

Никита Культин

Основы программирования в Delphi XE

Санкт-Петербург

«БХВ-Петербург»

2011

УДК 681.3.068+800.92Delphi XE

ББК 32.973.26-018.1

К90

Культин Н. Б.

К90 Основы программирования в Delphi XE. — СПб.: БХВ-Петербург, 2011. — 416 с.: ил. + CD-ROM — (Самоучитель)

ISBN 978-5-9775-0683-0

Книга является пособием для начинающих по программированию в Delphi. В ней в доступной форме изложены принципы визуального проектирования и событийного программирования, на конкретных примерах показана методика создания программ различного назначения, приведено описание среды разработки Delphi XE и базовых компонентов. Рассмотрены вопросы программирования графики, мультимедиа, разработки программ работы с базами данных Microsoft Access. Многочисленные примеры демонстрируют назначение компонентов, раскрывают тонкости программирования в Delphi. В приложении приведено описание базовых компонентов и наиболее часто используемых функций. Книга отличается доступностью изложения, большим количеством примеров. Прилагаемый компакт-диск содержит проекты, приведенные в книге.

Для начинающих программистов

УДК 681.3.068+800.92Delphi XE

ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.01.11.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 33,54.

Тираж 1500 экз. Заказ №

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

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09

от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

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

Оглавление

ПРЕДИСЛОВИЕ	1
ЧАСТЬ I. DELPHI XE	3
ГЛАВА 1. СРЕДА РАЗРАБОТКИ DELPHI XE	5
Установка	5
Первое знакомство	6
ГЛАВА 2. ПЕРВЫЙ ПРОЕКТ.....	11
Начало работы	11
Форма.....	11
Компоненты	16
Событие	25
Процедура обработки события.....	26
Редактор кода.....	30
Система подсказок.....	30
Шаблоны кода.....	32
Справочная информация.....	33
Сохранение проекта	33
Структура проекта	35
Компиляция.....	38
Ошибки	39
Предупреждения и подсказки.....	40
Запуск программы	41
Исключения.....	41
Обработка исключения	42
Внесение изменений.....	46
Настройка приложения	49
Установка приложения на другой компьютер.....	50

ГЛАВА 3. КОМПОНЕНТЫ	52
Базовые компоненты	52
<i>Label</i>	52
<i>Edit</i>	55
<i>Button</i>	57
<i>CheckBox</i>	60
<i>RadioButton</i>	63
<i>ComboBox</i>	65
<i>ListBox</i>	69
<i>Memo</i>	73
<i>Timer</i>	75
<i>Panel</i>	78
<i>ControlBar</i>	79
<i>SpeedButton</i>	80
<i>StatusBar</i>	83
<i>UpDown</i>	85
<i>ProgressBar</i>	88
<i>Image</i>	91
<i>MainMenu</i>	97
<i>OpenDialog</i>	101
<i>SaveDialog</i>	103
Компоненты Vista	105
<i>TaskDialog</i>	106
<i>FileOpenDialog</i> и <i>FileSaveDialog</i>	112
ЧАСТЬ II. ПРАКТИКУМ ПРОГРАММИРОВАНИЯ	121
ГЛАВА 4. ГРАФИКА	123
Графическая поверхность	123
Карандаш и кисть	126
Графические примитивы	127
Текст	128
Линия	133
Ломаная линия	138
Прямоугольник	139
Многоугольник (полигон)	143
Окружность и эллипс	147
Дуга	148
Сектор	148
Точка	154

Битовые образы.....	154
Мультипликация.....	159
Движение.....	159
Взаимодействие с пользователем	163
Использование битовых образов.....	168
ГЛАВА 5. МУЛЬТИМЕДИА	174
Функция <i>PlaySound</i>	174
Компонент <i>MediaPlayer</i>	175
Воспроизведение MIDI	183
Проигрыватель Audio CD	187
Просмотр видеороликов	195
Компонент <i>Animate</i>	201
ГЛАВА 6. БАЗЫ ДАННЫХ	204
База данных и СУБД.....	204
Локальные и удаленные базы данных	204
Структура базы данных.....	205
Механизмы доступа к данным	205
Компоненты доступа к данным	206
Создание базы данных	206
База данных Microsoft Access.....	206
Доступ к данным.....	207
Отображение данных	212
Выбор информации из базы данных.....	216
Работа с базой данных в режиме формы.....	222
Загрузка строки соединения из INI-файла.....	230
База данных Blackfish SQL	231
Доступ к серверу.....	232
Создание базы данных	232
Доступ к базе данных	237
Права пользователей	237
База данных "Книги"	239
Развертывание приложения работы с базой данных Blackfish SQL Server	244
ГЛАВА 7. КОМПОНЕНТ ПРОГРАММИСТА.....	248
Модуль компонента.....	249
Тестирование модуля компонента	257
Пакет компонентов.....	260
Создание пакета компонентов.....	261

Компиляция пакета компонентов	263
Установка пакета компонентов	264
Тестирование компонента	267
Установка программы на другой компьютер	270
Распространение компонента.....	270
ГЛАВА 8. СПРАВОЧНАЯ ИНФОРМАЦИЯ	271
Справочная система HTML Help	271
Подготовка справочной информации.....	272
Microsoft HTML Help Workshop.....	274
Файл проекта.....	274
Оглавление	277
Идентификаторы разделов.....	279
Компиляция.....	281
Отображение справочной информации.....	282
ГЛАВА 9. СОЗДАНИЕ УСТАНОВОЧНОГО ДИСКА	286
Утилита InstallAware	286
Новый проект	286
Общая информация	289
Программа и ее разработчик	289
Требования к системе.....	290
Компоненты	290
Архитектура	291
Возможности	291
Файлы.....	292
Ярлыки	293
Интерфейс	295
Диалоги	295
Информация о программе и лицензионное соглашение.....	296
Образ установочного диска	297
ГЛАВА 10. ПРИМЕРЫ ПРОГРАММ	299
Экзаменатор	299
Требования к программе.....	300
Файл теста	300
Форма приложения.....	303
Отображение иллюстрации	304
Доступ к файлу теста.....	305
Текст программы	306
Запуск программы	316

Сапер.....	317
Правила и представление данных.....	318
Форма.....	320
Игровое поле.....	321
Начало игры.....	321
Игра.....	325
Справочная информация.....	329
Информация о программе.....	330
Текст программы.....	333
MP3-плеер.....	343
Форма.....	344
Регулятор громкости.....	347
Перемещение окна.....	348
Текст программы.....	349

ПРИЛОЖЕНИЯ 357

ПРИЛОЖЕНИЕ 1. СПРАВОЧНИК..... 359

Форма.....	359
Базовые компоненты.....	360
<i>Label</i>	360
<i>Edit</i>	362
<i>Button</i>	362
<i>Memo</i>	363
<i>RadioButton</i>	364
<i>CheckBox</i>	365
<i>ListBox</i>	366
<i>ComboBox</i>	367
<i>StringGrid</i>	368
<i>Image</i>	369
<i>Timer</i>	370
<i>SpeedButton</i>	371
<i>UpDown</i>	372
<i>OpenDialog</i>	373
<i>SaveDialog</i>	374
<i>Animate</i>	375
<i>MediaPlayer</i>	376
Компоненты доступа/манипулирования данными.....	377
<i>ADODConnection</i>	377
<i>ADODTable</i>	377
<i>ADODDataSet</i>	378

<i>ADOQuery</i>	379
<i>DataSource</i>	380
<i>DBEdit, DBMemo, DBText</i>	380
<i>DBGrid</i>	381
<i>DBNavigator</i>	382
Графика.....	383
<i>PaintBox</i>	383
<i>Canvas</i>	384
<i>Pen</i>	386
<i>Brush</i>	387
Цвет	387
Функции.....	388
Функции ввода и вывода.....	388
Математические функции	388
Функции преобразования.....	389
Функции манипулирования датами и временем.....	390
События	391
Исключения.....	392

ПРИЛОЖЕНИЕ 2. СОДЕРЖИМОЕ КОМПАКТ-ДИСКА.....	393
--	------------

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	398
-----------------------------------	------------

Предисловие

Среда разработки Delphi является одним из популярнейших инструментов разработки прикладных программ. Она поддерживает так называемую быструю разработку, основанную на технологии визуального проектирования и событийного программирования, суть которой состоит в том, что среда разработки берет на себя большую часть рутинных операций, оставляя программисту работу по созданию диалоговых окон (визуальное проектирование) и процедур обработки событий (событийное программирование). Производительность программиста при этом просто фантастическая!

Delphi — это среда быстрой разработки приложений (RAD-среда, от *Rapid Application Development* — быстрая разработка приложений) на языке Delphi, в основе которого лежит хорошо знакомый многим программистам язык Pascal.

Изначально, вплоть до седьмой версии, Delphi была ориентирована на разработку Win32-приложений. После того как Microsoft стала продвигать технологию .NET, появилась Delphi 8 for The Microsoft .NET Framework — среда разработки .NET-приложений. Следующие версии Delphi выпускались в двух вариантах: для разработки Win32- и .NET-приложений. Теперь программистам стала доступна очередная версия Delphi — Embarcadero Delphi XE. Embarcadero — новое имя выделенного из Borland подразделения (изначально оно называлось CodeGear), отвечающего за инструменты разработки приложений.

Delphi XE существует в трех вариантах: Professional, Enterprise и Architect. Каждый комплект включает набор средств и компонентов, обеспечивающих разработку высокоэффективных приложений различного назначения, в том числе работы с базами данных InterBase, Blackfish SQL, Firebird, MySQL, Microsoft SQL Server, Oracle и др. Чем выше уровень пакета, тем больше возможностей он предоставляет программисту. Так, например, в Enterprise и Architect есть компоненты, позволяющие работать с удаленным сервером Blackfish SQL, а в Professional — только с локальным.

Среда Delphi XE доступна как отдельный инструмент разработки, а также как элемент Embarcadero RAD Studio XE.

Delphi XE может работать в среде операционных систем Microsoft Windows XP Home или Professional (SP2 или SP3), Microsoft Windows Vista SP2, Microsoft Windows Server 2003 (SP1) или 2008, а также в Microsoft Windows 7. Особых требований, по современным меркам, к ресурсам компьютера среда не предъявляет: процессор должен быть класса Intel Pentium (или совместимый) с частотой 1,4 ГГц (рекомендуется 2 ГГц и выше), 1 Гбайт оперативной памяти (рекомендуется 2 Гбайт и больше), 3,75 Гбайт свободного места на жестком диске (в том числе 750 Мбайт для Microsoft .NET Framework и Microsoft .NET SDK).

Книга, которую вы держите в руках, — это не описание среды разработки или языка программирования, это руководство по программированию в Delphi XE, разработке Win32-приложений. В ней представлена технология визуального проектирования и событийного программирования, подробно рассмотрен процесс создания программы, показано назначение базовых компонентов, описан процесс создания справочной системы и образа установочного диска, уделено внимание вопросам программирования графики, мультимедиа, разработки программ работы с базами данных.

Цель книги — научить программировать, создавать программы различного назначения: от простых однооконных приложений до программ работы с базами данных. Следует обратить внимание, что хотя книга ориентирована на читателя, обладающего начальными знаниями и опытом в программировании, она вполне доступна и начинающим.

Научиться программировать можно, только программируя, решая конкретные задачи. Поэтому чтобы получить максимальную пользу от книги, вы должны работать с ней активно. Изучайте листинги, старайтесь понять, как работают программы. Не бойтесь экспериментировать — совершенствуйте программы, вносите в них изменения. Чем больше вы сделаете самостоятельно, тем большему научитесь!

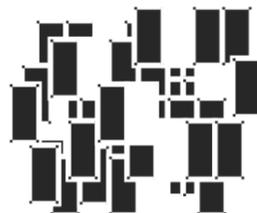


ЧАСТЬ I

DELPHI XE

В *данной части* книги приведено краткое описание среды разработки Delphi XE; на примере программы "Конвертер" показан процесс разработки приложения; приведены описание и примеры использования базовых компонентов.

ГЛАВА 1



Среда разработки Delphi XE

Установка

Основным вариантом поставки RAD Studio является "установка с сервера (Download)": при покупке программист получает серийный номер и ссылку на программу активизации установки, а все необходимые для установки файлы загружаются с сервера Embarcadero Technologies. При желании программист может за отдельную плату заказать DVD-диск и в дальнейшем использовать его, например, для установки отсутствующих компонентов.

Чтобы установить Delphi, надо с сайта компании Embarcadero Technologies загрузить пакет установки, представляющий собой ZIP-архив, распаковать его во временный каталог и запустить установщик (файл `install_RADStudio.exe`).

Delphi XE является .NET-приложением. Поэтому установка начинается с проверки наличия на компьютере разработчика Microsoft .NET Framework 3.5 SP1 Redistributable Package, Microsoft Visual J# version 2.0 Redistributable Package, Microsoft Data Access Components (MDAC) 2.8, Microsoft Core XML Services (MSXML) 6.0 и Language Pack for Microsoft .NET Framework 2.0. Если какой-либо из перечисленных компонентов отсутствует, то он устанавливается. После этого начинается установка Delphi.

Процесс установки обычный. Сначала на экране появляется окно лицензионного соглашения, затем — окно **Select Features**, в котором программист может выбрать необходимые для работы компоненты (точнее, отказаться от установки ненужных). По умолчанию на компьютер устанавливаются все доступные компоненты и, если на жестком диске достаточно свободного места, то в окне **Select Features** лучше ничего не трогать.

По окончании установки необходимо выполнить активацию продукта — ввести в окне активации серийный номер. Серийный номер передается на сервер регистрации, который в ответ пересылает на компьютер программиста файл активации. На этом процесс установки можно считать законченным.

Первое знакомство

Чтобы запустить Delphi XE, надо сделать щелчок на кнопке **Пуск** и в меню **Все программы** выбрать команду **Embarcadero RAD Studio XE ▶ Delphi XE**.

Затем, чтобы начать работу над новым *проектом*, надо в меню **File** выбрать команду **New ▶ VCL Forms Application — Delphi**.

Окно среды Delphi XE в начале работы над новым проектом приведено на рис. 1.1. В заголовке окна отображается имя проекта, над которым в данный момент работает программист. В верхней части окна находится строка меню и область отображения панелей инструментов.

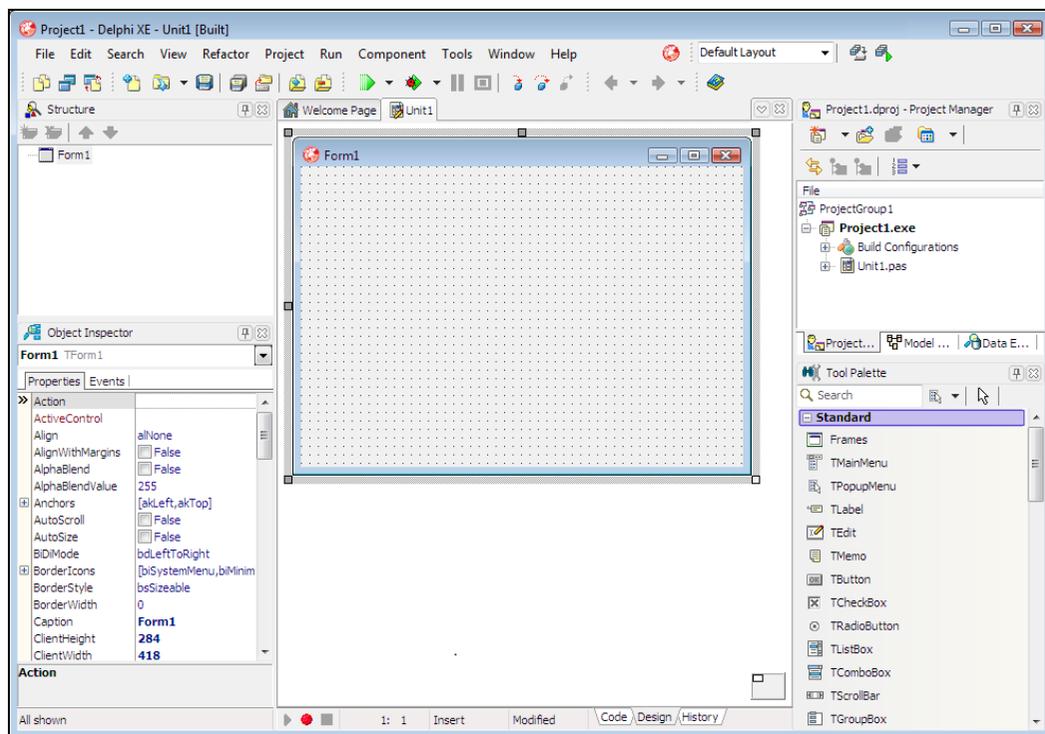


Рис. 1.1. Окно среды Delphi XE в начале работы над новым проектом

Центральную часть окна среды Delphi XE занимает окно конструктора (дизайнера) формы (рис. 1.2). В нем находится *форма* — заготовка окна приложения (окно программы во время его разработки принято называть формой).

За окном конструктора формы находится окно редактора кода (рис. 1.3). Доступ к окну редактора кода можно получить, сделав щелчок кнопкой мыши на ярлычке **Code** или нажав клавишу <F12> (повторное нажатие <F12> или щелчок кнопкой мыши на ярлычке **Design** активизирует конструктор формы).

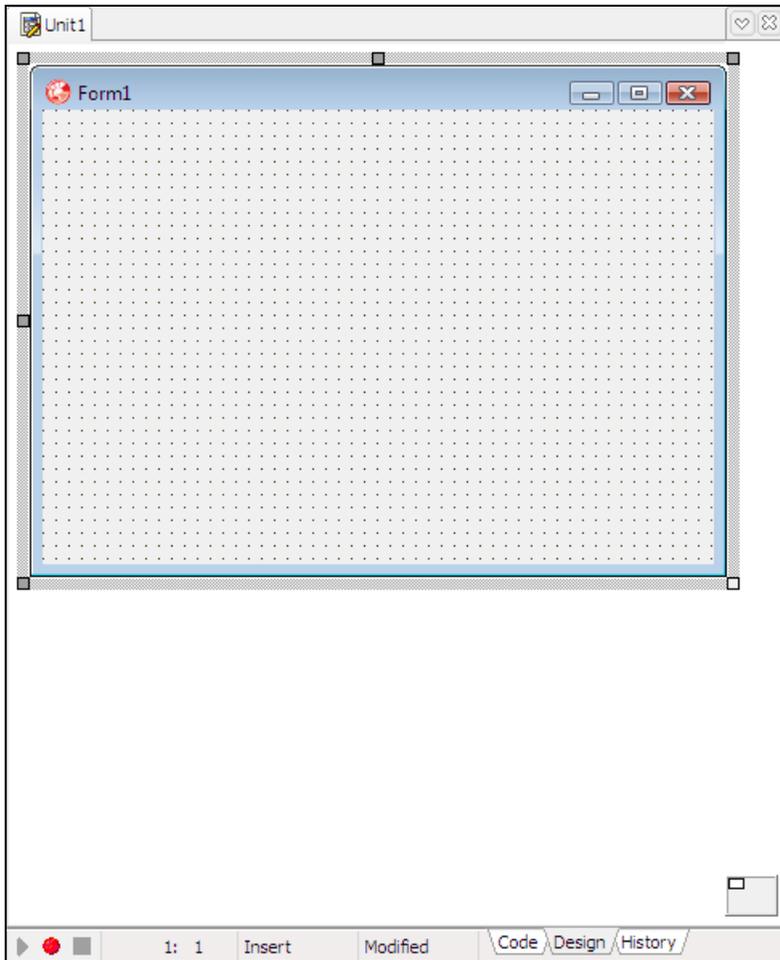


Рис. 1.2. Окно конструктора формы

Слева от окна дизайнера формы находится окно **Object Inspector** (рис. 1.4). Вкладка **Properties** этого окна используется для редактирования значений *свойств объектов*. Свойство (property) — это характеристика *объекта* (формы, командной кнопки, поля редактирования и т. д.). Свойства определяют вид объекта, его положение и поведение. Например, свойство `Caption` формы определяет текст, который отображается в ее заголовке, а свойства `Width` и `Height` — ее размеры (ширину и высоту). Справа от названия свойств указаны их значения. Свойства по функциональному признаку объединены в группы (названия групп выделены цветом). Так, например, свойства, определяющие внешний вид объекта, объединены в группу **Visual**. Программист может изменить способ отображения свойств, выбрав в контекстном меню вкладки **Properties** команду **Arrange ▸ by Name** (В алфавитном порядке) или **Arrange ▸ by Category** (По категориям). На вкладке **Events** окна **Object Inspector** перечислены *события*, которые может воспринимать объект.



Рис. 1.3. Окно редактора кода

На вкладках палитры компонентов (окно **Tool Palette**) находятся *компоненты* (рис. 1.5). Компонент — это элемент пользовательского интерфейса или объект, реализующий некоторую функциональность. Например, на вкладке **Standard** находятся компоненты, обеспечивающие взаимодействие с пользователем (Label — поле отображения текста; Edit — поле редактирования; Button — командная кнопка и др.), а на вкладке **dbGo** — компоненты доступа к базам данных.

Компоненты, обеспечивающие взаимодействие с пользователем, объединены в так называемую VCL-библиотеку (Visual Components Library). Поэтому приложения, использующие эти компоненты, называются VCL-приложениями (VCL Forms Application). Вспомните, чтобы начать работу над новым проектом (новой программой), надо в меню **File** выбрать команду **New ▶ VCL Forms Application**.

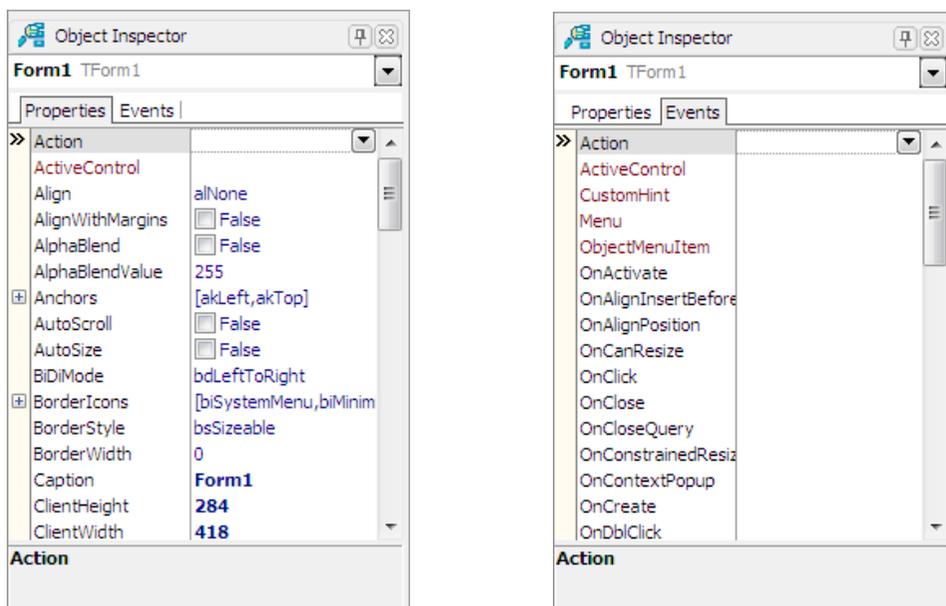


Рис. 1.4. В окне **Object Inspector** на вкладке **Properties** перечислены свойства объекта, а на вкладке **Events** — события, на которые объект может реагировать

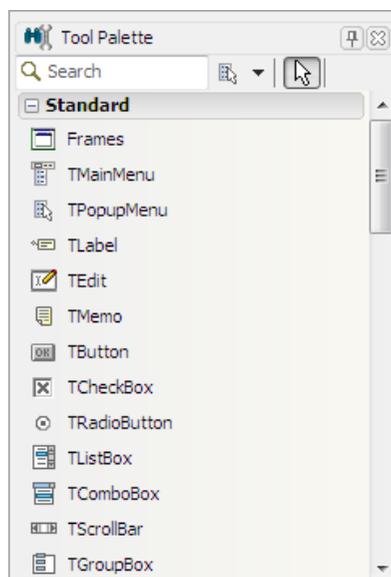


Рис. 1.5. Вкладка **Standard** содержит компоненты, обеспечивающие взаимодействие пользователя с программой

В окне **Project Manager** (рис. 1.6) отображается структура проекта, над которым в данный момент идет работа.

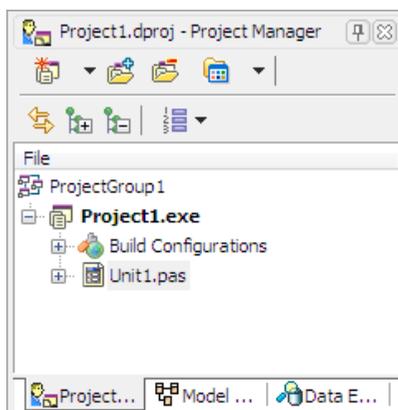


Рис. 1.6. В окне **Project Manager** отображается структура проекта

Если какое-либо из перечисленных окон на экране отсутствует, то для того чтобы его увидеть, надо в меню **View** выбрать соответствующую команду.

ГЛАВА 2



Первый проект

Процесс разработки программы в Delphi рассмотрим на примере — создадим *приложение* (так принято называть прикладную программу), с помощью которого можно пересчитать цену из долларов в рубли (рис. 2.1).

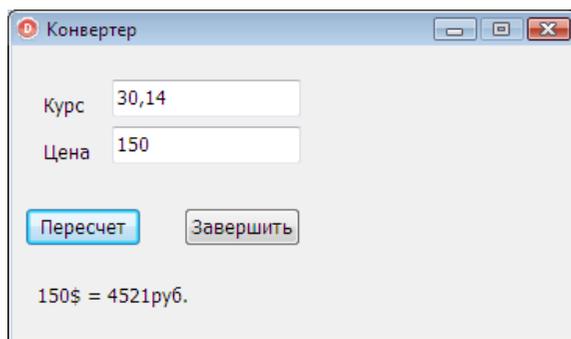


Рис. 2.1. Окно программы "Конвертер"

Начало работы

Чтобы начать работу над новым приложением, нужно в меню **File** выбрать команду **New ▶ VCL Forms Application - Delphi**.

Форма

Работа над приложением начинается с создания стартовой *формы* — главного окна программы.

Сначала нужно установить требуемые значения формы, затем — поместить на форму необходимые *компоненты* (поля ввода информации, командные кнопки, поля отображения текста и др.).

Настройка формы (а также компонентов) осуществляется путем изменения значений *свойств*. Свойства *объекта* (формы, компонента) определяют его вид и поведение. Например, свойство `Caption` определяет текст заголовка окна, а свойство `Position` — положение окна в момент появления на экране.

Основные свойства формы (объекта `TForm`) приведены в табл. 2.1.

Таблица 2.1. Свойства формы (объекта `TForm`)

Свойство	Описание
<code>Name</code>	Имя (идентификатор) формы. Используется для доступа к форме, ее свойствам и методам, а также для доступа к компонентам формы
<code>Caption</code>	Текст заголовка
<code>Width</code>	Ширина формы
<code>Height</code>	Высота формы
<code>Position</code>	Положение окна в момент первого его появления на экране (<code>poCenterScreen</code> — в центре экрана; <code>poOwnerFormCenter</code> — в центре родительского окна; <code>poDesigned</code> — положение окна определяют значения свойств <code>Top</code> и <code>Left</code>)
<code>Top</code>	Расстояние от верхней границы формы до верхней границы экрана
<code>Left</code>	Расстояние от левой границы формы до левой границы экрана
<code>BorderStyle</code>	Вид границы. Граница может быть обычной (<code>bsSizeable</code>), тонкой (<code>bsSingle</code>) или вообще отсутствовать (<code>bsNone</code>). Если у окна обычная граница, то во время работы программы пользователь может с помощью мыши изменить размер окна. Изменить размер окна с тонкой границей нельзя. Если граница отсутствует, то на экран во время работы программы будет выведено окно без заголовка. Положение и размер такого окна во время работы программы изменить нельзя
<code>BorderIcons</code>	Кнопки управления окном. Значение свойства определяет, какие кнопки управления окном будут доступны пользователю во время работы программы. Значение свойства задается путем присвоения значений уточняющим свойствам <code>biSystemMenu</code> , <code>biMinimize</code> , <code>biMaximize</code> и <code>biHelp</code> . Свойство <code>biSystemMenu</code> определяет доступность кнопки системного меню (значок в заголовке окна), <code>biMinimize</code> — кнопки Свернуть , <code>biMaximize</code> — кнопки Развернуть , <code>biHelp</code> — кнопки вывода справочной информации
<code>Icon</code>	Значок в заголовке диалогового окна, обозначающий кнопку вывода системного меню
<code>Color</code>	Цвет фона. Цвет можно задать, указав название цвета или привязку к текущей цветовой схеме операционной системы. Во втором случае цвет определяется текущей цветовой схемой, выбранным компонентом привязки и меняется при изменении цветовой схемы операционной системы

Таблица 2.1 (окончание)

Свойство	Описание
Font	Шрифт. Шрифт, используемый по умолчанию компонентами, находящимися на поверхности формы. Изменение свойства Font формы приводит к автоматическому изменению свойства Font компонента, располагающегося на поверхности формы. То есть компоненты наследуют свойство Font от формы (имеется возможность запретить наследование)

Для изменения значений свойств объектов используется вкладка **Properties** окна **Object Inspector**. В левой колонке этой вкладки перечислены свойства объекта, *выбранного* в данный момент, в правой — указаны значения свойств. Имя выбранного объекта отображается в верхней части окна **Object Inspector**.

На вкладке **Properties** свойства объединены в группы по функциональному признаку (названия групп выделены цветом). Например, группа **Visual** содержит свойства, определяющие вид объекта (для формы — заголовок, цвет фона, вид границы), а группа **Layout** — свойства, определяющие положение объекта (для формы — координаты левого верхнего угла). Некоторые свойства, например `width` и `height`, отображаются в нескольких группах (**Visual** и **Layout**).

Программист может изменить способ отображения свойств в окне **Object Inspector**. Например, чтобы свойства отображались в алфавитном порядке, в контекстном меню вкладки **Properties** надо выбрать команду **Arrange ▸ by Name**.

Чтобы в заголовке формы вместо `Form1` появилось название программы — текст `Конвертер`, следует изменить значение свойства `Caption`. Чтобы это сделать, надо в окне **Object Inspector** щелкнуть левой кнопкой мыши в строке свойства (в результате будет выделено текущее значение свойства и появится курсор), ввести текст `Конвертер` и нажать клавишу `<Enter>` (рис. 2.2).

Аналогичным образом можно установить значения свойств `height` и `width`, которые определяют высоту и ширину формы. Размер формы, а также размер других компонентов, задают в пикселах (точках). Свойствам `height` и `width` надо присвоить значения 215 и 366 соответственно.

Размер формы можно изменить и с помощью мыши, точно так же, как и любого окна, т. е. путем перемещения границы. По окончании перемещения границы значения свойств `height` и `width` будут соответствовать установленному размеру формы.

Положение окна на экране в момент его первого появления соответствует положению формы, заданному во время разработки программы. Положение можно задать, установив значение свойств `top` (отступ от верхней границы экрана) и `left` (отступ от левой границы экрана) или задав значение свойства `position`.

При выборе некоторых свойств, например `borderStyle`, справа от текущего значения свойства появляется значок раскрывающегося списка. Очевидно, что значение таких свойств можно задать путем выбора из списка (рис. 2.3).

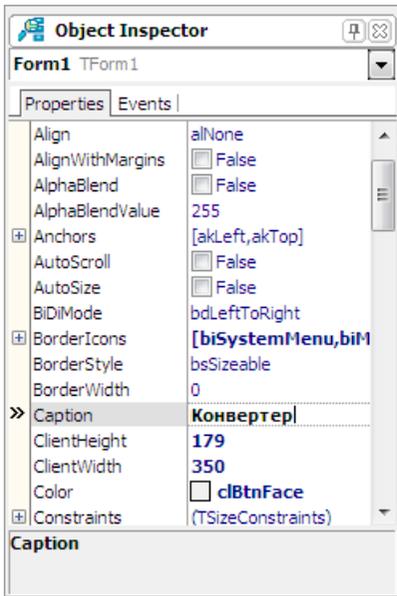


Рис. 2.2. Изменение значения свойства Caption путем ввода нового значения

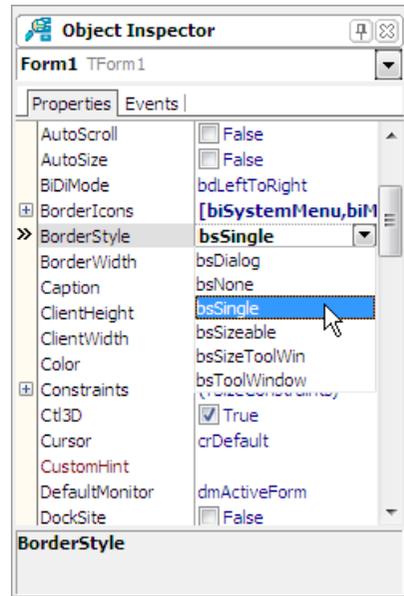


Рис. 2.3. Установка значения свойства путем выбора из списка

Некоторые свойства являются сложными, т. е. их значение определяется совокупностью значений других (уточняющих) свойств. Например, свойство `BorderIcons` определяет кнопки управления окном, которые будут доступны во время работы программы. Значения этого свойства определяются совокупностью значений свойств `biSystemMenu`, `biMinimize`, `biMaximize` и `biHelp`, каждое из которых, в свою очередь, определяет наличие соответствующей командной кнопки в заголовке окна во время работы программы. Перед именами сложных свойств стоит значок "+", в результате щелчка на котором раскрывается список уточняющих свойств (рис. 2.4). Значение уточняющего свойства можно задать обычным образом (ввести значение в поле редактирования или выбрать в списке).

В результате выбора некоторых свойств, например `Font`, в поле значения свойства отображается кнопка, на которой изображены три точки. Это значит, что задать значение свойства можно в дополнительном диалоговом окне, которое появится в результате щелчка на этой кнопке. Например, значение свойства `Font` можно задать путем ввода значений уточняющих свойств (`Name`, `Size`, `Style` и др.), а можно воспользоваться стандартным диалоговым окном **Шрифт**, которое появится в результате щелчка на кнопке с тремя точками (рис. 2.5).

В табл. 2.2 приведены значения свойств стартовой формы программы "Конвертер". Значения остальных свойств формы оставлены без изменения и поэтому в таблице не представлены. Обратите внимание, в именах некоторых свойств есть точка. Это значит, что это значение уточняющего свойства.

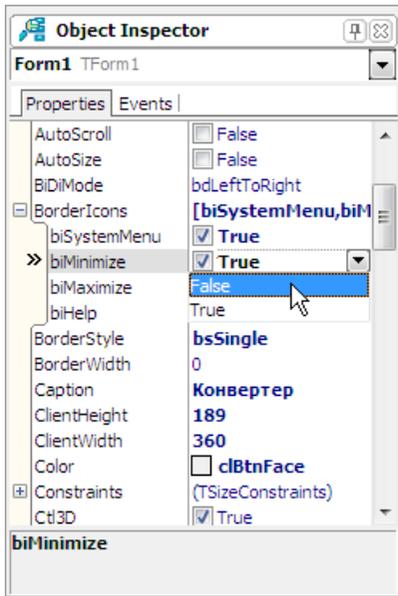


Рис. 2.4. Изменение значения уточняющего свойства

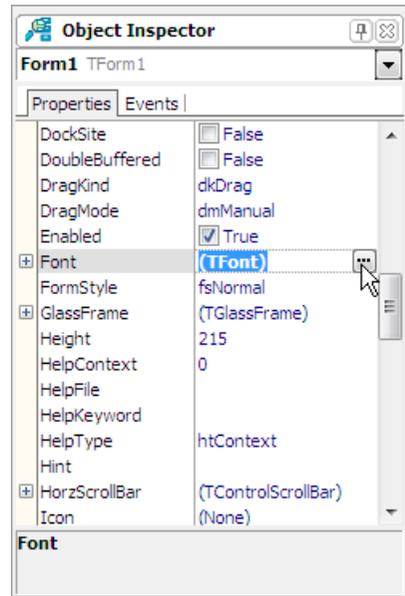


Рис. 2.5. Чтобы задать свойства шрифта, щелкните на кнопке с тремя точками

Таблица 2.2. Значения свойств стартовой формы

Свойство	Значение	Комментарий
Caption	Конвертер	
Height	215	
Width	366	
BorderStyle	bsSingle	Тонкая граница формы. Во время работы программы пользователь не сможет изменить размер окна путем захвата и перемещения его границы
Position	poDesktopCenter	Окно программы появится в центре рабочего стола
BorderIcons.biMaximize	False	В заголовке окна не отображать кнопку Развернуть
Font.Name	Tahoma	
Font.Size	10	

После того как будут установлены значения свойств формы, она должна выглядеть так, как показано на рис. 2.6. Теперь на форму надо добавить *компоненты*.

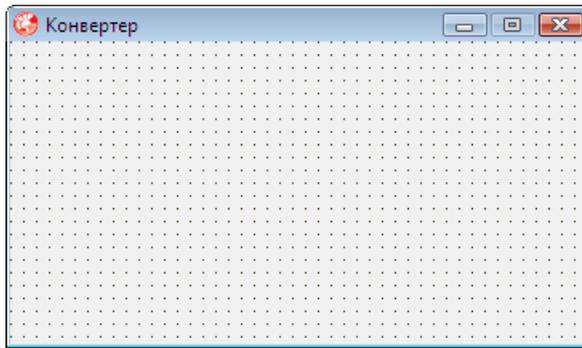


Рис. 2.6. Форма после изменения значений ее свойств

Компоненты

Поля редактирования, поля отображения текста, командные кнопки, списки, переключатели и другие элементы, обеспечивающие взаимодействие программы с пользователем, называют *компонентами пользовательского интерфейса*.

Компоненты, которые программист может использовать в процессе разработки программы, находятся в палитре компонентов (**Tool Palette**). На вкладках **Standard**, **Additional** и **Win32** располагаются часто используемые компоненты пользовательского интерфейса.

Программа "Конвертер" для пересчета цены из долларов в рубли должна получить от пользователя цену в долларах и курс. Для ввода данных с клавиатуры предназначен компонент `Edit`. Поэтому на форму разрабатываемого приложения нужно поместить два компонента `Edit`.

Чтобы на форму добавить компонент `Edit`, надо:

1. В палитре компонентов (окно **Tool Palette**) раскрыть вкладку **Standard**. Сделать щелчок на значке компонента `Edit` (рис. 2.7). Здесь следует обратить внимание, что в палитре компонентов, рядом со значком указывается тип компонента, а не его название.
2. Установить указатель мыши в ту точку формы, в которой должен быть левый верхний угол компонента.
3. Сделать щелчок левой кнопкой мыши.

В результате на форме появляется поле редактирования — компонент `Edit` (рис. 2.8).

Каждому добавленному компоненту среда разработки присваивает имя, которое состоит из названия компонента и его порядкового номера. Например, первый компонент `Edit` получает имя `Edit1`, второй — `Edit2`. Программист путем изменения значения свойства `Name` может поменять имя компонента. Однако в простых программах имена компонентов, как правило, не изменяют.

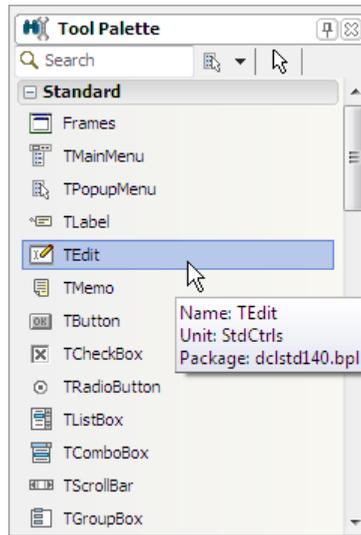


Рис. 2.7. Выбор компонента в палитре (компонент `Edit` — поле редактирования)

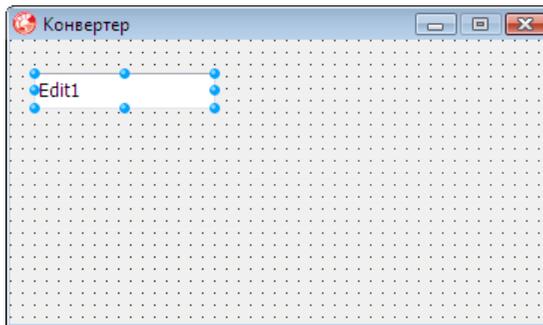


Рис. 2.8. Результат добавления на форму компонента `Edit`

Основные свойства компонента `Edit` приведены в табл. 2.3.

Таблица 2.3. Свойства компонента `Edit` (объект типа `TEdit`)

Свойство	Описание
Name	Имя компонента. Используется для доступа к компоненту и его свойствам
Text	Текст, который находится в поле редактирования
Left	Расстояние от левой границы компонента до левой границы формы
Top	Расстояние от верхней границы компонента до верхней границы формы
Height	Высота компонента

Таблица 2.3 (окончание)

Свойство	Описание
Width	Ширина компонента
Font	Шрифт, используемый для отображения текста в поле компонента
ParentFont	Признак наследования шрифта от формы. Если значение свойства равно True, то для отображения текста в поле компонента используется шрифт формы
MaxLength	Количество символов, которое можно ввести в поле редактирования. Если значение свойства равно нулю, ограничения на количество символов нет
TabOrder	Определяет порядок перемещения фокуса (курсора) с одного элемента управления на другой в результате нажатия клавиши <Tab>

На рис. 2.9 приведен вид формы после добавления двух полей редактирования. Один из компонентов *выбран* (выделен) — помечен маленькими кружками. Свойства выбранного компонента отображаются в окне **Object Inspector**. Чтобы увидеть и, если надо, изменить свойства другого компонента, нужно этот компонент выбрать — щелкнуть левой кнопкой мыши на изображении компонента или выбрать его имя в раскрывающемся списке, который находится в верхней части окна **Object Inspector** (рис. 2.10). Компонент, свойства которого надо изменить, можно выбрать и в окне **Structure** (рис. 2.11).

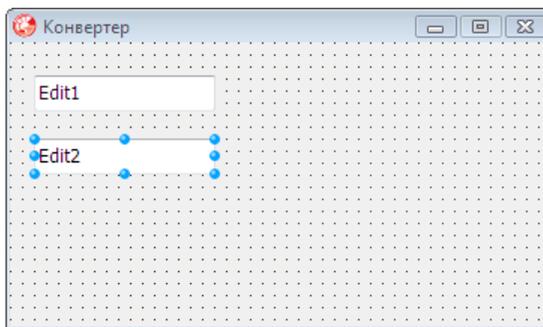


Рис. 2.9. Форма с двумя компонентами

Значения свойств, определяющих размер и положение компонента на поверхности формы, можно изменить с помощью мыши.

Чтобы изменить положение компонента, необходимо установить курсор мыши на его изображение, нажать левую кнопку мыши и, удерживая ее нажатой, переместить компонент в нужную точку формы. Во время перемещения компонента (рис. 2.12) отображаются текущие значения координат левого верхнего угла компонента (значения свойств `Left` и `Top`).

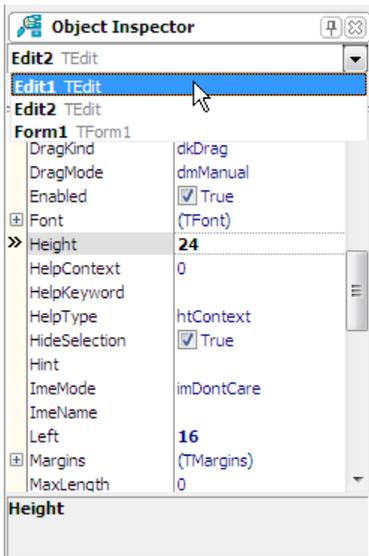


Рис. 2.10. Выбор компонента в окне **Object Inspector**

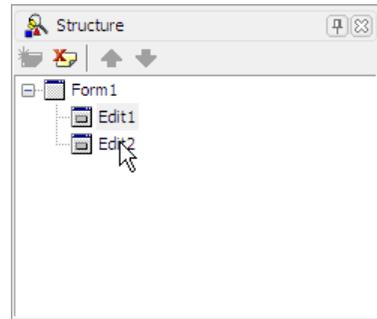


Рис. 2.11. Выбор компонента в окне **Structure**

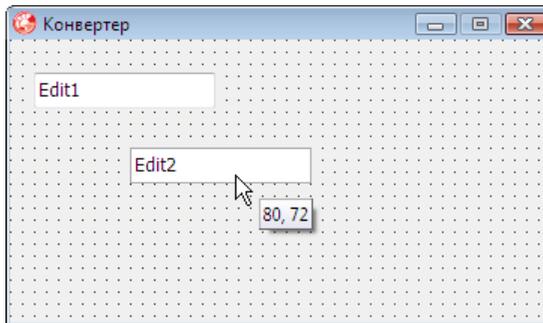


Рис. 2.12. Отображение значений свойств *Left* и *Top* при изменении положения компонента

Для того чтобы изменить размер компонента, необходимо сделать щелчок на его изображении (в результате чего компонент будет выделен), установить указатель мыши на один из маркеров, помечающих границу компонента, нажать левую кнопку мыши и, удерживая ее нажатой, изменить положение границы компонента. Во время изменения размера компонента отображаются его текущие размеры: ширина и высота (значения свойств *Width* и *Height*) (рис. 2.13).

В табл. 2.4 приведены значения свойств компонентов *Edit1* и *Edit2* (прочерк показывает, что значением свойства *Text* является пустая строка). Значения остальных свойств компонентов *Edit* оставлены без изменения, и поэтому в таблице не показаны. Компонент *Edit1* предназначен для ввода курса, *Edit2* — цены. Так как значения свойства *Font* компонентов *Edit* не указаны явно, то во время работы программы текст в полях редактирования будет отображаться шрифтом, заданным

для формы. Компонент `Edit`, как и другие компоненты, наследует значение свойства `Font` от своего родителя — объекта, на поверхности которого он находится. Поэтому если изменить значение свойства `Font` формы, автоматически изменится значение свойства `Font` компонентов, находящихся на форме. Если требуется, чтобы текст в поле компонента отображался другим шрифтом, нужно явно задать значение свойства `Font` этого компонента. Чтобы запретить автоматическое изменение значения свойства `Font` компонента при изменении свойства `Font` формы, надо свойству `ParentFont` компонента присвоить значение `False`.

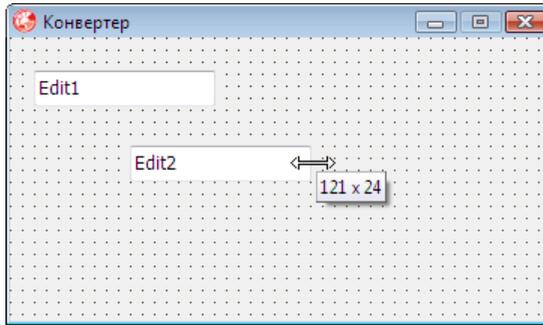


Рис. 2.13. Отображение значений свойств `Width` и `Height` при изменении размера компонента

Форма программы "Конвертер" после настройки компонентов `Edit` приведена на рис. 2.14.

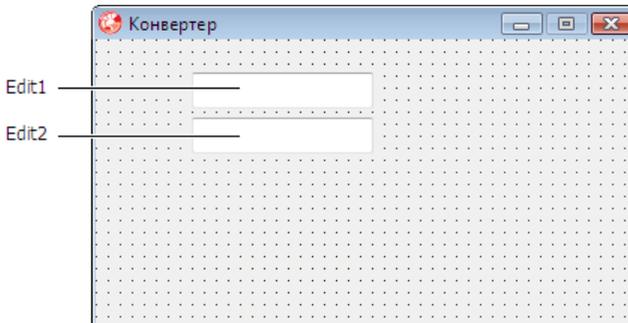


Рис. 2.14. Форма после настройки компонентов `Edit`

Таблица 2.4. Значения свойств компонентов `Edit`

Компонент	Свойство	Значение
Edit1	Top	64
	Left	22
	Text	-
	TabOrder	0

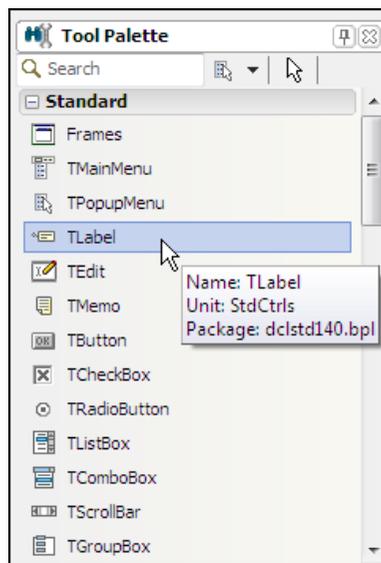
Таблица 2.4 (окончание)

Компонент	Свойство	Значение
Edit2	Top	52
	Left	22
	Text	-
	TabOrder	1

Отображение текста на поверхности формы обеспечивает компонент `Label`. В рассматриваемой программе текст отображается слева от полей редактирования (информация о назначении полей ввода). Результат расчета также отображается в окне программы. Поэтому в форму надо добавить три компонента `Label` (рис. 2.15).

Добавляются компоненты `Label` на форму точно так же, как и поля редактирования (компонент `Edit`).

Основные свойства компонента `Label` перечислены в табл. 2.5.

Рис. 2.15. Компонент `Label` — поле отображения текстаТаблица 2.5. Свойства компонента `Label`

Свойство	Описание
Name	Имя компонента. Используется для доступа к компоненту
Caption	Отображаемый текст

Таблица 2.5 (окончание)

Свойство	Описание
Font	Шрифт, используемый для отображения текста
ParentFont	Признак наследования характеристик шрифта от объекта (формы), на котором компонент находится
AutoSize	Признак автоматического изменения размера компонента при изменении текста, отображаемого в поле компонента
Left	Расстояние от левой границы поля вывода до левой границы формы
Top	Расстояние от верхней границы поля вывода до верхней границы формы
Height	Высота поля вывода
Width	Ширина поля вывода
WordWrap	Признак того, что слова, которые не помещаются в текущей строке, автоматически переносятся на следующую строку (значение свойства <code>AutoSize</code> должно быть <code>False</code>)

Если поле компонента `Label` должно содержать несколько строк текста, то перед тем как изменить значение свойства `Caption`, сначала надо присвоить свойству `AutoSize` значение `False`, а свойству `WordWrap` — `True`. Затем нужно установить требуемый размер компонента (с помощью мыши или вводом значений свойств `Width` и `Height`) и только после этого ввести значение свойства `Caption`.

На форму разрабатываемого приложения надо добавить три компонента `Label`. В полях `Label1` и `Label2` отображается информация о назначении полей ввода, поле `Label3` используется для вывода результата расчета. Значения свойств компонентов `Label` приведены в табл. 2.6.

Таблица 2.6. Значения свойств компонентов `Label`

Компонент	Свойство	Значение
Label1	Left	20
	Top	30
	Caption	Курс
Label2	Left	20
	Top	60
	Caption	Цена

Таблица 2.6 (окончание)

Компонент	Свойство	Значение
Label3	Left	16
	Top	152
	AutoSize	False
	Width	297
	Height	24
	Caption	-

После настройки компонентов `Label` форма разрабатываемого приложения должна выглядеть так, как показано на рис. 2.16.

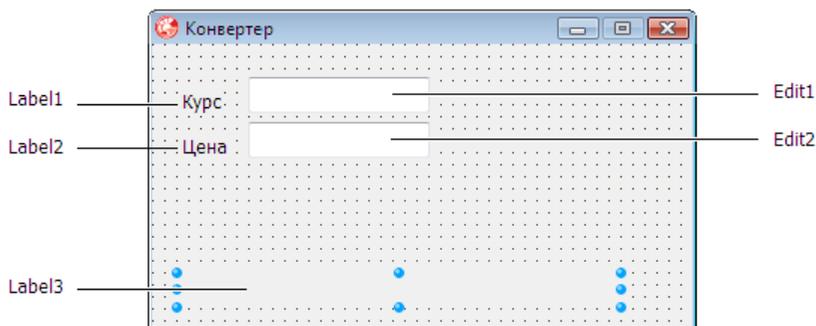


Рис. 2.16. Вид формы после настройки полей отображения текста

Последнее, что надо сделать на этапе создания формы, — добавить на форму две командные кнопки: **Пересчет** и **Завершить**. Назначение этих кнопок очевидно.

Командная кнопка, компонент `Button`, добавляется на форму точно так же, как и другие компоненты. Значок компонента `Button` находится на вкладке **Standard** (рис. 2.17). Основные свойства компонента `Button` приведены в табл. 2.7.

Таблица 2.7. Свойства компонента `Button`

Свойство	Описание
Name	Имя компонента. Используется для доступа к компоненту и его свойствам
Caption	Текст на кнопке
Enabled	Признак доступности кнопки. Кнопка доступна (программа реагирует на ее нажатие), если значение свойства равно <code>True</code> , и не доступна, если значение свойства равно <code>False</code>