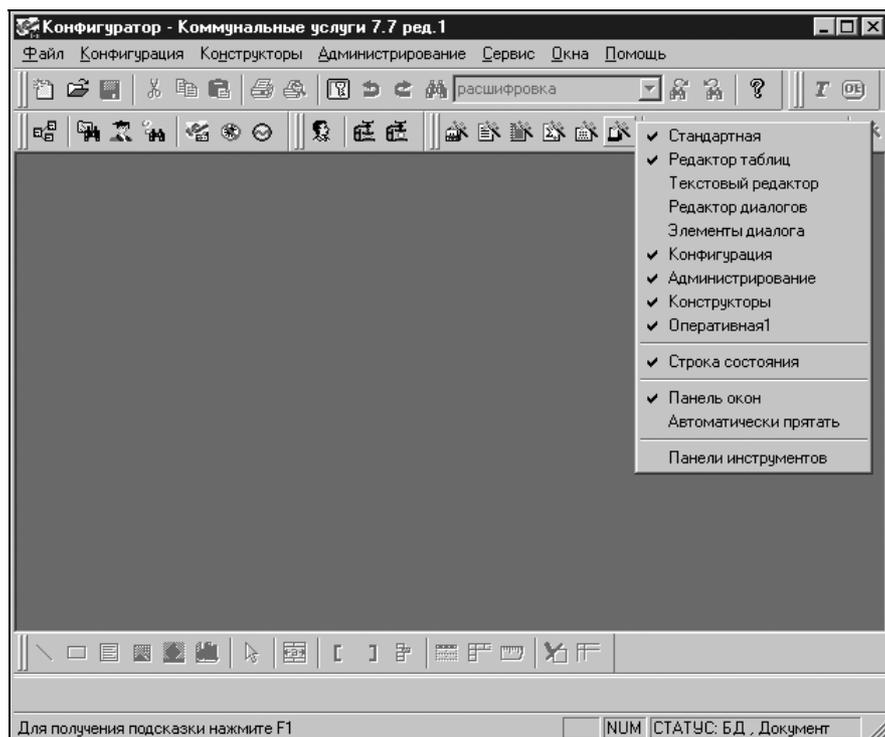


Рис. 5.31. Контекстное меню панели инструментов

экране панелей. Выполненная настройка запоминается только для текущего режима работы — при его смене состав панелей на экране также изменится.

Если опция **Единая настройка для всех режимов работы** включена, то на экране вне зависимости от режима работы постоянно будет присутствовать один набор панелей инструментов. Этот набор можно редактировать, однако для всех режимов он всегда будет одинаковым.

Настройка параметров

Настройка параметров Конфигуратора производится в диалоге **Настройка параметров системы**.

Для вызова диалога на экран необходимо выбрать пункт **Параметры** в меню **Сервис** главного меню Конфигуратора.

Вкладка **Модули** позволяет настроить параметры редактора текстов, используемые при редактировании модулей (рис. 5.32).

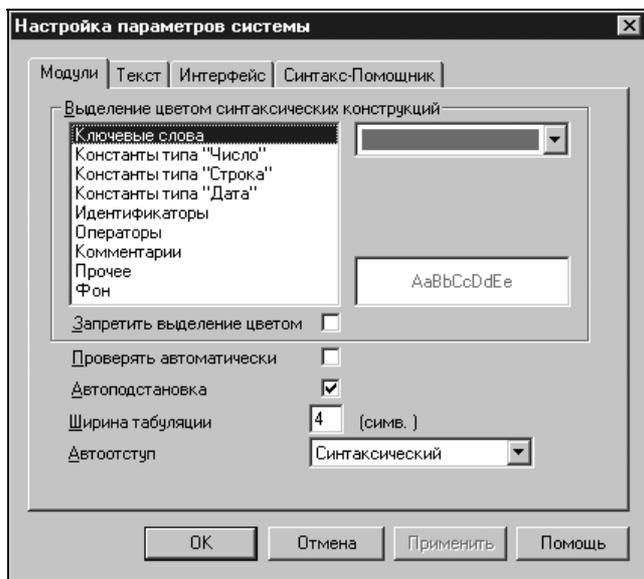


Рис. 5.32. Диалоговое окно **Настройка параметров системы** на вкладке **Модули**

В списке в левом верхнем углу представлены типы элементов встроенного языка системы 1С:Предприятие, для которых можно настроить цвет.

Палитра цветов позволяет назначить цвет для выбранного в списке типа элементов встроенного языка. Если указано слово "Авто", то цвет синтаксического элемента будет определяться установками Windows.

Если опция **Запретить выделение цветом** включена, отменяется выделение цветом синтаксических конструкций в модулях. Назначенные ранее цвета при этом сохраняются. Отключение данной опции восстанавливает цветовую палитру модулей.

Если включена опция **Автоподстановка**, при редактировании модулей будет задействована функция автоматической замены шаблонов.

Поле **Ширина табуляции** — целое число от 1 до 16 включительно. Назначает в символах ширину шага табуляции, который будет использоваться при редактировании модулей.

Поле **Автоотступ** определяет стиль автоматического форматирования управляющих конструкций встроеного языка при редактировании модулей:

- Синтаксический** — выполняет форматирование модуля, смещая вправо текст, расположенный внутри управляющих конструкций типа **Если...Тогда...КонецЕсли**, **Пока...Цикл...КонецЦикла** и подобных. Смещение выполняется за счет добавления в начало строк необходимого количества знаков табуляции;
- Включить** — "обыкновенный" отступ выравнивает текст строки по левой границе предыдущей не пустой строки;
- Отключить** — форматирование текста модуля выполняться не будет.

Управляющие элементы вкладки **Текст** позволяют настроить параметры редактора текстов, используемые при редактировании текстовых документов (рис. 5.33).

Палитра **Цвет закладок** задает цвет, который используется для пометки строк в редакторе текстов.

Если опция **Открывать текст как модуль** включена, то при создании нового или открытии существующего текстового документа редактор текстов будет использовать специфические возможности, применяемые при редактировании модулей: выделение цветом синтаксических конструкций, использование синтаксического автоотступа и др.

Если опция **Перетаскивание текста** включена, в редакторе текстов для перемещения и копирования блоков текста можно использовать режим "drag & drop" ("перенеси и оставь").

Если опция **Автоподстановка** включена, при редактировании текста будет задействована функция автоматической замены шаблонов.

Поле **Ширина табуляции** — целое число от 1 до 16 включительно. Назначает ширину шага табуляции в символах, который будет использоваться при редактировании текстовых документов.

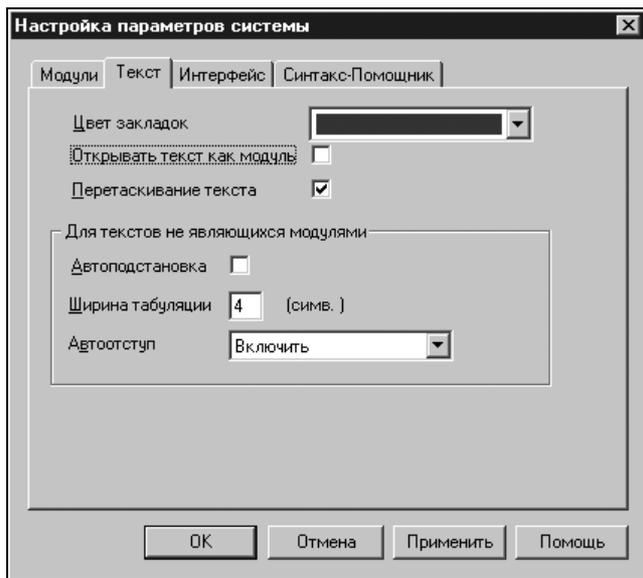


Рис. 5.33. Диалоговое окно **Настройка параметров системы** на вкладке **Текст**

Если **Автоотступ** включен, то при редактировании текстов редактор системы 1С:Предприятие автоматически выравнивает текст новой строки по левой границе предыдущей не пустой строки.

Вкладка **Интерфейс** позволяет настроить используемые шрифты и управляет показом пиктограмм в меню (рис. 5.34).

Шрифты — список режимов системы 1С:Предприятие, для которых можно указать использование определенного шрифта.

Строки в списке режимов программы имеют следующий смысл:

- Формы** — все экранные формы: стандартные и созданные в процессе настройки конфигурации задачи;
- Табличные документы** — документы, созданные при помощи функции **Новый** — **Таблица** из меню **Файл**, таблицы в редакторе форм;
- Текстовые документы** — документы, созданные при помощи функции **Новый** — **Текст** из меню **Файл**;
- Модули** — программные модули системы 1С:Предприятие;
- Табло** — информация в окне **Табло**;
- Окно сообщений** — сообщения в окне сообщений;
- Описание** — текст темы в окне описания.

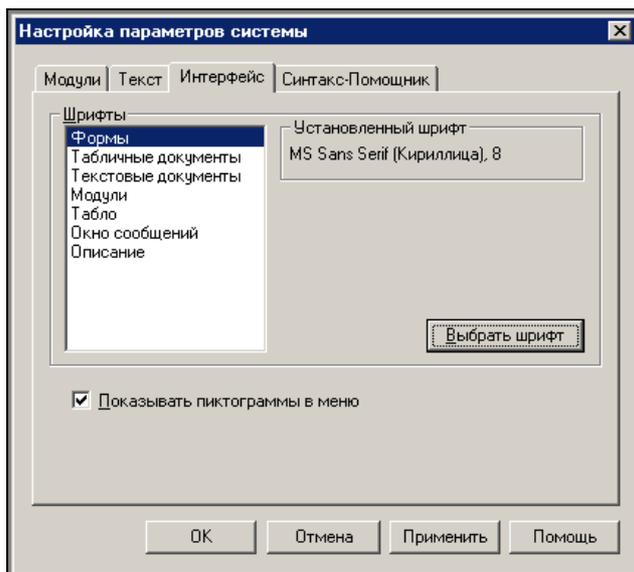


Рис. 5.34. Диалоговое окно **Настройка параметров системы** на вкладке **Интерфейс**

Установленный шрифт — текущий шрифт для выбранного в списке **Шрифты** режима.

Нажатие кнопки **Выбрать шрифт** вызывает стандартный диалог настройки шрифта (рис. 5.35).

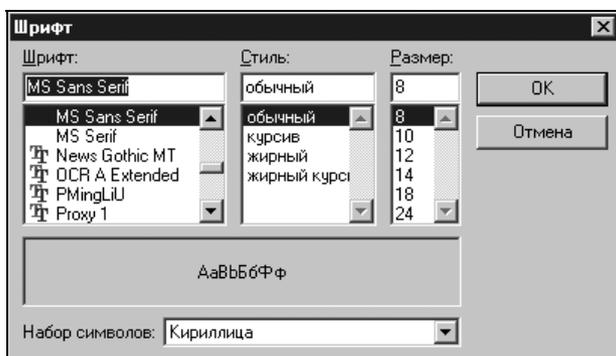


Рис. 5.35. Стандартное окно **Шрифт**

Опция **Показывать пиктограммы в меню** включает показ в главном и контекстных меню образов команд меню. Аналогичные образы используются на кнопках панелей инструментов.

Вкладка **Синтакс-Помощник** позволяет указать имя файла шаблонов и выбрать язык, на котором будут выдаваться элементы встроенного языка в окне **Синтакс-Помощника** и создаваться "заготовки" элементов встроенного языка в редакторе текстов (рис. 5.36).

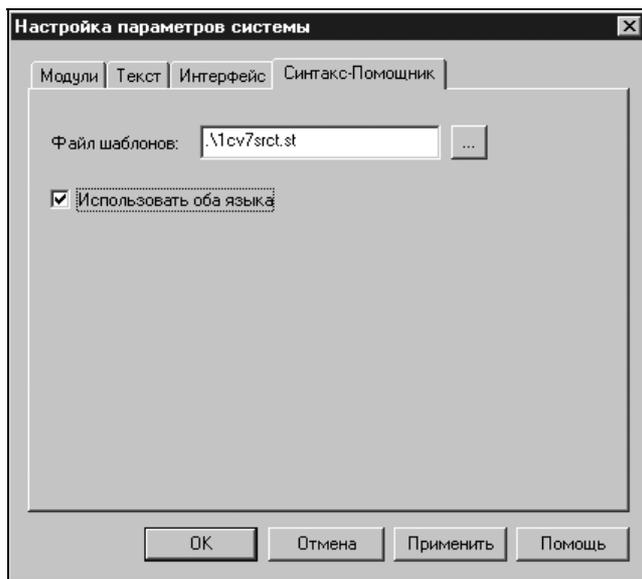
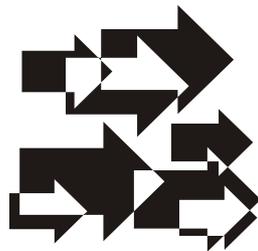


Рис. 5.36. Диалоговое окно **Настройка параметров системы** на вкладке **Синтакс-Помощник**

Список шаблонов сохраняется в файле с зарезервированным именем 1cv7srct.st и по умолчанию помещается в каталог информационной базы. В поле **Файл шаблонов** можно указать другое имя и путь к файлу шаблонов.

Если флажок **Использовать оба языка** установлен, названия элементов встроенного языка в верхнем окне **Синтакс-Помощника** будут выдаваться по-русски и по-английски. Описание элементов встроенного языка в нижнем окне **Синтакс-Помощника** всегда выдается по-русски.

ГЛАВА 6



Взаимодействие конфигураций

Восстановление базы данных

В главе 6 речь пойдет о приемах восстановления базы данных в тех случаях, когда не было несанкционированных вмешательств со стороны "дурака", т. е. экспериментальных перемещений файлов базы данных. База данных "портится", в основном, по причине аварийного выхода из программы на одной рабочей станции в режиме разделенного доступа. Ничего страшного не происходит, если сразу переиндексировать базу данных. Возможно, ничего не произойдет, если переиндексировать базу позже. Все зависит от того, в какой момент произошел аварийный выход из программы. При нестабильной работе локальной сети такие ситуации возникают чаще, следовательно, вероятность возникновения нарушения целостности базы данных выше.

Некорректность бухгалтерских итогов

Признаком "порчи" бухгалтерских итогов является то обстоятельство, что входящее сальдо по счету в отчете "Оборотно-сальдовая ведомость" не совпадает с исходящим сальдо за предыдущий период. Не пытайтесь "починить" таблицу бухгалтерских итогов непосредственно в файле. Скорее всего, вы все окончательно испортите. Для восстановления таблицы бухгалтерских итогов необходимо выполнить следующие действия:

1. В меню программы "1С:Предприятие" выбрать пункт **Операции**, затем **Управление бухгалтерскими итогами** (рис. 6.1).
2. В диалоговом окне **Управление бухгалтерскими итогами** в поле **по задать период**, предшествующий началу работы бухгалтерии в программе, т. е. период, предшествующий вводу начальных остатков (рис. 6.2).

3. Нажмите на кнопку **Установить расчет**. После смены периода в поле **по** нажмите снова указанную кнопку и так до тех пор, пока не восстановится текущий период в поле **Расчет итогов установлен по**.

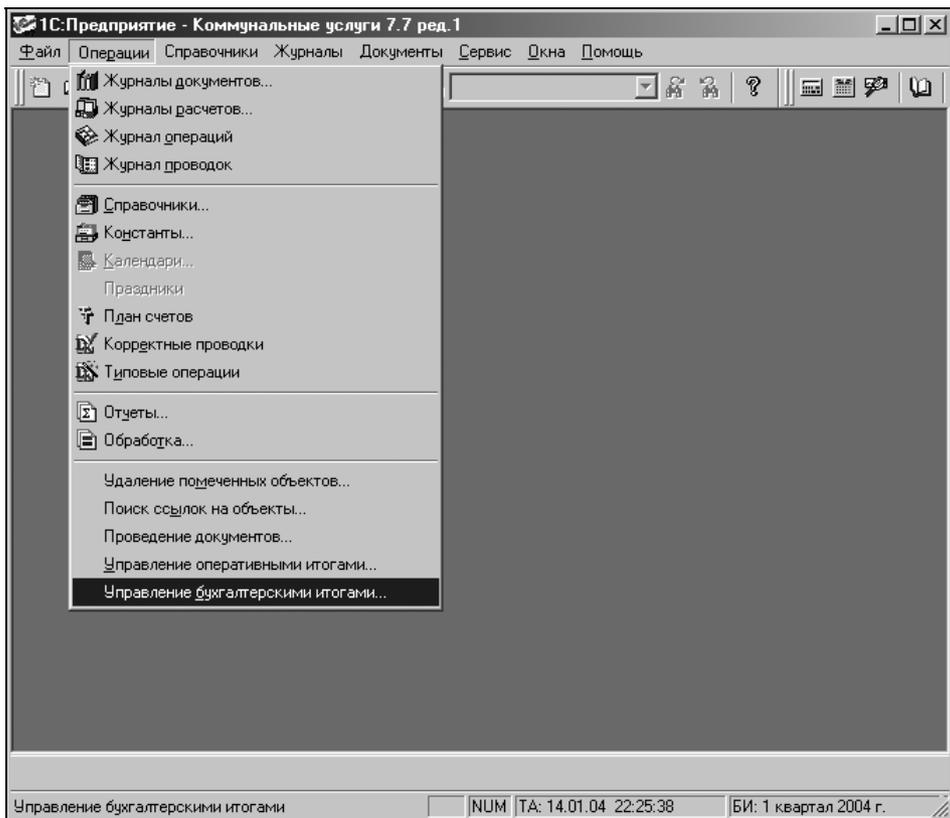


Рис. 6.1. Меню Операции

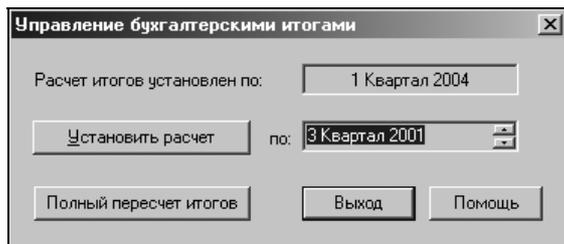


Рис. 6.2. Диалоговое окно Управление бухгалтерскими итогами

Внимание!

Простое нажатие на кнопку **Полный пересчет итогов** к восстановлению таблицы бухгалтерских итогов не приведет.

Тестирование и исправление информационной базы

Возникновение документов и операций без номеров или дат создания, исчезновение документов с присутствием их операций, исчезновение ссылок на объекты и т. п. свидетельствует о нарушении целостности базы данных и требует ремонта. Иногда нарушения носят фатальный характер, в таких случаях информационная база (ИБ) не загружается. Для устранения подобных ситуаций в Конфигураторе предусмотрен режим восстановления ИБ.

Для запуска режима восстановления необходимо выполнить следующие действия:

1. В меню Конфигуратора выбрать пункт **Администрирование**, затем пункт **Тестирование** и исправление информационной базы (рис. 6.3).

Диалог **Тестирование и исправление информационной базы** служит для установки параметров тестирования информационной базы: выбора этапов тестирования и выбора режима (рис. 6.4).

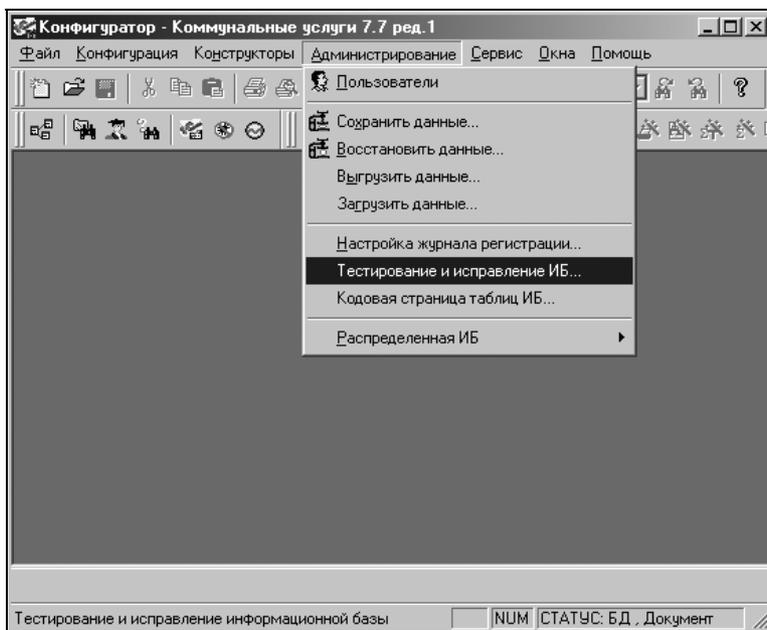


Рис. 6.3. Меню Администрирование

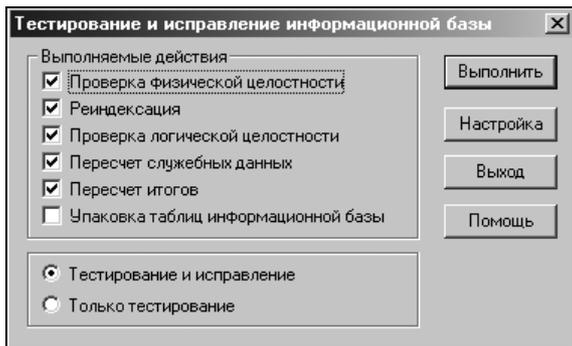


Рис. 6.4. Диалоговое окно Тестирование и исправление информационной базы

2. Задать параметры тестирования и исправления информационной базы. Рекомендуется перед исправлением информационной базы провести ее тестирование, на основе которого выбирать способы исправления ИБ, которые задаются в диалоговом окне **Настройка исправления информационной базы** (рис. 6.5) по нажатию кнопки **Настройка**. По умолчанию установлены все этапы тестирования, кроме упаковки таблиц информационной базы. Установки можно изменить. Этапы тестирования можно производить независимо друг от друга, устанавливая любые флажки. Правда, если восстановление физической целостности базы нарушило ее индексную структуру, реиндексация будет проведена независимо от того, установлен ли соответствующий флажок. Если предполагается провести только тестирование информационной базы, то выбирается опция **Только тестирование**, если же требуется еще и исправление ИБ, то отмечается опция **Тестирование и исправление**.
3. Задать способы исправления информационной базы в диалоговом окне **Настройка исправления информационной базы** по нажатию кнопки **Настройка**.

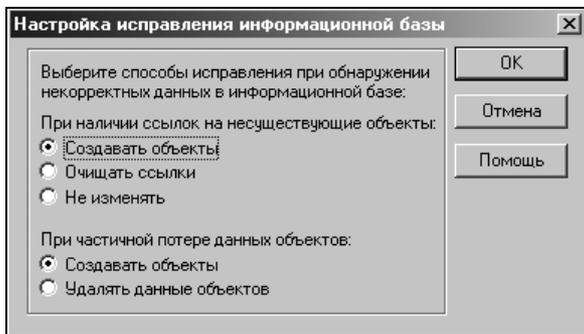


Рис. 6.5. Диалоговое окно Настройка исправления информационной базы

Группа переключателей **При наличии ссылок на несуществующие объекты** определяет, должна ли программа при обнаружении ссылок на несуществующие объекты создавать эти объекты или удалять ссылки. Можно также отказаться от исправлений ошибок этого типа.

Группа переключателей **При частичной потере данных объектов** задает поведение программы, если было обнаружено, что данные объектов были частично потеряны, но оставшихся данных достаточно для того, чтобы восстановить потерянные.

Ошибка запуска Конфигуратора

Ошибка запуска типа RunTime Error возникает по нескольким причинам. Иногда перезагрузка системы приводит к устранению ошибки. При нестабильной работе локальной сети иногда возникает ситуация, когда режим блокировки файла не снимается, атрибуты доступа не соответствуют допустимым параметрам запуска. В таких случаях не запускается даже Конфигуратор. Для устранения данной неисправности необходимо скопировать весь каталог базы данных в другое место и переназначить путь к нему. Как правило, атрибуты доступа при копировании восстанавливаются.

Ошибка открытия файлов

Довольно распространенной является ошибка открытия файла. При всех внешних признаках "благополучия" файла, он все же не открывается. Причиной этого является недостаток ресурсов компьютера, на котором расположена ИБ. Имя этого ресурса — количество одновременно открытых файлов. Чаще всего это происходит в системе Windows 98, у которой данный ресурс слабый. Замена системы на Windows 2000 Server устраняет ошибку, т. к. данный ресурс в ней в два раза выше.

Обновление конфигурации

Для осуществления обновлений конфигурации на другой релиз с сохранением данных в информационной базе рекомендуется режим объединения конфигураций, в которой программист может принимать решения по замене старых объектов на новые. Главным достоинством данного режима является то, что после анализа отчета по обновлению можно внести корректировки или вовсе отказаться от объединения.

Для обновления конфигурации в режиме объединения необходимо выполнить следующие действия:

1. В меню Конфигуратора выбрать пункт **Конфигурация**, затем пункт **Объединение конфигураций** (рис. 6.6).

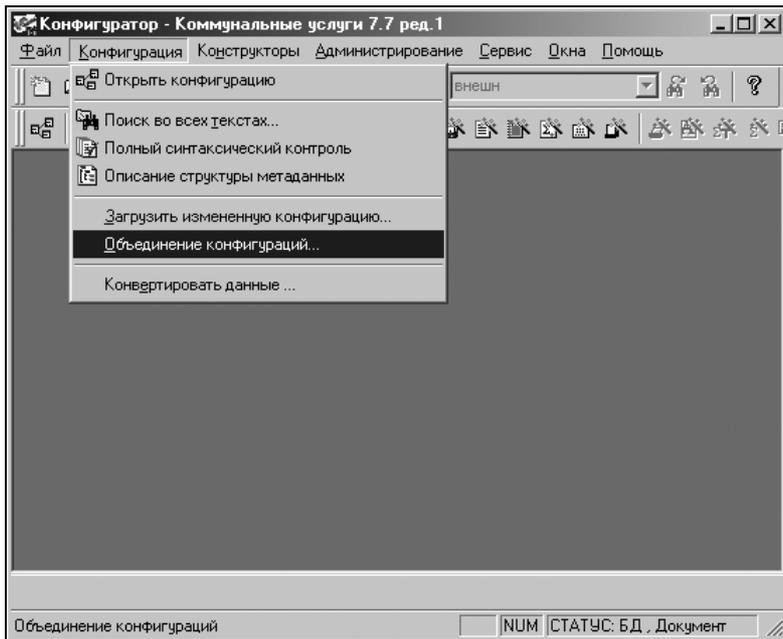


Рис. 6.6. Меню Конфигурация

2. Задать в диалоговом окне **Открыть файл конфигурации** файл конфигурации, с которой следует объединить текущую конфигурацию, 1cv7.md (рис. 6.7).

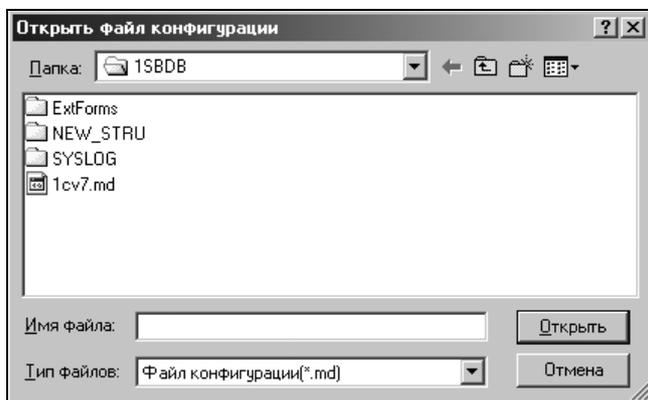


Рис. 6.7. Диалоговое окно выбора файла конфигурации для объединения

3. В окне **Объединение конфигураций** задать признаки объектам, которые будут участвовать в обновлении (рис. 6.8).

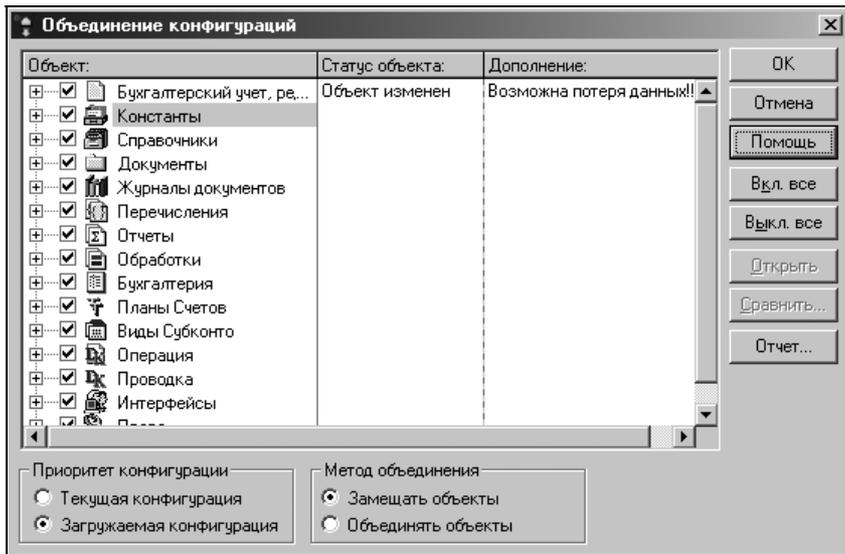


Рис. 6.8. Диалоговое окно **Объединение конфигураций**

В данном окне показаны объекты метаданных, по которым найдены какие-либо изменения. Установленный флажок показывает, что данные объекты двух конфигураций будут объединяться. Те объекты, в строке которых текст написан серым, не могут быть самостоятельно включены в объединение конфигураций, т. к. являются элементами других (агрегатных) объектов. Элементы агрегатного объекта могут быть включены или выключены вместе с самим объектом. Колонка **Статус объекта** может для конкретного объекта содержать указание **Объект изменен**, **Объект добавлен** или не содержать никакого указания. Последнее означает, что различий непосредственно по данному объекту метаданных в сравниваемых конфигурациях не найдено. Это означает, что данный объект — агрегатный, и что изменения найдены для какого-то из составляющих его элементов. Чтобы обнаружить измененный или добавленный элемент, нужно развернуть соответствующую ветвь дерева. Колонка **Дополнение** для некоторых объектов содержит сообщение "Возможна потеря данных!!!". Оно выдается, если есть вероятность, что изменения могут привести к потере данных. (При этом имеется в виду потеря данных в текущей конфигурации, а не в загружаемой).

4. Задать метод объединения. При выборе метода **Замещать объекты** объект метаданных будет добавлен, если он новый; или замещен, если он измененный. Причем переносится вся структура объекта, модули, описания, формы. При замещении сохраняется объект той конфигурации, у которой

установлен приоритет. При выборе метода **Объединять объекты** элементы объектов, имеющиеся только в одной из объединяемых конфигураций, сохраняются в объединенной конфигурации. Отличающиеся элементы выбираются в зависимости от установленного приоритета.

5. Задать приоритет конфигурации. **Текущая конфигурация** выбирается, если из загружаемой конфигурации необходимо взять только новое и максимально сохранить старое. Отличающиеся элементы добавляются в виде комментария (текст) или с измененным названием (таблица). При выборе приоритета **Загружаемая конфигурация** отличающиеся элементы текущей конфигурации превращаются в комментарий (текст) или им изменяется название (таблица).
6. Просмотреть изменения в отчете, сформированном с помощью кнопки **Отчет**, скорректировать в случае необходимости установки.
7. Запустить процесс объединения нажатием кнопки **ОК**.
8. После объединения конфигураций новая конфигурация еще не записана. Перед сохранением изменений будет предложено последнее предупреждение для принятия решения (рис. 6.9).

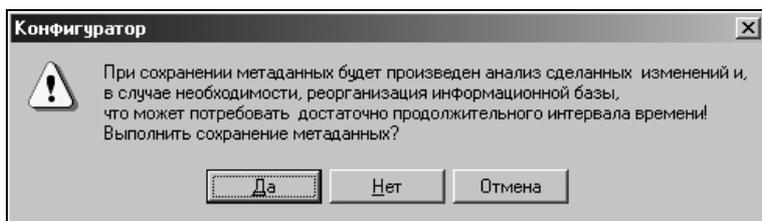


Рис. 6.9. Предупреждение перед записью изменений конфигурации

Методические рекомендации по модификации конфигурации

Обновление релиза конфигурации приводит к потере ранее разработанных изменений для тех объектов метаданных, которые предусмотрены в штатном составе конфигурации. На первый взгляд, самое простое решение проблемы — скопировать объект метаданных, переименовать его, модифицировать и включить в соответствующий журнал и пункт меню. Такие объекты обновлением не затрагиваются. Но не все так просто на самом деле. Во многих модулях ведется анализ вида документа, не говоря уже о предназначении справочника. Поскольку наш новый вид в штатной конфигурации не предусмотрен, то и из анализа он, грубо говоря, выпадает. Можно вставить в анализ

вида документа вновь созданный документ, но тогда придется следить за обновлением уже этого объекта. Можно отследить всю цепочку связанных таким образом объектов, но что делать с Глобальным модулем?

Внимание!

Категорически не рекомендуется отказываться от обновления глобального модуля. Изменения в служебных функциях Глобального модуля не описываются в инструкциях по обновлению, а используются они повсеместно.

Таким образом, данная методика хороша при условии, что вновь созданный документ не должен участвовать в анализе вида документа в Глобальном модуле. Например, при заполнении счета-фактуры на основании другого документа ведется анализ:

```
Если ДокОсн.Вид()="ПередачаОС" Тогда
    ...
ИначеЕсли ДокОсн.Вид()="ПередачаНМА" Тогда
    ...
ИначеЕсли ДокОсн.Вид()="ОтпускМатериаловНаСторону" Тогда
    ...
ИначеЕсли (ДокОсн.Вид()="ОказаниеУслуг") или
    (ДокОсн.Вид()="РасходнаяНакладная") или
    (ДокОсн.Вид()="РеализацияОтгруженнойПродукции") или
    (ДокОсн.Вид()="ВыполнениеЭтапаРабот") Тогда
    ...
ИначеЕсли (ДокОсн.Вид() = "Выписка") или (ДокОсн.Вид() =
"ПриходныйОрдер") Тогда
    ...
КонецЕсли;
```

А вот и "базовая" функция в Глобальном модуле. Здесь **Конт** — контекст вызывающей формы.

```
Процедура глДляЗаполненияКнигиПокупок(Конт) Экспорт
    Если Конт.Договор.ОплатаДоговора = 2 Тогда
        Возврат;
    КонецЕсли;
    БезНДС20=0;
    БезНДС10=0;
    НДС20=0;
    НДС10=0;
    НДС0=0;
    Освобождаемые=0;
    Если (Конт.Вид() = "УслугиСтороннихОрганизаций") или
        (Конт.Вид() = "ПоступлениеТоваров") или
```

(Конт.Вид() = "ПоступлениеМатериалов") **или**
(Конт.Вид() = "ПоступлениеОборудования") **или**
(Конт.Вид() = "ПоступлениеОС") **или**
(Конт.Вид() = "ПоступлениеНМА") **Тогда**

Если Конт.НДСвключатьВСтоимость = 1 **Тогда**

Возврат;

КонецЕсли;

Курс = ?(Конт.Курс=0,1,Конт.Курс);

Всего = (Конт.Итог("Всего") - Конт.Итог("НП"))*Курс;

Конт.ВыбратьСтроки();

Пока Конт.ПолучитьСтроку() = 1 **Цикл**

Если Конт.Вид() = "ПоступлениеТоваров" **Тогда**

Если Конт.Товар.ТипТовара =
 Перечисление.ТипыТоваров.НаКомиссии **Тогда**

 Продолжить;

КонецЕсли;

КонецЕсли;

СуммаБезНалогов = (Конт.Всего - Конт.НДС - Конт.НП)*Курс;

Если СуммаБезНалогов > 0 **Тогда**

 СтавкаНДС = 100*Конт.НДС/(Конт.Всего - Конт.НДС - Конт.НП);

Иначе

 СтавкаНДС = 20;

КонецЕсли;

Если СтавкаНДС>10.5 **Тогда**

 НДС20=НДС20+Конт.НДС*Курс;

 БезНДС20=БезНДС20+СуммаБезНалогов;

ИначеЕсли СтавкаНДС>0 **Тогда**

 НДС10=НДС10+Конт.НДС*Курс;

 БезНДС10=БезНДС10+СуммаБезНалогов;

Иначе

 Освобождаемые=Освобождаемые+СуммаБезНалогов;

КонецЕсли;

КонецЦикла;

ИначеЕсли Конт.Вид() = "СчетФактураПолученный" **Тогда**

Курс = ?(Конт.Курс=0,1,Конт.Курс);

Всего = Конт.Всего*Курс;

БезНДС20=Конт.СуммаБезНДС20*Курс;

БезНДС10=Конт.СуммаБезНДС10*Курс;

НДС20=Конт.НДС20*Курс;

НДС10=Конт.НДС10*Курс;

Освобождаемые=Конт.СуммаСовсемБезНДС*Курс;

Если Конт.НДСпоСтавкеНольПроцентов = 1 **Тогда**

 НДС10 = 0;

 НДС20 = 0;

БезНДС10 = 0;

БезНДС20 = 0;

НДС0 = Всего;

КонецЕсли;

КонецЕсли;

Если БезНДС20>0 **Тогда**

Конт.Операция.НоваяПроводка ();

Конт.Операция.НомерЖурнала = "АВ";

Конт.Операция.ПервичныйДокумент = глПредставлениеПервичногоДокумента
(Конт);

Конт.Операция.Кредит.Счет = СчетПоКоду ("ЗПК.20.Б");

Конт.Операция.Кредит.Субконто (1, Конт.Контрагент);

Конт.Операция.Кредит.Субконто (2, Конт.Договор);

Конт.Операция.Кредит.Субконто (3, Конт.ТекущийДокумент ());

Конт.Операция.Сумма = БезНДС20;

Конт.Операция.СодержаниеПроводки = "Данные для автоматического учета
НДС";

КонецЕсли;

Если НДС20>0 **Тогда**

Конт.Операция.НоваяПроводка ();

Конт.Операция.НомерЖурнала = "АВ";

Конт.Операция.ПервичныйДокумент = глПредставлениеПервичногоДокумента
(Конт);

Конт.Операция.Кредит.Счет = СчетПоКоду ("ЗПК.20.Н");

Конт.Операция.Кредит.Субконто (1, Конт.Контрагент);

Конт.Операция.Кредит.Субконто (2, Конт.Договор);

Конт.Операция.Кредит.Субконто (3, Конт.ТекущийДокумент ());

Конт.Операция.Сумма = НДС20;

Конт.Операция.СодержаниеПроводки = "Данные для автоматического учета
НДС";

КонецЕсли;

Если БезНДС10>0 **Тогда**

Конт.Операция.НоваяПроводка ();

Конт.Операция.НомерЖурнала = "АВ";

Конт.Операция.ПервичныйДокумент = глПредставлениеПервичногоДокумента
(Конт);

Конт.Операция.Кредит.Счет = СчетПоКоду ("ЗПК.10.Б");

Конт.Операция.Кредит.Субконто (1, Конт.Контрагент);

Конт.Операция.Кредит.Субконто (2, Конт.Договор);

Конт.Операция.Кредит.Субконто (3, Конт.ТекущийДокумент ());

Конт.Операция.Сумма = БезНДС10;

Конт.Операция.СодержаниеПроводки = "Данные для автоматического учета
НДС";

КонецЕсли;

Если НДС10>0 Тогда

```
Конт.Операция.НоваяПроводка ();
Конт.Операция.НомерЖурнала = "АВ";
Конт.Операция.ПервичныйДокумент = глПредставлениеПервичногоДокумента
(Конт);
Конт.Операция.Кредит.Счет = СчетПоКоду ("ЗПК.10.Н");
Конт.Операция.Кредит.Субконто (1, Конт.Контрагент);
Конт.Операция.Кредит.Субконто (2, Конт.Договор);
Конт.Операция.Кредит.Субконто (3, Конт.ТекущийДокумент ());
Конт.Операция.Сумма = НДС10;
Конт.Операция.СодержаниеПроводки = "Данные для автоматического учета
НДС";
```

КонецЕсли;**Если НДС0>0 Тогда**

```
Конт.Операция.НоваяПроводка ();
Конт.Операция.НомерЖурнала = "АВ";
Конт.Операция.ПервичныйДокумент = глПредставлениеПервичногоДокумента
(Конт);
Конт.Операция.Кредит.Счет = СчетПоКоду ("ЗПК.0");
Конт.Операция.Кредит.Субконто (1, Конт.Контрагент);
Конт.Операция.Кредит.Субконто (2, Конт.Договор);
Конт.Операция.Кредит.Субконто (3, Конт.ТекущийДокумент ());
Конт.Операция.Сумма = НДС0;
Конт.Операция.СодержаниеПроводки = "Данные для автоматического учета
НДС";
```

КонецЕсли;**Если Освобождаемые>0 Тогда**

```
Конт.Операция.НоваяПроводка ();
Конт.Операция.НомерЖурнала = "АВ";
Конт.Операция.ПервичныйДокумент = глПредставлениеПервичногоДокумента
(Конт);
Конт.Операция.Кредит.Счет = СчетПоКоду ("ЗПК.ВН");
Конт.Операция.Кредит.Субконто (1, Конт.Контрагент);
Конт.Операция.Кредит.Субконто (2, Конт.Договор);
Конт.Операция.Кредит.Субконто (3, Конт.ТекущийДокумент ());
Конт.Операция.Сумма = Освобождаемые;
Конт.Операция.СодержаниеПроводки = "Данные для автоматического учета
НДС";
```

КонецЕсли;**КонецПроцедуры**

Аналогичная ситуация с книгой продаж.

Из изложенного следует, что лучше избегать включения в Глобальный модуль своих процедур, функций и переменных. Если этого избежать невоз-

можно, придется модифицировать Глобальный модуль после каждого обновления.

Сохранение конфигурации

Рассмотрим элементы диалога.

Список **Сохраняемые файлы** содержит имена и маски файлов, которые будут сохраняться в архивной копии. Пользователю не обязательно знать смысл перечисляемых файлов, первоначально список содержит файлы, жизненно важные для функционирования системы 1С:Предприятие (рис. 6.10).

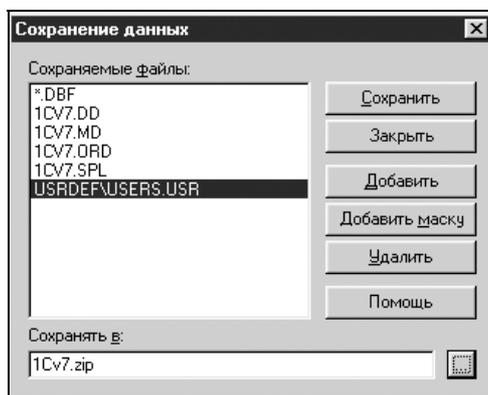


Рис. 6.10. Диалоговое окно **Сохранение данных**

Продвинутый пользователь может воспользоваться кнопкой **Добавить** и ввести дополнительные файлы для архивирования, но при этом он должен точно знать, зачем ему это надо (рис. 6.11).

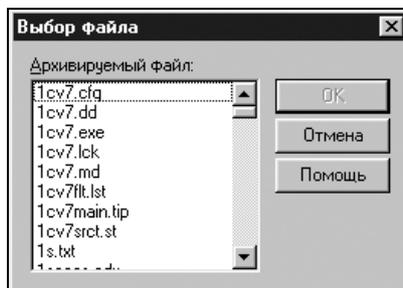


Рис. 6.11. Диалоговое окно **Выбор файла** для добавления в список архивируемых файлов

Кнопкой **Удалить** погорячившийся пользователь может удалить текущий файл из списка. Из списка нельзя удалить файлы, жизненно важные для функционирования системы 1С:Предприятие, — при выборе таких файлов кнопка **Удалить** станет недоступной.

Он также может воспользоваться кнопкой **Добавить маску** для задания маски выбираемых файлов в диалоговом окне (рис. 6.12) наподобие уже заданной маски *.DBF (см. рис. 6.10).

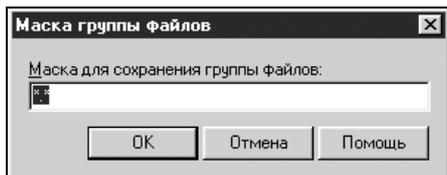


Рис. 6.12. Диалоговое окно задания добавляемой маски в список архивируемых файлов

Стандартно архивному файлу присваивается имя 1CV7.ZIP, можно (но не нужно) указать для архивного файла любое другое имя. Нажатие кнопки  открывает стандартное диалоговое окно для выбора каталога и имени архивного файла. Для выбора устройства в поле **Папка** в выпадающем списке необходимо выбрать каталог (рис. 6.13) и нажать кнопку **Сохранить**.

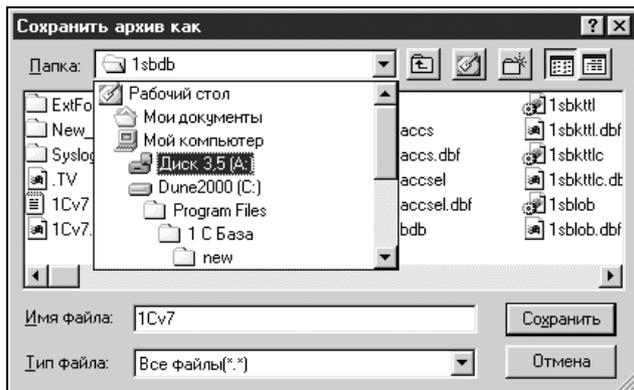


Рис. 6.13. Диалоговое окно для выбора каталога архива

Нажатием кнопки **Сохранить** запускается процесс сохранения данных.

Процесс начинается с архивирования и разбиения данных на тома во временный файл на жестком диске. Если было выбрано устройство **Диск 3,5(A:)**, запрос на установку дискеты № 1 появится чуть позже (рис. 6.14).

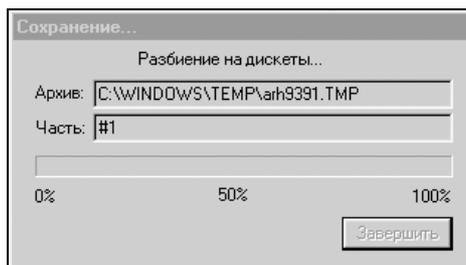


Рис. 6.14. Архивирование данных во временный файл

Внимание!

Следует обратить внимание на то, что первый запрос на смену дискеты будет к дискете № 1, а она уже стоит в дисковом дисководе, поэтому менять ее не следует (рис. 6.15). Просто нажмите кнопку **ОК**.

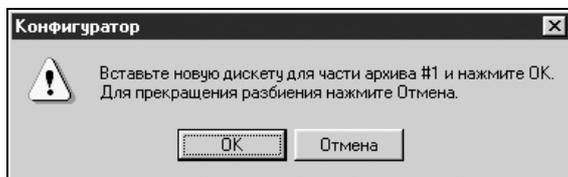


Рис. 6.15. Запрос на смену дискеты при сохранении данных

При следующих запросах на смену дискеты вставляйте дискету с требуемым номером. Если дискета не будет заменена, на экране появится соответствующее сообщение (рис. 6.16). Нажмите кнопку **ОК** и подождите повторного запроса на смену дискеты.

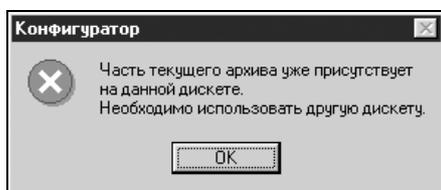


Рис. 6.16. Сообщение об ошибке смены дискеты

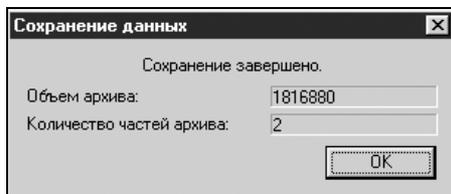


Рис. 6.17. Сообщение о результатах архивирования

Дождитесь сообщения о завершении сохранения и нажмите кнопку **ОК** (рис. 6.17).

Кнопка **Заккрыть** закрывает диалог **Сохранение данных**.

Внимание!

При повторном архивировании на те же дискеты на экран неоднократно будут выдаваться запросы на подтверждение перезаписи старого архива (рис. 6.18).

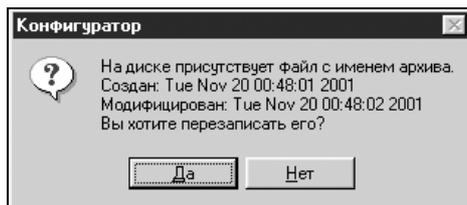


Рис. 6.18. Запрос на перезапись архива на дискете

Восстановление конфигурации

Сохраненные в архиве БД и конфигурация рано или поздно будут востребованы. Для восстановления из архива БД и конфигурации предусмотрен режим восстановления данных. Диалог восстановления данных вызывается выбором пункта **Восстановить данные** меню **Администрирования** (рис. 6.19).

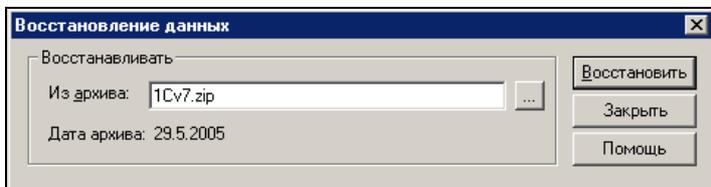


Рис. 6.19. Диалог восстановления данных

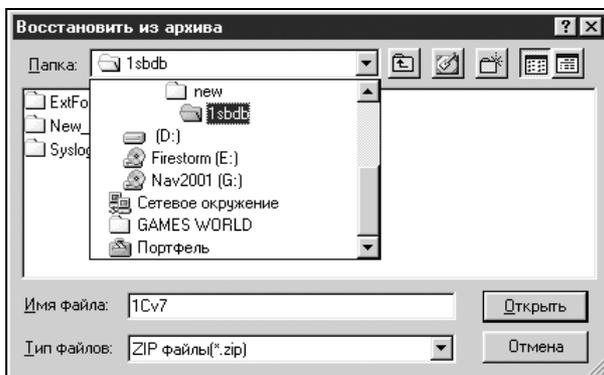


Рис. 6.20. Диалоговое окно выбора каталога архива

Нажатие кнопки  открывает стандартный диалог для выбора каталога и имени архивного файла (рис. 6.20).

После выбора архива нажатием кнопки **Восстановить** запускается процесс восстановления данных. Прежде чем программа заменит текущую информационную базу, на экран будет выведен запрос на подтверждение замены (рис. 6.21).

Дождитесь сообщения о завершении восстановления и нажмите кнопку **Да**.

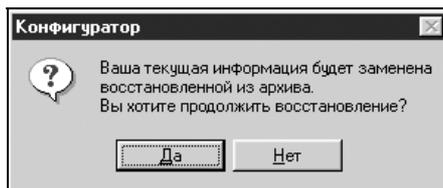
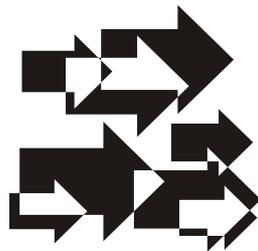


Рис. 6.21. Запрос на подтверждение замены текущих данных

ГЛАВА 7



Несколько слов о распределенной ИБ

Нет ничего удивительного в том, что распределенные информационные базы (ИБ) не нашли надлежащего применения на периферии. Даже довольно серьезные московские компании, имеющие филиалы во всех регионах России, сравнительно недавно организовали работу в распределенной информационной базе. Остается только предположить, что такая ситуация сложилась из-за того, что при отсутствии компоненты "Управление распределенными информационными базами" все основные режимы автообмена недоступны. Не имея возможности увидеть все диалоги, относящиеся к управлению распределенными информационными базами, трудно представить возможности данного механизма.

В обычную конфигурацию без включенного механизма управления распределенными базами невозможно добавить информацию без специальных обработок. Можно только заменить всю информацию с помощью механизма восстановления базы данных из архива, а с помощью механизма управления распределенными базами данных можно добавлять информацию в центральную базу данных из дочерних.

Префиксы

Идентификация информации в центральной базе данных (БД) основана на использовании префиксов. Рассмотрим реализацию ведения префиксов в конфигурации "Бухгалтерский учет". В начале работы с распределенной информационной базой необходимо установить константы **Префикс ИБ** и **Не применять префиксы ИБ**. Данные константы устанавливаются в диалоговом окне **Общая настройка конфигурации** (рис. 7.1).

Слой УРИБ (Управление распределенными информационными базами) реализуется в форме вкладки в предопределенной процедуре `ПриОткрытии()` :

```

Если ПустаяСтрока (ТекущаяИБКод ()) = 0 Тогда
    ПрефиксИБ = Константа.ПрефиксИБ;
    НеПрименятьПрефиксыИБ = Константа.НеПрименятьПрефиксыИБ;
КонецЕсли;
...
Если ПустаяСтрока (ТекущаяИБКод ()) = 0 Тогда
    Форма.Закладки.ДобавитьЗначение (5, "УРИБ");
КонецЕсли;

```

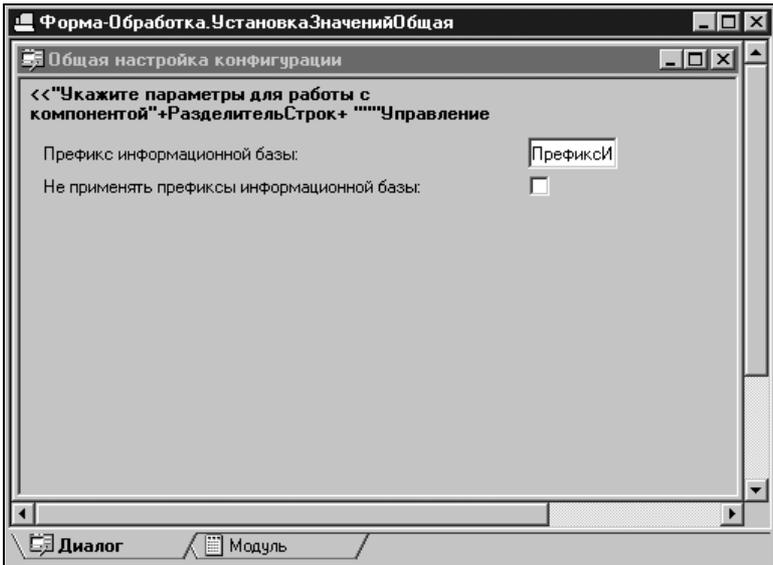


Рис. 7.1. Форма обработки УстановкаЗначенийОбщая в слое УРИБ

При попытке пользователя установить данные константы в окне **Список констант** (рис. 7.2) в отсутствии компоненты "Управление распределенными информационными базами" организовано сообщение (рис. 7.3) с помощью анализа в процедуре глобального модуля ПриЗаписиКонстанты () :

```

...
// Для работы с компонентой "Управление распределенными информационными
базами"
ИначеЕсли (ИмяКонстанты = "ПрефиксИБ") Тогда
    Если (Значение>="0") И (Значение<="9") Тогда
        Предупреждение ("Префикс ИБ должен быть буквенным!");
        СтатусВозврата (0);
    Возврат;
КонецЕсли;

```

Если ПустаяСтрока (ТекущаяИБКод ()) =1 **Тогда**

Предупреждение ("Данная константа используется только для распределенной ИБ");

СтатусВозврата (0);

Иначе

Предупреждение ("После изменения данной константы следует перезапустить 1С:Предприятие");

КонецЕсли;

ИначеЕсли (ИмяКонстанты = "НеПрименятьПрефиксыИБ") **Тогда**

Если ПустаяСтрока (ТекущаяИБКод ()) =1 **Тогда**

Предупреждение ("Данная константа используется только для распределенной ИБ");

СтатусВозврата (0);

Иначе

Предупреждение ("После изменения данной константы следует перезапустить 1С:Предприятие");

КонецЕсли;

Код	Наименование	Значение
Основное подразделение	Значение по умолчанию	Основное
Основная ставка НДС	Значение по умолчанию	18%
Основная ставка НП	Значение по умолчанию	Без налога
Основная единица измерения	Значение по умолчанию	шт
Основной вариант расчета н	Значение для подстановки в документ по умолчанию	НДС свер
Основной вид договоров	Значение по умолчанию	Основной
Номер листа кассовой книги	Устанавливается при формировании кассовой книги	
Ставка рефинансирования	Ставка рефинансирования ЦБР	13.00
Процент обязательной прод	Процент обязательной продажи валюты	75.00
Использовать список корре	Определяет режим выбора счета в документах	Нет
Все документы помещать в:	1 - помещать в ЖО все док-ты, Нет - только формирующие бух. провод	Да
При изменении количества €	1 - менять сумму и налоги, 2 - менять цену	
Дата запрета редактировани	Дата запрета редактирования документов	01.01.80
Префикс ИБ	Префикс информационной базы (для работы с распределенными инф	
Не применять префиксы ИБ	1 - Отключение использования префиксов информационных баз	
Код ИМНС	Номер налоговой инспекции	
Организация является плате	Организация является плательщиком налога на добавленную стоимо	Да
Организация является плате	Организация является плательщиком единого налога на вмененный	Нет
Организации предоставлена	Организации предоставлена льгота согласно п.1.2 ст.239 НК РФ	Нет
База распределения косвенн	Константа учетной политики	Сумма пр
Директ-костинг. Виды проду	Константа учетной политики	По всем в
Директ-костинг. База распр	Константа учетной политики	Выручка о
Учет з/пл во внешней прог.	Учет оплаты труда во внешней программе	Нет
Пересчитывать покупную це	Пересчитывать покупную цену	Нет
Основной вариант расчета н	Значение для подстановки в документ по умолчанию	НДС свер

Рис. 7.2. Диалоговое окно Список констант

При формировании номера нового документа префикс автоматически цепляется слева к очередному номеру. Аналогично при формировании кода нового элемента справочника префикс автоматически цепляется к очередному коду.

В центральной базе после очередного добавления информации префиксы сортируют и идентифицируют информацию данной информационной базы.

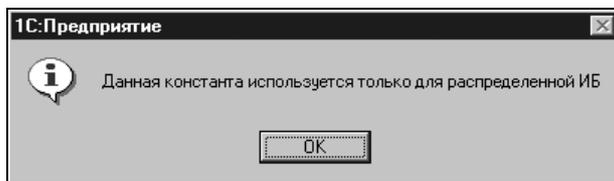


Рис. 7.3. Предупреждение об отсутствии компоненты "Управление распределенными информационными базами"

Автообмен

Для организации экспорта данных из периферийной базы в центральную предназначен диалог **Автообмен** (рис. 7.4).

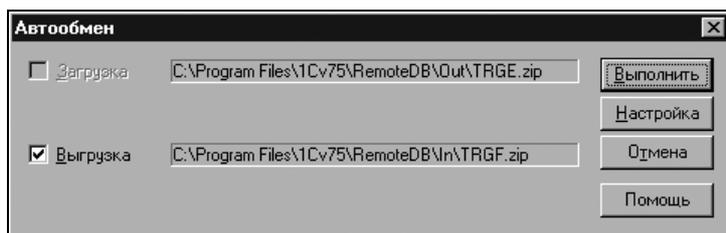


Рис. 7.4. Диалоговое окно Автообмен

Для вызова диалога необходимо закрыть все открытые окна Конфигуратора и выбрать пункт **Распределенная ИБ — Автообмен** из меню **Администрирование** главного меню Конфигуратора.

При попытке пользователя вызвать данный режим в отсутствии компоненты "Управление распределенными информационными базами" выдается сообщение (рис. 7.5)

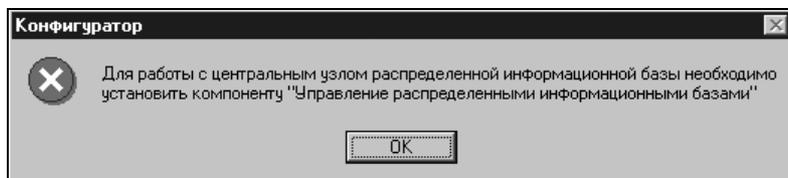


Рис. 7.5. Предупреждение об отсутствии компоненты "Управление распределенными информационными базами"

Опции **Загрузка**, **Выгрузка** определяют, какие операции следует выполнить.

Нажатие кнопки **Выполнить** запускает процесс обмена.

Нажатие кнопки **Настройка** вызывает диалог настройки параметров автообмена (рис. 7.6).

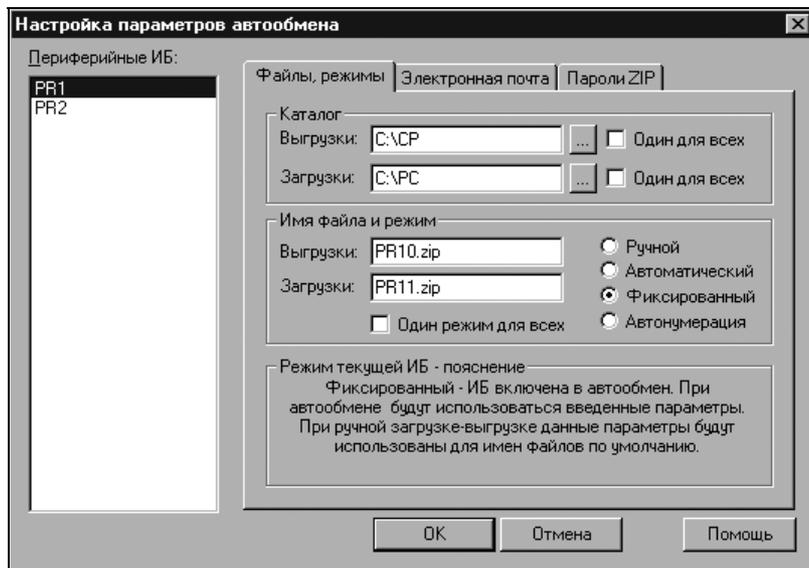


Рис. 7.6. Диалоговое окно **Настройка параметров автообмена** на вкладке **Файлы, режимы**

Настройка параметров автообмена

Для облегчения и ускорения процесса загрузки-выгрузки файлов обмена данных и уменьшения вероятности ошибки при вводе имени файла, в особенности на центральной ИБ при большом количестве периферийных, система поддерживает специальный механизм обмена файлами переноса данных — автоматический обмен файлами переноса данных (автообмен).

Список **Периферийные ИБ** содержит список существующих периферийных информационных баз.

В этом списке можно выбрать информационную базу, для которой будут настраиваться параметры автообмена.

На вкладке **Файлы, режимы** настраивается режим автообмена и имена файлов.

Группа **Каталог** — каталоги, которые будут использоваться центральной ИБ во время выполнения переноса данных.

Внимание!

Указанные каталоги должны существовать во время выполнения переноса данных.

Флажок **Один для всех** определяет условие, при котором указанные каталоги будут использоваться для всех периферийных информационных баз из списка.

В поле **Загрузки** задается имя файла, который будет использоваться распределенной ИБ во время выполнения загрузки данных.

В поле **Выгрузки** задается имя файла, который будет использоваться распределенной ИБ во время выполнения выгрузки данных.

Переключатель **Режим** позволяет выбрать режим автообмена:

- Ручной** — данная ИБ исключается системой из процесса автообмена. Введенные имена файлов и каталоги будут использоваться как путь по умолчанию при ручной загрузке и выгрузке;
- Автоматический** — ИБ включена в автообмен. Имена файлов задаются системой автоматически на основании кода ИБ и не редактируются. При ручной загрузке-выгрузке введенные имена файлов и каталоги будут использоваться как путь по умолчанию;
- Фиксированный** — то же, что **Автоматический**, но имена файлов задаются пользователем;
- Автоумерация** — то же, что **Автоматический**, но расширение файла и первые четыре символа имени задаются системой автоматически, а последние 3 символа имени, представляющие собой порядковый номер файла по основанию 36 (0...9, A...Z), могут быть заданы пользователем. В процессе проведения загрузок-выгрузок "номер" в имени файла будет инкрементироваться, и новые файлы переноса данных не будут затирать старые. Таким образом, будет сохранена история обмена файлами переноса данных между данной периферийной и центральной информационными базами.

На вкладке **Электронная почта** настраиваются параметры автоматического приема и отправки файлов переноса с использованием электронной почты (рис. 7.7).

Группа **Центральная ИБ** определяет параметры электронной почты для центральной информационной базы:

- Адрес** — адрес, который будет использоваться всеми периферийными ИБ для отправки сообщений, содержащих файл переноса данных;
- Конфигурация ЭП** — установленный в операционной системе клиент электронной почты может поддерживать работу с набором конфигураций,

определяющих параметры доступа к почтовому серверу. Данный управляющий элемент позволяет определить, какая конфигурация почтового клиента будет использоваться при автообмене;

- ❑ **Пароль** — вызов установленного в операционной системе клиента электронной почты может быть защищен паролем от несанкционированного доступа. В таком случае, в данном поле необходимо указать пароль запуска почтового клиента;
- ❑ **Обновлять содержимое почтового ящика** — если флажок установлен, то при каждом входе в диалог автообмена будет установлена связь с сервером электронной почты и содержимое папки входящих сообщений почтового клиента будет обновлено.

Внимание!

В зависимости от вариантов связи с почтовым сервером, эта операция может потребовать продолжительного времени.

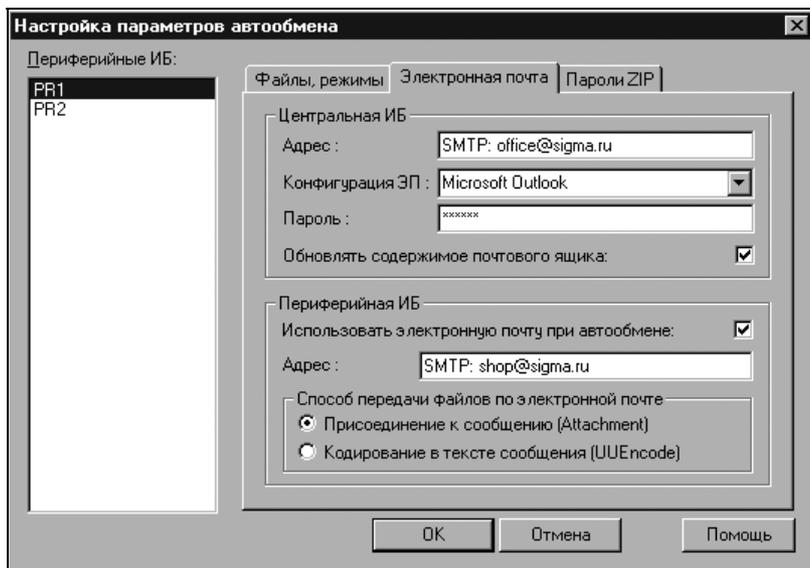


Рис. 7.7. Диалоговое окно Настройка параметров автообмена на вкладке Электронная почта

Группа **Периферийная ИБ** определяет параметры электронной почты для автообмена с конкретной ИБ. Следующие параметры должны быть заданы отдельно для каждой периферийной ИБ из списка:

- ❑ **Использовать электронную почту при автообмене** — если данный флажок не установлен, подсистема обмена сообщениями при автообмене

использоваться не будет. Заметим, что, хотя оставшиеся 2 параметра имеют значение только при установленном флажке, система хранит их значение и при выключении флага. Это позволяет временно отключать использование электронной почты при автообмене с данной периферийной ИБ, сохраняя значения остальных параметров, что может потребоваться, например, при ремонте телефонной линии, используемой периферийной ИБ для связи с почтовым сервером;

- **Адрес** — адрес, который будет использоваться центральной ИБ при посылке сообщений, содержащих файл переноса данных для данной периферийной информационной базы;
- **Способ передачи файлов по электронной почте** — определяет, будет ли при передаче файла использоваться механизм присоединения (attachment) к сообщению, либо кодирование в тексте сообщения. Устанавливается в зависимости от способности почтовых клиентов на обеих сторонах (центральной и периферийной ИБ) обрабатывать включение указанным способом файла в сообщение.

На вкладке **Пароли ZIP** задается пароль архива (рис. 7.8).

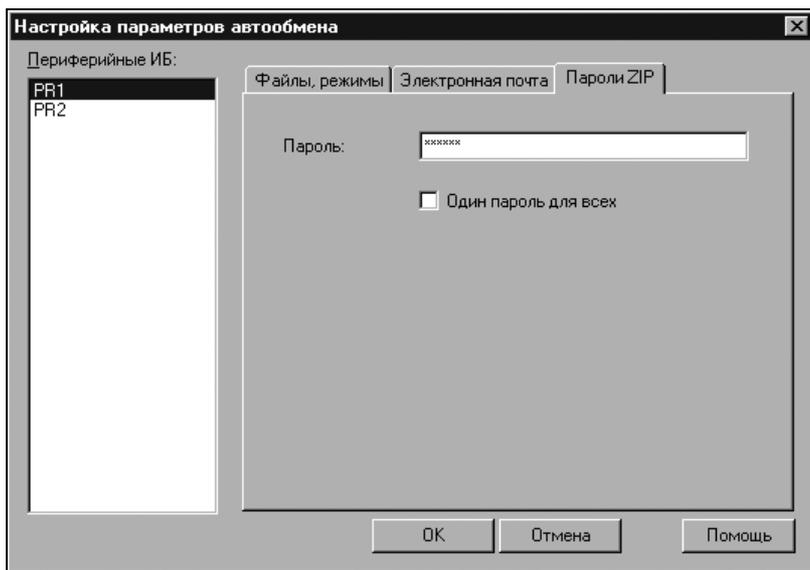


Рис. 7.8. Диалоговое окно **Настройка параметров автообмена** на вкладке **Пароли ZIP**

Пароль позволяет защитить файл переноса от несанкционированного использования, что особенно актуально при передаче по открытым каналам связи.

В качестве пароля может выступать любая последовательность символов. Заданный пароль будет использоваться при создании файлов переноса для

выбранной периферийной ИБ и при приеме файлов переноса от выбранной периферийной ИБ.

При изменении пароля первый файл переноса не сможет быть принят периферийной ИБ автоматически, т. к. информация об изменении пароля к ней еще дошла, и первый раз после изменения пароля прием файла переноса следует осуществить вручную, явно указав новый пароль.

Если флажок **Один пароль для всех** установлен, пароль текущей ИБ будет присвоен всем информационным базам, созданным на текущий момент, а также тем, которые будут создаваться в дальнейшем в диалоге **Управление распределенными информационными базами**.

Настройка параметров миграции объектов метаданных

Если установлена компонента "Управление распределенными информационными базами", в палитре свойств объекта метаданных появляется вкладка **Миграция** (рис. 7.9).

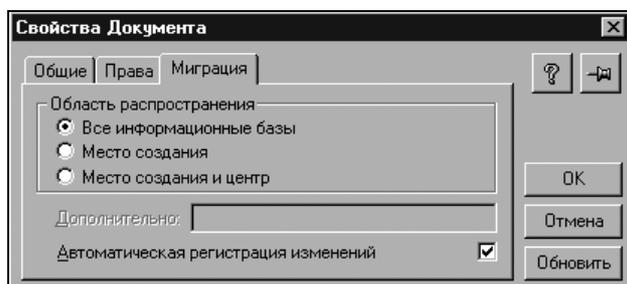


Рис. 7.9. Диалоговое окно **Свойства Документа** на вкладке **Миграция**

Управляющие элементы вкладки позволяют указать "направление" миграции конкретного объекта данных:

- Все информационные базы** — определяет полную миграцию: содержимое объекта метаданных будет синхронизироваться всеми информационными базами;
- Место создания** — содержимое объекта метаданных является строго индивидуальным для каждой информационной базы, и никак не синхронизируется;
- Место создания и центр** — устанавливает поведение системы, при котором вновь созданный на периферийной ИБ объект мигрирует на центральную ИБ и остается доступным только по месту создания и в центре.

Поле **Дополнительно** активизируется при установке переключателя **Область распространения** в режим **Место создания и центр**. В этом поле можно через запятую ввести коды дополнительных ИБ. В этом случае редактируемый объект метаданных будет мигрировать, помимо центральной ИБ, также на указанные периферийные информационные базы. Допускается использование символов-заменителей: например, "W**" означает все периферийные ИБ, чей код начинается с символа "W".

Если флажок **Автоматическая регистрация изменений** установлен, система автоматически фиксирует все изменения объектов, чтобы затем выполнить перенос измененных данных в другие информационные базы.

В большинстве случаев данный признак устанавливается. Отключение режима автоматической регистрации изменений производится в специальных случаях, когда объекты данного вида не должны при изменениях автоматически переноситься в другие информационные базы. В этом случае регистрация изменений будет производиться только при определенных обстоятельствах, регулируемых средствами встроенного языка в конфигурации.

Настройка параметров миграции операций

Если установлена компонента "Управление распределенными информационными базами", в палитре свойств операции появляется вкладка **Миграция** (рис. 7.10).

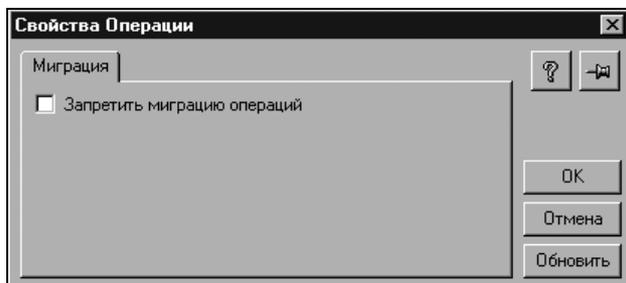


Рис. 7.10. Диалоговое окно **Свойства Операции** на вкладке **Миграция**

Установка флажка **Запретить миграцию операций** отключает миграцию бухгалтерских операций.

Отключение миграции для операции используется в специальных случаях, когда данные бухгалтерского учета имеют смысл только в тех информационных базах, в которых он ведется.

Настройка параметров миграции регистров

Если установлена компонента "Управление распределенными информационными базами", в палитре свойств регистров появляется вкладка **Миграция** (рис. 7.11).

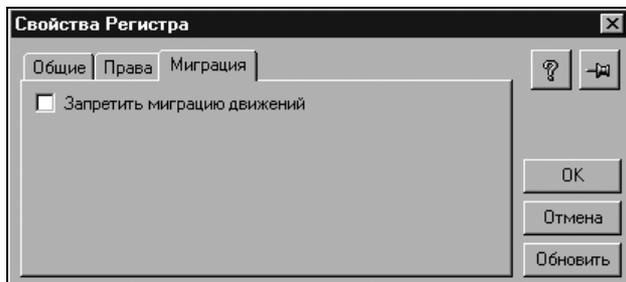


Рис. 7.11. Диалоговое окно **Свойства Регистра** на вкладке **Миграция**

Установка флажка **Запретить миграцию движений** отключает миграцию движений регистров. Отключение миграции для регистров используется в специальных случаях, когда движения регистров имеют смысл только в тех информационных базах, в которых они записывались при проведении документов.