

1С:Предприятие 8.1 учимся программировать *3-е издание*

НА ПРИМЕРАХ

Объекты конфигурации системы
1С:Предприятие 8.1

Встроенный язык
программирования

Разработка прикладных
решений для офиса

Интеграция 1С:Предприятие 8.1
с приложениями Microsoft Excel
и Microsoft Word

УДК 681.3.06
ББК 32.973.26-018.2
К31

Кашаев С. М.

К31 1С:Предприятие 8.1. Учимся программировать на примерах. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 384 с.: ил. + CD-ROM

ISBN 978-5-9775-0516-1

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

В третьем издании добавлены новая глава, касающаяся формирования статистических отчетов по имеющейся информации в базе данных, и материал по системе компоновки данных, обновлен также и ряд других разделов.

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

Для начинающих 1С-программистов и опытных пользователей

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

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

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.12.09.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 30,96.

Тираж 2500 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

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

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

ISBN 978-5-9775-0516-1

© Кашаев С. М., 2010
© Оформление, издательство "БХВ-Петербург", 2010

Оглавление

Введение	7
Читателям третьего издания книги.....	8
Общий взгляд на систему 1С:Предприятие.....	9
Краткое содержание книги.....	11
Компакт-диск.....	13
От автора.....	14
Глава 1. Представление об объектах системы 1С:Предприятие	15
Архитектура системы 1С:Предприятие.....	16
Взгляд на встроенный язык программирования.....	18
Постановка задачи.....	22
Окно запуска.....	22
Создание новой информационной базы.....	23
Перечисления.....	26
Справочники.....	29
Справочник <i>Должности</i>	30
Справочник <i>Специалисты</i>	33
Справочник <i>Курсы</i>	41
Печатная форма справочника.....	48
Внесение изменений в печатную форму.....	57
Справочник <i>Клиенты</i>	63
Документы.....	71
Документ <i>Внесение оплаты</i>	72
Заполнение закладки <i>Данные</i>	73
Форма документа.....	74
Программирование формы документа.....	77
Создание печатной формы документа <i>ВнесениеОплаты</i>	81
Список документов.....	82
Отчеты.....	83
Разработка макета.....	84
Процедура заполнения отчета.....	88

Отчет по клиентам	93
Отчет по клиентам с детализацией курсов.....	96
Итоги знакомства с конфигуратором	97
Глава 2. Документы.....	99
Подготовка конфигурации	101
Документ <i>Заказ</i>	101
Программное создание и запись документов	104
Поиск документов.....	107
Включение в условие отбора параметра	109
Упорядочивание	111
Программная установка свойств документов.....	112
Заполнение документа на основании имеющихся данных.....	113
Обработка событий	117
Глава 3. Регистры накопления.....	119
Регистр накопления остатков.....	120
Изменения в документе <i>Поступление</i>	122
Расход по регистру накопления	127
Оборотный регистр накопления	129
Получение выборки информации из регистра.....	133
Получение остатков	135
Глава 4. Регистры сведений.....	137
Организация неперiodических регистров сведений.....	137
Создание справочника <i>Номенклатура</i>	140
Документ <i>Поступление</i>	141
Процедура установки ответственного в поле документа.....	142
Создание периодического регистра сведений <i>Цены</i>	144
Использование регистра <i>Цены</i> в документе <i>Поступление</i>	147
Глава 5. Программирование форм	151
Постановка задачи	151
Представление информации о тарифах.....	154
Вспомогательный справочник <i>ВидТарифа</i>	154
Регистр сведений <i>Тарифы</i>	155
Контроль ошибок в форме регистра <i>Тарифы</i>	157
Справочник <i>Договоры</i>	159
Регистр сведений <i>Квитанции</i>	160
Форма обработки для учета квитанций по оплате	162
Процедуры для работы с формой	167
Процедура, выполняемая при выборе договора.....	169
Обработка нажатия кнопки <i>Расчет тарифа</i>	175
Процедура расчета суммы квитанции	176
Запись информации в регистр <i>Квитанции</i>	177

Форма регистрации квитанций	179
Формирование документа Microsoft Word.....	182
Внесение дополнений в бланк квитанции	183
Разработка процедуры заполнения бланка	186
Заполнение бланка с двумя квитанциями	189
Глава 6. Извлечение информации с помощью запросов.....	193
Язык запросов. Основные конструкции	195
Обработка <i>ИзвлечениеИнформации</i>	196
Конструкции <i>ВЫБРАТЬ</i> и <i>ИЗ</i>	198
Упорядочивание в запросе	200
Отбор максимальных (минимальных) значений	201
Ключевое слово <i>ИЕРАРХИЯ</i>	203
Исключение в отборе повторяющейся информации.....	204
Ключевое слово <i>ГДЕ</i>	206
Использование логических операторов в запросе.....	207
Агрегатные функции в запросах	209
Использование параметров в запросах.....	214
Ключевое слово <i>МЕЖДУ</i>	216
Функция расчета итогов в запросах.....	219
Объединение результатов нескольких запросов.....	223
Более сложные конструкции языка запросов	225
Отбор специалистов по указанной должности	225
Отбор курсов преподавателя	229
Соединение таблиц при организации запроса	230
Левое внешнее соединение.....	231
Правое внешнее соединение	234
Полное внешнее соединение	236
Ключевое слово <i>ВЫБОР</i>	237
Встроенные функции языка запросов	241
Функция <i>ПОДСТРОКА</i>	241
Функция <i>ДобавитьКДате</i>	242
Функция <i>РАЗНОСТЬДАТ</i>	243
Функция <i>ПРЕДСТАВЛЕНИЕ</i>	243
Оператор <i>ПОДОБНО</i>	244
Результат использования запросов	246
Глава 7. Отчеты и система компоновки данных.....	247
Отчет по остаткам товаров.....	249
Использование конструкторов настроек.....	258
Отчет в виде таблицы	261
Отчет в виде диаграммы.....	266
Формирование отчета по поступлениям с итоговой информацией.....	268
Формирование в отчете двух таблиц.....	272
Отчет по документам <i>Заявка</i>	272

Отчет по заявкам	276
Отчеты по информационной базе главы 5	281
Отчет по оплаченным квитанциям	282
Отчет по договорам.....	286
Отчет по квитанциям с группировкой	288
Отчеты по информационной базе главы 1	290
Отчет по учащимся	290
Отчет по курсам	292
Глава 8. Автоматизация работы с данными	295
Учет начислений преподавателям	296
Информация о тарифах.....	296
Вспомогательные справочники.....	299
Документ <i>Учет</i>	301
Процедуры обработки событий на форме документа.....	303
Просмотр актуальных тарифов	306
Оборотный регистр накопления	309
Движения по регистру <i>Начисление</i>	310
Отчет по начислению.....	313
Реализация множественного подбора	316
Использование системы компоновки данных для создания сводной таблицы.....	319
Формирование сводной таблицы с использованием конструктора	326
Передача данных между приложениями.....	329
Взгляд на технологии обмена данными	330
Использование СОМ-технологии	331
Передача информации в Microsoft Excel с использованием технологии OLE Automation	334
Выполнение макросов при передаче информации в Microsoft Excel	336
Интеграция трех офисных приложений	339
Глава 9. Анализ данных.....	347
Создание новой конфигурации	349
Общая статистика.....	353
Поиск ассоциаций	357
Поиск последовательностей.....	359
Статистика в учебном центре.....	362
Статистические отчеты к главе 2.....	370
Заключение.....	375
Приложение. Описание компакт-диска.....	377
Литература	379
Предметный указатель	381



Представление об объектах системы 1С:Предприятие

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

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

Будет правильно, если мы начнем с рассмотрения достаточно простых ситуаций, которые должны быть понятны читателям, только начинающим знакомиться со средствами разработки, имеющимися в программных продуктах фирмы "1С". Разумеется, сразу возникает вопрос: а что нужно предварительно знать для успешного восприятия изложенного в книге материала? Конечно, желательны навыки в области программирования, а также приветствуется

любой опыт работы с информационной системой 1С:Предприятие. Все это придаст читателю больше уверенности при работе с книгой, увеличит скорость рассмотрения примеров и позволит ему быстрее перейти к решению собственных задач. Однако можно обойтись и без этого.

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

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

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

Архитектура системы 1С:Предприятие

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

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

Подобная организация системы обеспечивает достаточную гибкость прикладных решений. В этом случае программист, даже не участвовавший в создании конкретной разработки, располагает всеми необходимыми ресурсами

для внесения в нее изменений. Следует заметить, что в настоящее время имеется большое количество стандартных и узкоспециализированных конфигураций, которые постоянно корректируются и совершенствуются. Такая открытость системы для модификации делает ее очень удобной для программистов, и они могут легко дорабатывать и развивать существующие типовые разработки.

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

В технологической платформе выделяются две составляющие:

- среда исполнения;
- среда разработки.

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

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

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

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

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

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

При создании прикладных решений применяются как визуальные средства разработки, так и программирование на встроенном языке. Любое новое прикладное решение — это совокупность метаданных. Однако когда вы начинаете новую разработку, то начинаете ее не с нуля. Платформа при создании новой конфигурации автоматически создает некую базовую структуру метаданных. Она уже сама по себе представляет собой разработку, которая может быть выполнена на платформе 1С. Конечно, какой-то практической пользы от этого нет, т. к. необходимой функциональности в нее еще не заложено. Задача разработчика как раз и заключается в том, чтобы эту функциональность создать путем доработки структуры метаданных. В дальнейшем после такой доработки при переходе в пользовательский режим работы система увидит эти изменения в структуре метаданных и реализует необходимую функциональность.

Набор объектов метаданных ограничен (справочник, документ, регистр накопления и еще ряд других) и разработчик не может создать свой, принципиально новый, объект конфигурации. Он может только добавлять в разрабатываемое прикладное решение новый объект конфигурации, который базируется на одном из прототипов, поддерживаемых системой.

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

Взгляд на встроенный язык программирования

В этом разделе мы дадим первые пояснения по встроенному языку программирования, который будет использоваться для прикладных разработок на

протяжении всех глав книги. Встроенный язык представляет собой предметно-ориентированный язык программирования, специально разработанный с учетом возможности его применения не только профессиональными программистами. Данный язык является двуязычным — все операторы языка имеют как русское, так и англоязычное написание.

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

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

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

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

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

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

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

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

Значения типа Дата содержат дату от Рождества Христова (с 01 января 0001 года) и время с точностью до секунды.

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

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

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

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

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

Здесь <Имя переменной 1> — переменная, которой при каждом повторении цикла присваивается значение очередного элемента коллекции. Другая переменная (<Имя переменной 2>) представляет саму коллекцию. Элементам этой коллекции будут периодически присваиваться переменной <Имя переменной 1>. Операторы, следующие за ключевым словом Цикл, выполняются до тех пор, пока не будут перебраны все элементы коллекции. Существует возможность завершить любой цикл досрочно. Для этого следует воспользоваться оператором Прервать.

Оператор Если управляет выполнением программы, основываясь на результате одного или более логических выражений. Этот оператор может содержать

любое количество групп, начинаемых с конструкции `ИначеЕсли`. Синтаксис рассматриваемого оператора выглядит так:

```
Если <Логическое выражение> Тогда
    // Операторы
ИначеЕсли <Логическое выражение> Тогда
    // Операторы
Иначе
    // Операторы
КонецЕсли;
```

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

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

```
Новый <Имя типа>
```

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

```
Пока Логическое выражение Цикл
    // Операторы
КонецЦикла
```

Операторы, следующие за ключевым словом `Цикл`, выполняются, пока результат логического выражения равен `Истина`.

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

Постановка задачи

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

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

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

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

Окно запуска

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

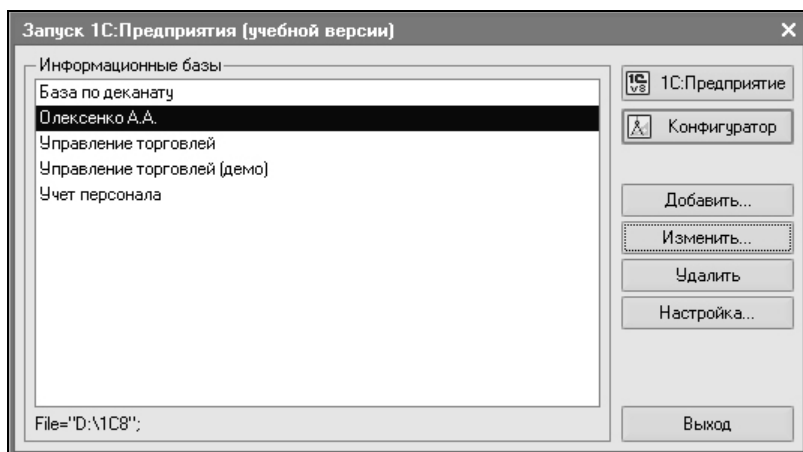


Рис. 1.1. Стартовое окно информационной системы 1С:Предприятие 8.1

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

Примечание

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

Создание новой информационной базы

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

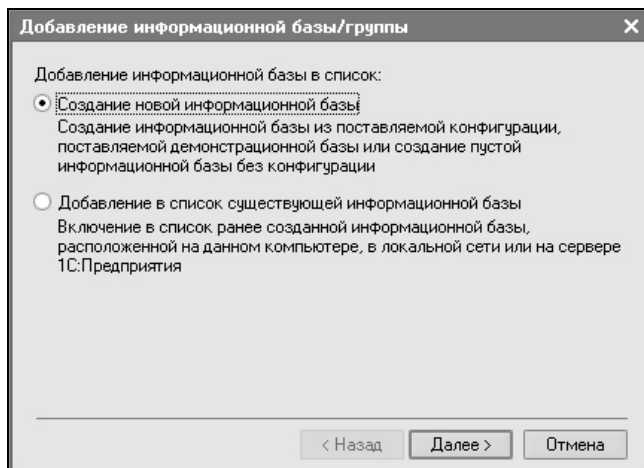


Рис. 1.2. Окно для добавления новой (существующей) информационной базы

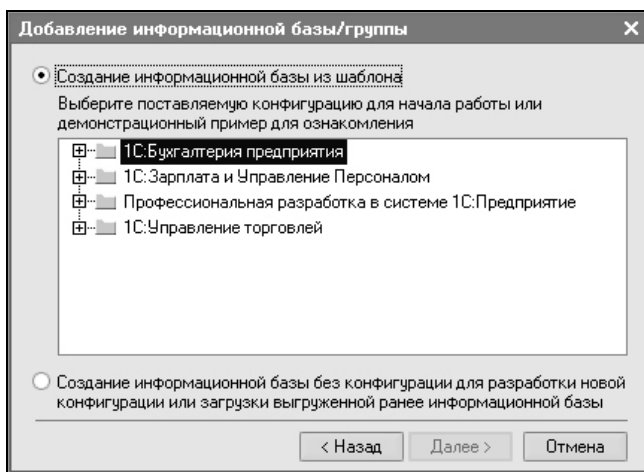


Рис. 1.3. Окно выбора способа создания информационной базы

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

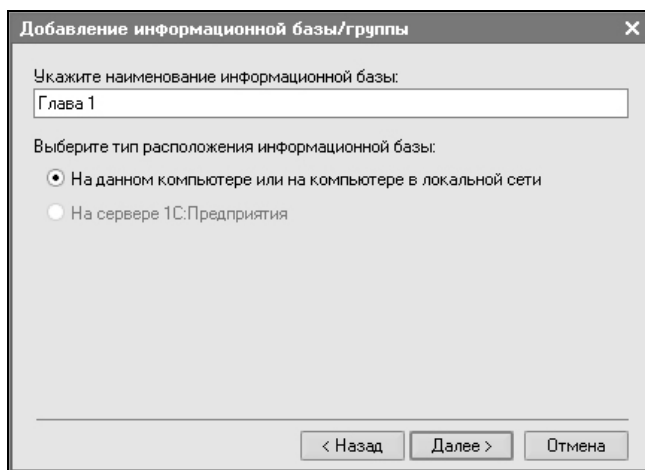


Рис. 1.4. Окно для указания наименования создаваемой информационной базы

После выбора названия информационной базы на экране повторно появляется окно, которое мы уже видели (см. рис. 1.1), но уже с новой строкой, содержащей название только что созданной информационной базы. Теперь с помощью кнопки **Конфигуратор** следует перейти в режим разработки объектов конфигурации, на основе которых системой будет построена база данных.

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

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

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

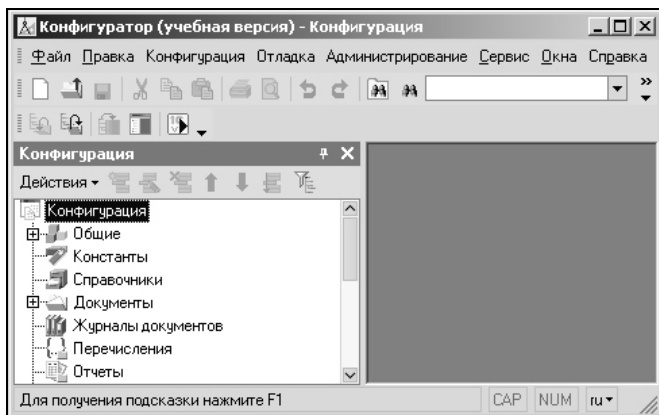


Рис. 1.5. Окно системы 1С:Предприятие 8.1 в режиме конфигуратора

Примечание

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

Перечисления

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

И первым объектом, который мы создадим в конфигураторе, будет перечисление с именем *Статус*, отражающее статус конкретного учащегося. Дело в том, что статус не подразумевает какого-то большого разнообразия возможных значений. Кроме того, данный параметр продолжительное время не изменяется (если происходит добавление какого-то нового значения для статуса в силу изменения методики работы нашего учебного центра, то это случается

очень редко). Фактически для подобных ситуаций объект типа *перечисление* и был разработан. В нашей ситуации будем считать значениями статуса учащегося следующие:

- Зачислен;
- Отчислен;
- Сертификат.

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

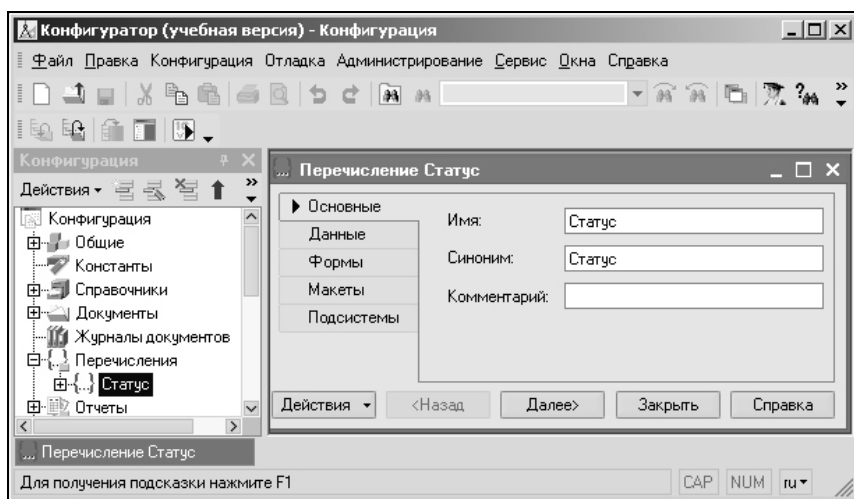


Рис. 1.6. Окно редактирования перечисления Статус

Примечание

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

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

Примечание

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

Таким образом, несложные действия привели к тому, что мы легко создали первый объект нашей конфигурации, который, правда, пока пуст. Для того чтобы внести в перечисление *Статус* возможные значения, следует перейти на закладку **Данные** и добавить необходимую информацию (рис. 1.7). Для внесения значений следует воспользоваться пиктографическим меню в верхней части окна редактирования объекта конфигурации. С помощью этих кнопок легко не только добавлять, но и корректировать и удалять элементы.

Вернемся к перечислениям. У читателя наверняка возникает вопрос: как перечисления, и в частности перечисление *Статус*, использовать? В этой главе мы создадим еще один объект — справочник *Клиенты*, в котором введем реквизит с именем *Статус*. Он будет принимать варианты только из набора зна-