



Самоучитель

Никита Культин

C++ Builder

2-е издание



Среда разработки

Базовые компоненты

Программирование графики,
мультимедиа и баз данных

Создание справочной системы
и установочного CD



+ CD

Никита Культин

C++ Builder

2-е издание

Санкт-Петербург

«БХВ-Петербург»

2008

УДК 681.3.068+800.92С++

ББК 32.973.26-018.1

К90

Культин Н. Б.

К90 С++ Builder. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2008. — 464 с.: ил. + CD-ROM — (Самоучитель)

ISBN 978-5-9775-0268-9

Книга представляет собой пособие по программированию в С++ Builder, в ней подробно рассмотрен процесс создания программы: от разработки диалогового окна и функций обработки событий до создания справочной системы и образа установочного компакт-диска. Изложены принципы визуального проектирования и событийного программирования, на конкретных примерах показаны возможности среды разработки, назначение компонентов, методика создания программ различного назначения. Рассмотрены вопросы программирования графики, анимации, мультимедиа, разработки программ работы с базами данных. В приложении приведено описание базовых компонентов и наиболее часто используемых функций. Книга адресована студентам, школьникам старших классов и всем, кто хочет научиться программировать в С++ Builder.

Компакт-диск содержит проекты, приведенные в книге в качестве примеров.

Для начинающих программистов

УДК 681.3.068+800.92С++

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>Римма Смоляк</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</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 30.05.08.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 37,41.

Тираж 3000 экз. Заказ №

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

Отпечатано с готовых диапозитивов

в ГУП "Типография "Наука"

199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0268-9

© Культин Н. Б., 2008

© Оформление, издательство "БХВ-Петербург", 2008

Оглавление

Предисловие	9
C++ Builder — что это?	9
Об этой книге	10
ЧАСТЬ I. СРЕДА РАЗРАБОТКИ C++ BUILDER	13
Глава 1. Начало работы	14
Установка.....	14
Активация	15
Первое знакомство	15
Глава 2. Первый проект	20
Начало работы.....	20
Форма.....	21
Компоненты.....	26
Событие и функция обработки события	36
Редактор кода	41
Система подсказок	41
Шаблоны кода	43
Справочная информация	43
Сохранение проекта	44
Структура проекта.....	45
Компиляция	49
Ошибки.....	51
Предупреждения и подсказки.....	52
Компоновка.....	53
Запуск программы.....	55
Исключения	55
Обработка исключений.....	56
Внесение изменений	60
Настройка приложения.....	66
Завершение проекта	68
Установка приложения на другой компьютер.....	68

Глава 3. Базовые компоненты.....	70
<i>Label</i>	70
<i>Edit</i>	73
<i>Button</i>	76
<i>CheckBox</i>	79
<i>RadioButton</i>	82
<i>ComboBox</i>	85
<i>ListBox</i>	88
<i>Memo</i>	93
<i>Timer</i>	96
<i>Panel</i>	98
<i>ControlBar</i>	100
<i>SpeedButton</i>	101
<i>StatusBar</i>	105
<i>UpDown</i>	108
<i>TrayIcon</i>	112
<i>ProgressBar</i>	115
<i>Image</i>	118
<i>MainMenu</i>	124
<i>PopupMenu</i>	132
<i>OpenDialog</i>	133
<i>SaveDialog</i>	137
ЧАСТЬ II. ПРАКТИКУМ ПРОГРАММИРОВАНИЯ	141
Глава 4. Графика.....	142
Графическая поверхность.....	142
Вывод графики	145
Перо и кисть	145
Графические примитивы	147
Текст.....	148
Линия.....	151
Ломаная линия.....	156
Прямоугольник.....	157
Многоугольник.....	160
Окружность и эллипс	161
Дуга.....	162
Сектор.....	163
Точка	163
Битовые образы	164
Загрузка битового образа из файла.....	165
Отображение битового образа	165
Метод <i>Draw</i>	165
Метод <i>CopyRect</i>	169

Загрузка битового образа из ресурса.....	172
Файл ресурсов.....	173
Подключение файла ресурсов.....	174
Загрузка битового образа.....	175
Анимация.....	175
Движение.....	176
Использование битовых образов.....	180
Взаимодействие с пользователем.....	186
Глава 5. Мультимедиа.....	193
Функция <i>PlaySound</i>	193
Компонент <i>MediaPlayer</i>	194
Простой MP3-плеер.....	198
Проигрыватель CD.....	203
Воспроизведение MIDI-музыки.....	209
Просмотр видеороликов.....	214
Компонент <i>Animate</i>	221
Глава 6. Базы данных.....	225
База данных и СУБД.....	225
Локальные и удаленные базы данных.....	225
Структура базы данных.....	226
Механизмы доступа к данным.....	227
Компоненты доступа к данным.....	228
Компоненты отображения данных.....	228
База данных Microsoft Access.....	228
Доступ к данным.....	229
Отображение данных.....	235
Выбор информации из базы данных.....	240
SQL-запрос.....	240
Фильтр.....	245
Работа с базой данных в режиме формы.....	246
Загрузка строки соединения из ini-файла.....	254
Сервер InterBase.....	255
Утилита IBConsole.....	258
Регистрация сервера.....	258
Подключение к серверу.....	260
Регистрация пользователя.....	261
Регистрация существующей базы данных.....	261
Соединение с БД.....	262
Создание базы данных.....	263
Создание таблицы.....	264
Ввод данных в таблицу.....	265
Удаление таблицы.....	266
Управление доступом к базе данных.....	266

Утилиты gsec и isql.....	267
Защита данных	269
Администратор	269
Псевдоним базы данных	269
Полномочия пользователя	270
Сценарии.....	271
Приложение работы с базой данных InterBase	271
Компоненты dbExpress	279
Установка программы работы с БД на другой компьютер	287
База данных Blackfish SQL.....	287
Установка.....	288
Доступ к серверу	288
Создание базы данных.....	289
Доступ к базе данных.....	293
Права пользователей.....	294
База данных <i>Книги</i>	296
Развертывание приложения работы с базой данных.....	301
Установка и настройка Blackfish SQL.....	301
Установка программы работы с базой данных	303
Глава 7. Компонент программиста	305
Модуль компонента	306
Тестирование компонента.....	315
Пакет компонентов	318
Создание пакета.....	319
Компиляция пакета	322
Установка пакета.....	323
Тестирование компонента.....	324
Установка программы на другой компьютер	327
Распространение компонента	327
Глава 8. Справочная информация	328
Справочная система HTML Help.....	328
Подготовка справочной информации.....	329
Создание chm-файла	331
Файл проекта	331
Оглавление.....	332
Идентификаторы разделов	336
Компиляция	338
Отображение справочной информации.....	338
Глава 9. Создание установочного диска.....	342
Утилита InstallAware	342
Новый проект	343

Общая информация.....	345
Программа и ее разработчик.....	345
Требования к системе.....	346
Компоненты.....	346
Архитектура.....	347
Возможности.....	347
Файлы.....	348
Ярлыки.....	349
Интерфейс.....	351
Диалоги.....	351
Информация о программе и лицензионное соглашение.....	353
Образ установочного диска.....	353
Глава 10. Примеры программ.....	356
Экзаменатор.....	356
Требования к программе.....	357
Файл теста.....	357
Форма.....	360
Отображение иллюстраций.....	361
Доступ к файлу теста.....	361
Текст программы.....	363
Настройка ОС.....	374
Сапер.....	375
Правила и представление данных.....	375
Форма.....	378
Игровое поле.....	379
Начало игры.....	379
Игра.....	382
Справочная информация.....	386
Информация о программе.....	388
Программа.....	390
MP3-плеер.....	399
Форма.....	400
Регулятор громкости.....	403
Перемещение окна.....	404
Листинг.....	405
Заключение.....	412
ПРИЛОЖЕНИЯ.....	413
Приложение 1. Справочник.....	414
Форма.....	414
Базовые компоненты.....	415
<i>Animate</i>	416
<i>Button</i>	416

<i>CheckBox</i>	417
<i>ComboBox</i>	418
<i>Edit</i>	419
<i>Image</i>	420
<i>Label</i>	421
<i>ListBox</i>	422
<i>MediaPlayer</i>	423
<i>Memo</i>	424
<i>OpenDialog</i>	425
<i>RadioButton</i>	426
<i>SaveDialog</i>	427
<i>SpeedButton</i>	428
<i>StringGrid</i>	429
<i>Timer</i>	431
<i>UpDown</i>	431
Компоненты доступа/манипулирования данными	432
<i>ADODConnection</i>	432
<i>ADODDataSet</i>	433
<i>ADOQuery</i>	434
<i>ADOTable</i>	435
<i>DataSource</i>	435
<i>DBEdit, DBMemo, DBText</i>	436
<i>DBGrid</i>	437
<i>DBNavigator</i>	438
Графика	440
<i>PaintBox</i>	440
<i>Canvas</i>	440
<i>Brush</i>	442
<i>Pen</i>	443
Цвет	444
Функции	445
Функции ввода и вывода	445
Математические функции	446
Функции преобразования	446
Функции манипулирования строками	447
Функции манипулирования датами и временем	448
События	451
Исключения	451
Приложение 2. Описание компакт-диска	453
Рекомендуемая литература	458
Предметный указатель	459

Предисловие

C++ Builder — что это?

Среда разработки Borland C++ Builder является одним из популярнейших инструментов разработки прикладных программ (приложений) различного назначения. Она ориентирована на так называемую "быструю" разработку, в основе которой лежит технология визуального проектирования и событийного программирования. Суть этой технологии в том, что среда разработки берет на себя большую часть рутинной работы, оставляя программисту работу по созданию диалоговых окон и процедур обработки событий. Производительность программиста при этом просто фантастическая!

C++ Builder — это среда разработки приложений (RAD-среда, от Rapid Application Development — быстрая разработка приложений) на языке C++. Среда является интегрированной, т. е. объединяет в себе специализированный редактор кода (текста программ), высокопроизводительный компилятор и отладчик. Следует обратить внимание на то, что хотя среда разработки C++ Builder объектно-ориентированная, тем не менее для работы в ней не требуется фундаментальных знаний в области объектно-ориентированного программирования — достаточно понимания концепции.

Изначально, вплоть до шестой версии, C++ Builder существовала как самостоятельное средство разработки. Затем вместе с другими инструментами Borland (Delphi и C# Builder) она была интегрирована в Borland Developer Studio. Теперь после реорганизации Borland Corporation и создания в ее структуре нового подразделения CodeGear разработчикам стала доступной очередная версия C++ Builder — CodeGear C++ Builder 2007.

CodeGear C++ Builder существует в двух вариантах: Professional и Enterprise. Каждый комплект включает набор инструментов и компонентов, обеспечивающих разработку высокоэффективных приложений различного назначе-

ния. Возможности Enterprise несколько больше (например, в него включены драйверы доступа к базам данных Oracle, MS SQL Server, Informix, которых нет в версии Professional). Помимо среды разработки, в комплект поставки C++ Builder включен новый сервер баз данных CodeGear Blackfish SQL Server.

Следует обратить внимание: среда C++ Builder 2007 доступна как отдельный инструмент разработки, а также как элемент CodeGear RAD Studio 2007. Более подробную информацию о продуктах CodeGear можно найти на сайте фирмы (<http://codegear.com>).

CodeGear C++ Builder 2007 может работать в среде операционных систем Microsoft Windows 2000, Microsoft Windows XP Professional, Microsoft Windows Server 2003, а также в Microsoft Windows Vista. Особых требований, по современным меркам, к ресурсам компьютера среда разработки не предъявляет:

- процессор должен быть класса Pentium III 1,4 ГГц или более мощный (рекомендуется Pentium 4 с частотой 2 ГГц + МГц);
- 512 Мбайт оперативной памяти (рекомендуется более 1 Гбайт);
- 750 Мбайт свободного места на жестком диске, в т. ч. для Microsoft .NET Framework и Microsoft .NET SDK (среда C++ Builder 2007 является .NET-приложением).

Об этой книге

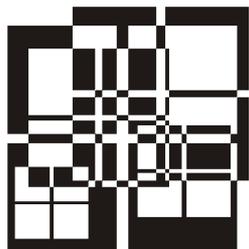
Книга, которую вы держите в руках, — это не описание среды разработки или языка программирования. Это пособие по программированию в C++ Builder и по разработке Win32-приложений. В ней представлена технология визуального проектирования и событийного программирования, на конкретных примерах рассмотрен процесс создания программы: от разработки диалогового окна и процедур обработки событий до создания справочной системы и образа установочного диска.

В первой части приводится краткое описание среды разработки C++ Builder, вводятся основные понятия и термины технологии визуального проектирования и событийного программирования, приводится описание базовых компонентов.

Вторая часть посвящена практике программирования. В ней рассматриваются задачи программирования графики, разработки программ работы с базами данных, создания компонентов. Уделено внимание вопросам создания справочной системы и развертывания приложений.

Цель книги — научить читателя создавать программы различного назначения: от простых однооконных приложений до программ работы с графикой и базами данных. Следует обратить внимание на то, что хотя книга ориентирована на тех, кто обладает начальными знаниями и опытом в программировании, но тем не менее 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!

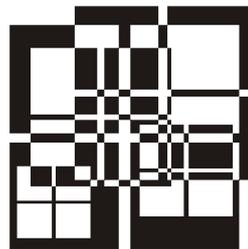


часть I

Среда разработки C++ Builder

- Глава 1. Начало работы
- Глава 2. Первый проект
- Глава 3. Базовые компоненты

ГЛАВА 1



Начало работы

Установка

Установка C++ Builder на компьютер выполняется с DVD-диска, на котором, помимо C++ Builder, находится сервер баз данных InterBase 2007 (версия Developer Edition), утилита InstallAware, а также набор .NET-компонентов.

Процесс инсталляции, как правило, активизируется автоматически, после того как установочный диск будет помещен в дисковод.

Следует обратить внимание на то, что C++ Builder является .NET-приложением. Поэтому установка начинается с проверки наличия на компьютере .NET-компонентов (Microsoft .NET 2.0 Framework, Microsoft .NET 2.0 Framework SDK и Microsoft JSharp Runtime 2.0), необходимых для работы C++ Builder. Если какого-либо из перечисленных компонентов нет, то он автоматически устанавливается.

Процесс установки C++ Builder обычный. Сначала надо подтвердить согласие с условиями лицензионного соглашения, ввести серийный номер продукта и указать компоненты, которые следует установить (точнее — те, которые устанавливать не надо). Также можно указать диск и каталог, куда следует установить C++ Builder (по умолчанию C++ Builder 2007 устанавливается в каталог C:\Program Files\CodeGear\RAD Studio\5.0). По завершении подготовительных действий начинается установка.

Замечание

Помимо C++ Builder на компьютер программиста автоматически устанавливается сервер баз данных Blackfish SQL Server. Сервер InterBase, утилита InstallAware по умолчанию на компьютер разработчика не устанавливаются.

Активация

CodeGear требует активации всех своих продуктов (в том числе trial-версий). Активация (привязка установленного продукта к конкретному компьютеру и пользователю) выполняется через Интернет при первом запуске C++ Builder: после ввода в окне активации серийного номера, имени пользователя и пароля устанавливается соединение с сервером и на компьютер программиста передается активационный файл.

Первое знакомство

Чтобы запустить C++ Builder, надо сделать щелчок на кнопке **Пуск** и в меню **Все программы** выбрать команду **CodeGear Studio ▶ C++ Builder**.

Для того чтобы начать работу над новой программой (Win32-приложением) или, как принято говорить, *проектом*, надо в меню **File** выбрать команду **New ▶ VCL Forms Application — C++ Builder**.

Окно **C++ Builder** в начале работы над новым проектом приведено на рис. 1.1. В верхней части окна находится строка меню и панели инструментов.

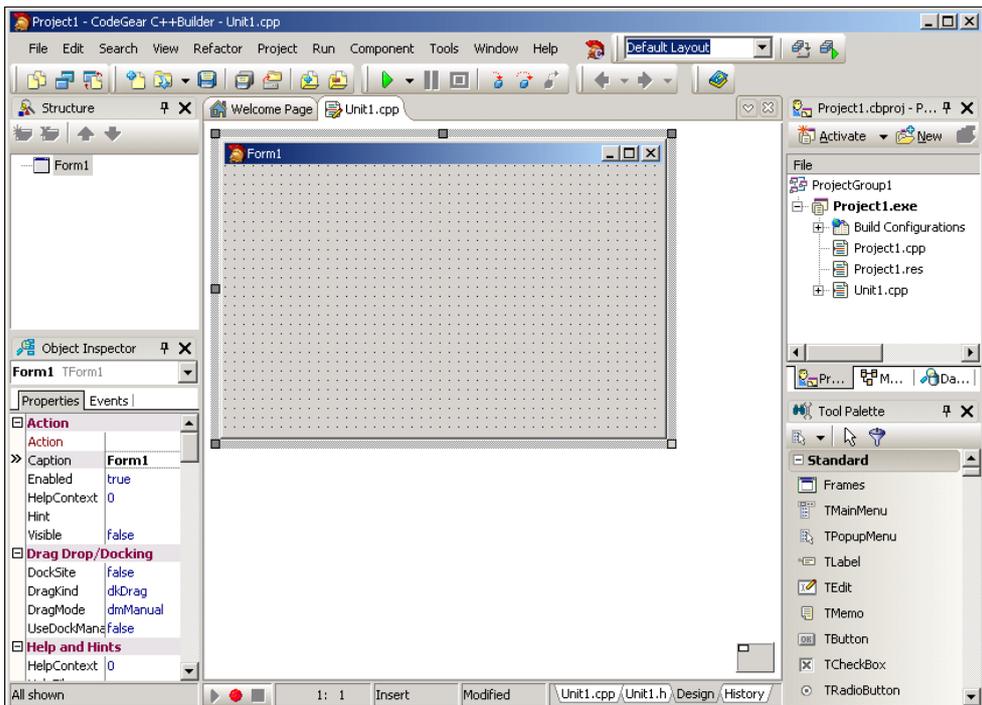


Рис. 1.1. Окно C++ Builder в начале работы над новым проектом

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

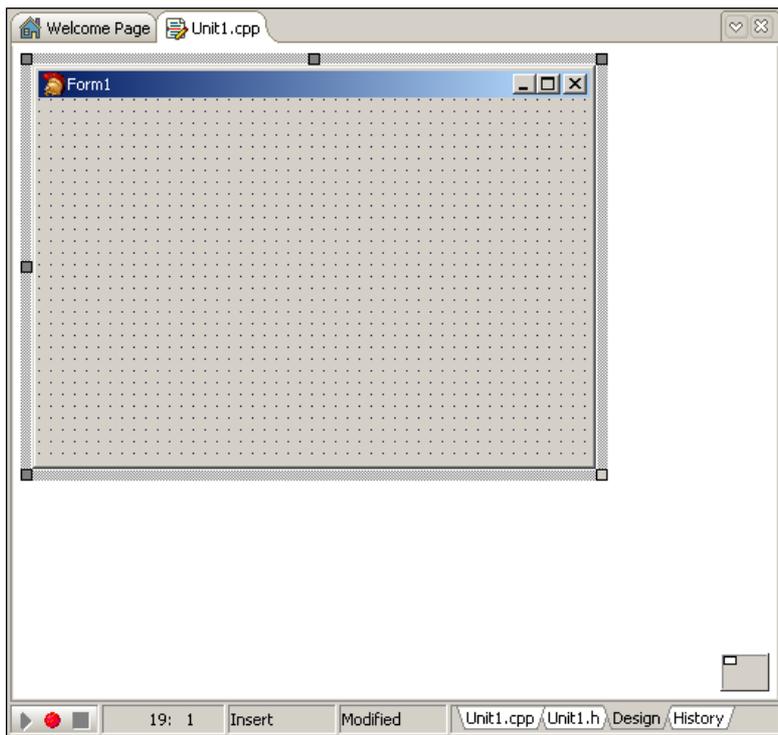


Рис. 1.2. Окно дизайнера формы

За окном дизайнера формы находится окно редактора кода (рис. 1.3), доступ к которому можно получить, сделав щелчок на находящемся в нижней части окна дизайнера формы ярлыке **Unit1.cpp** (главный модуль формы) или **Unit1.h** (заголовочный файл модуля формы) или нажав клавишу <F12>. Для того чтобы вновь стало доступно окно редактора формы, надо сделать щелчок на ярлыке **Design** или нажать клавишу <F12>.

Слева от окна дизайнера формы находится окно **Object Inspector**. В верхней части окна указано имя выбранного в данный момент объекта, на вкладке **Properties** (рис. 1.4, а) перечислены свойства объекта, а на вкладке **Events** (рис. 1.4, б) — события, на которые объект может реагировать.

Вкладка **Properties** (свойства) используется для редактирования (изменения) значений свойств. Свойство — это характеристика *объекта* (формы, командной кнопки, поля редактирования и т. д.). Свойства определяют вид объекта,

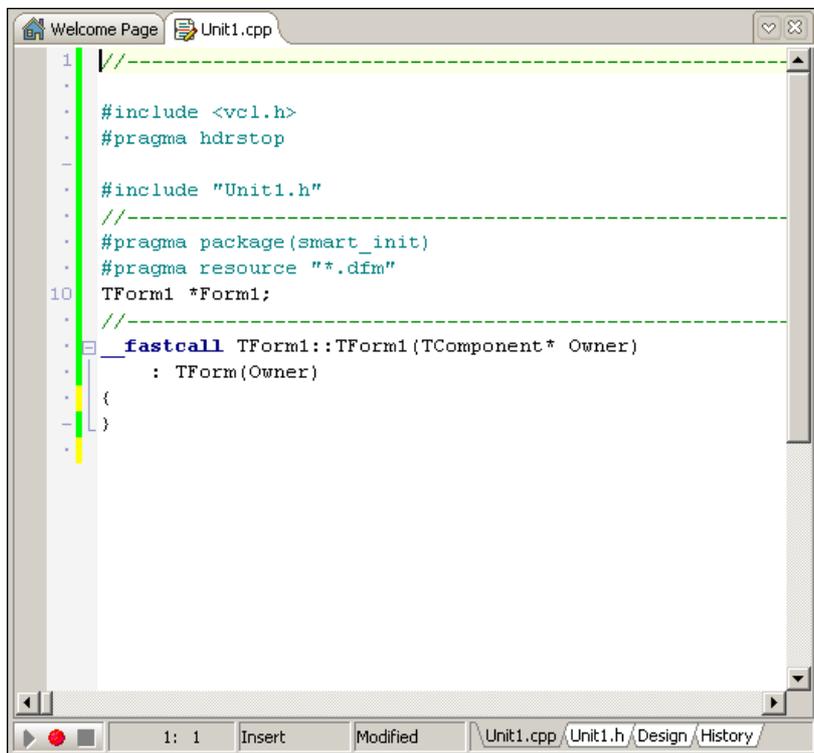


Рис. 1.3. Окно редактора кода

его положение относительно других объектов, а также поведение (реакцию на действия пользователя). Например, свойство `Caption` формы определяет текст, который отображается в заголовке формы, а свойства `Width` и `Height` — ее размер (ширину и высоту). Значения свойств указаны в правой колонке. По умолчанию свойства на вкладке **Properties** объединены по функциональному признаку 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).

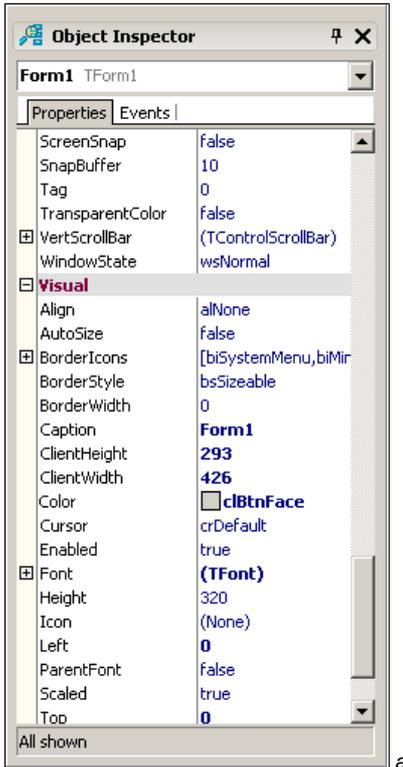
На вкладке **Events** перечислены *события*, которые может воспринимать объект (строго говоря, на вкладке **Events** перечислены свойства, значения которых определяют реакцию объекта на соответствующие события).

На вкладках окна **Tool Palette** (рис. 1.5) находятся компоненты.

Компонент — это объект, обеспечивающий решение некоторой задачи. Например, на вкладке **Standard** находятся компоненты пользовательского

интерфейса (Label — поле отображения текста, Button — командная кнопка, Edit — поле редактирования и др.), а на вкладке **dbGo** находятся компоненты, обеспечивающие доступ к базам данных.

В окне **Project Manager** (рис. 1.6) отображается структура проекта, над которым в данный момент идет работа (в простейшем случае проект образуют главный модуль (сpp-файл), модуль формы (dfm-, h- и сpp-файлы) и файл ресурсов (res-файл).



а



б

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

Программист может расположить окна так, как ему удобно, может изменить размер окна или сделать его всплывающим. Перемещение и изменение размера окна выполняются обычным способом — перетаскиванием области заголовка (перемещение окна) или границы (изменение размера). Чтобы окно стало всплывающим (появлялось при позиционировании мыши на его заголовке), надо сделать щелчок на находящейся в его заголовке изображении канцелярской кнопки.

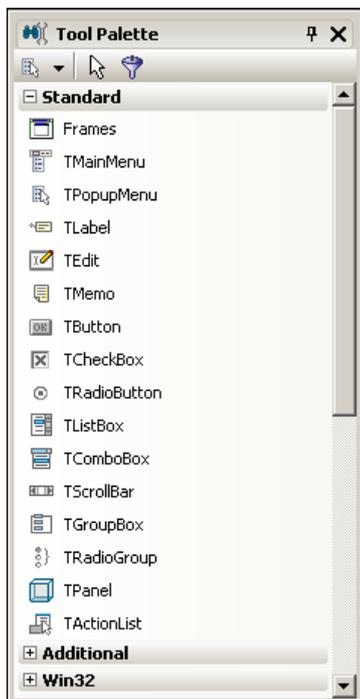


Рис. 1.5. Вкладка **Standard** содержит компоненты, обеспечивающие взаимодействие пользователя с программой

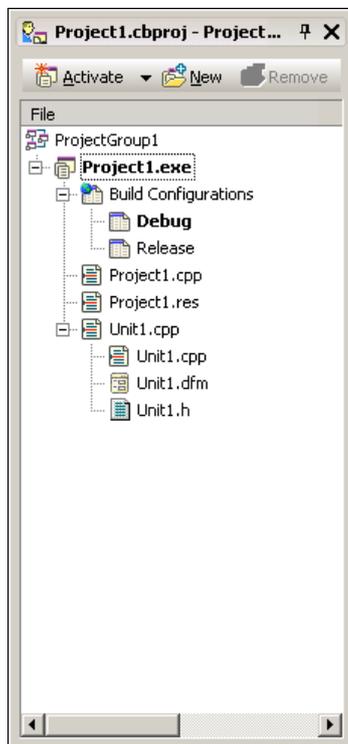


Рис. 1.6. В окне **Project Manager** отображается структура проекта

Иногда возникает необходимость восстановить исходное (стандартное) расположение окон. Чтобы это сделать, надо раскрыть список **Desktop speedsetting** и выбрать **Default Layout** (рис. 1.7).

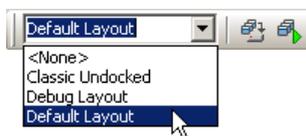
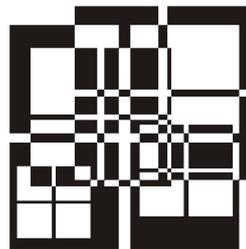


Рис. 1.7. Чтобы восстановить исходное расположение окон, выберите **Default Layout**

ГЛАВА 2



Первый проект

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

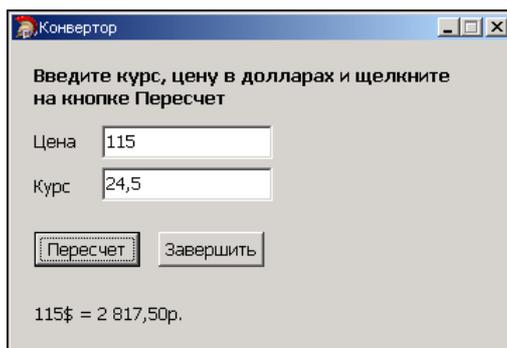


Рис. 2.1. Окно программы Конвертор

Начало работы

Чтобы начать работу над новым приложением, нужно в меню **File** выбрать команду **New ▶ VCL Forms Application — C++ Builder**. В результате будет создан новый проект: модуль формы (файлы Unit1.cpp, Unit1.h, Unit1.dfm), главный модуль (файл Project1.cpp) и файл ресурсов (Project1.res). Форма, которая отображается в окне дизайнера формы (на вкладке **Design**) — это окно программы, которое появится на экране в результате запуска программы.

Форма

Работа над приложением начинается с настройки стартовой формы путем изменения значений ее свойств (табл. 2.1).

Таблица 2.1. Свойства формы (объекта *TForm*)

Свойство	Описание
Name	Имя (идентификатор) формы. Используется для доступа к форме, ее свойствам и методам, а также для доступа к компонентам, которые находятся в форме
Caption	Текст заголовка
Width	Ширина формы
Height	Высота формы
ClientWidth	Ширина внутренней области формы (без учета ширины вертикальных границ)
ClientHeight	Высота внутренней области формы (без учета ширины горизонтальных границ и высоты заголовка)
Position	Положение окна в момент первого его появления на экране: <code>poDesktopCenter</code> — в центре рабочего стола; <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>
Top	Расстояние от верхней границы формы до верхней границы экрана
Left	Расстояние от левой границы формы до левой границы экрана
BorderStyle	Вид границы. Граница может быть обычной (<code>bsSizeable</code>), тонкой (<code>bsSingle</code>) или отсутствовать (<code>bsNone</code>). Если у окна обычная граница, то во время работы программы пользователь сможет 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
BorderIcons	Кнопки управления окном. Значение свойства определяет кнопки управления окном, которые будут доступны пользователю во время работы программы.

Таблица 2.1 (окончание)

Свойство	Описание
	<p>Значение свойства задается путем присвоения значений уточняющим свойствам:</p> <p><code>biSystemMenu</code> — определяет доступность кнопки системного меню;</p> <p><code>biMinimize</code> — определяет доступность кнопки Свернуть;</p> <p><code>biMaximize</code> — определяет доступность кнопки Развернуть;</p> <p><code>biHelp</code> — определяет доступность кнопки отображения справочной информации</p>
Icon	Значок в заголовке окна
Color	<p>Цвет фона. Цвет можно задать, указав название цвета (<code>clSilver</code>, <code>clWhite</code> и др.) или привязку к текущей цветовой схеме операционной системы (<code>clBtnFace</code>, <code>clBtnText</code> и др.). По умолчанию используется второй способ, при котором цвет определяется текущей цветовой схемой, выбранным компонентом привязки и меняется при изменении цветовой схемы операционной системы</p>
Font	<p>Шрифт. Шрифт, используемый "по умолчанию" компонентами, находящимися на поверхности формы. Изменение свойства <code>Font</code> формы приводит к автоматическому изменению свойства <code>Font</code> компонента, располагающегося на поверхности формы, т. е. компоненты наследуют свойство <code>Font</code> от формы (имеется возможность запретить наследование)</p>

Для изменения значений свойств формы (и других объектов) используется вкладка **Properties** окна **Object Inspector**. В левой колонке вкладки перечислены свойства *выбранного* объекта, в правой — указаны значения его свойств.

По умолчанию свойства на вкладке **Properties** объединены в группы по функциональному признаку (названия групп выделены цветом). Например, в группу **Visual** объединены свойства, определяющие вид объекта (для формы — заголовок, цвет фона, вид границы).

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**.

Изменить значение свойства можно путем непосредственного редактирования текущего значения или путем выбора из списка.

Чтобы изменить текст в заголовке формы, надо изменить значение свойства `Caption`. Для этого нужно в окне **Object Inspector** щелкнуть левой кнопкой мыши в строке свойства `Caption` (в результате будет активизирован режим редактирования значения свойства, появится курсор), ввести текст нового значения и нажать клавишу `<Enter>` (рис. 2.2).

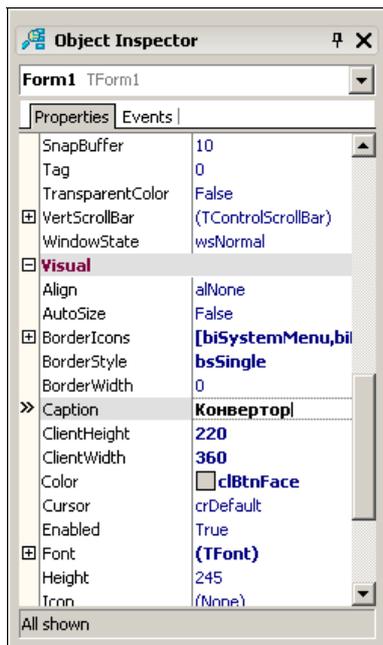


Рис. 2.2. Изменение значения свойства `Caption` путем ввода значения

Аналогичным образом можно установить значения других свойств, например, `Width` и `Height`, которые определяют размер (ширину и высоту) формы.

Форма — это обычное окно. Поэтому размер формы можно изменить точно так же, как и любого другого 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 можно задать, установив значение свойств `Top` (отступ от верхней границы экрана) и `Left` (отступ от левой границы экрана) или свойства `Position`.

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

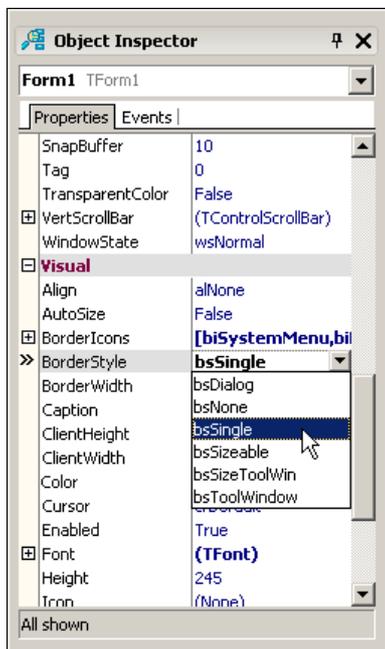


Рис. 2.3. Установка значения свойства путем выбора из списка

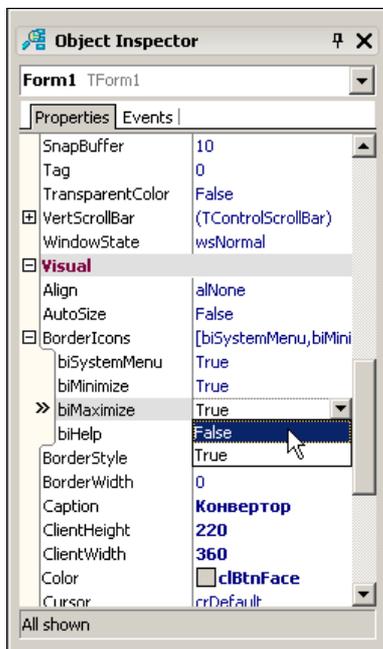


Рис. 2.4. Изменение значения уточняющего свойства

Некоторые свойства являются сложными. Например, свойство `BorderIcons` задает кнопки управления окном, которые будут доступны во время работы программы. Значение этого свойства определяется 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 которого 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 свойства `Font`, в поле значения свойства отображается кнопка, на которой видны 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).

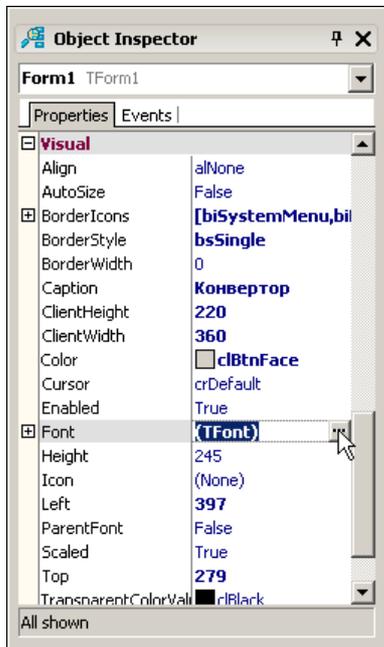


Рис. 2.5. Чтобы задать свойства шрифта, щелкните на кнопке с тремя точками

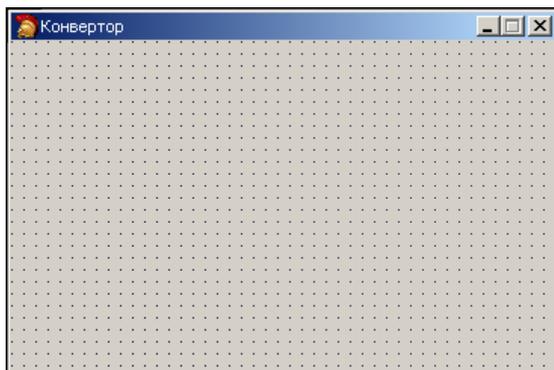


Рис. 2.6. Так должна выглядеть форма после ее настройки

В табл. 2.2 приведены значения свойств стартовой формы программы **Конвертор** (значения других свойств оставлены без изменения и поэтому в таблице не представлены). В таблице в именах некоторых свойств есть точка. Это значит, что надо задать значение уточняющего свойства. После того как будут установлены значения свойств формы, она должна выглядеть так, как показано на рис. 2.6.

Таблица 2.2. Значения свойств стартовой формы программы **Конвертор**

Свойство	Значение	Комментарий
Caption	Конвертор	
Width	360	
Height	245	
BorderStyle	bsSingle	Тонкая граница. Во время работы программы пользователь не сможет изменить размер окна путем перемещения его границы

Таблица 2.2 (окончание)

Свойство	Значение	Комментарий
BorderIcons.biMaximize	false	В заголовке окна не отображать кнопку Развернуть . Во время работы программы пользователь не сможет развернуть окно на весь экран
Font.Name	Tahoma	
Font.Size	9	
Position	poDesktopCenter	В момент запуска программы окно появится в центре рабочего стола

Компоненты

Различают два типа компонентов: визуальные и невидимые.

Визуальными называют компоненты, отображаемые в окне программы во время ее работы. Поле редактирования (`Edit`), поле отображения текста (`Label`), кнопка (`Button`), список (`ListBox`), поле отображения иллюстрации (`Image`) — все это примеры визуальных компонентов. Имеется библиотека визуальных компонентов — `Visual Component Library`. Именно поэтому приложения, использующие VCL-компоненты, называют VCL-приложениями.

Невидимые компоненты отображаются только на форме во время разработки программы. Таймер (`Timer`) — типичный невидимый компонент.

Компоненты, которые программист может использовать при разработке программ, находятся на вкладках *палитры компонентов* (окно **Tool Palette**). На вкладках **Standard**, **Additional** и **Win32** находятся компоненты пользовательского интерфейса. Вкладка **Dialogs** содержит компоненты, обеспечивающие отображение стандартных диалогов (**Открыть**, **Сохранить** и др.). Вкладки **Data Access**, **Data Controls**, **dbExpress**, **dbGo** содержат компоненты, обеспечивающие работу с базами данных.

Рассматриваемая нами программа пересчета цены из долларов в рубли должна получить от пользователя исходные данные: цену в долларах и курс. Ввод данных с клавиатуры обеспечивает компонент `Edit`. Поэтому в форму разрабатываемого приложения нужно добавить два компонента `Edit`.

Для того чтобы в форму добавить компонент `Edit`, надо:

1. В палитре компонентов раскрыть вкладку **Standard**.
2. Сделать щелчок левой кнопкой мыши на значке компонента `Edit` (рис. 2.7).

3. Сделать щелчок левой кнопкой мыши в той точке формы, в которой должен быть левый верхний угол компонента.

В результате выполнения описанных выше действий на форме должен появиться компонент `Edit` — поле редактирования (рис. 2.8).

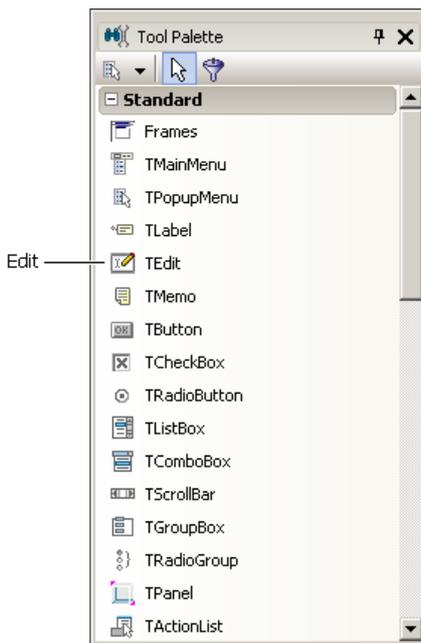


Рис. 2.7. Компонент `Edit` — поле редактирования

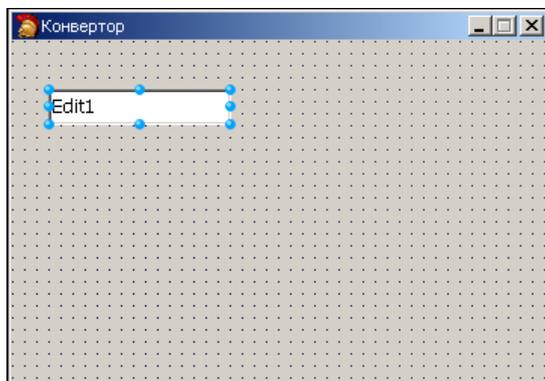


Рис. 2.8. Результат добавления на форму компонента `Edit`

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

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

Таблица 2.3. Свойства компонента `Edit`

Свойство	Описание
<code>Name</code>	Имя (идентификатор) компонента
<code>Text</code>	Текст, который находится в поле редактирования
<code>Left</code>	Расстояние от левой границы компонента до левой границы формы
<code>Top</code>	Расстояние от верхней границы компонента до верхней границы формы
<code>Height</code>	Высота компонента
<code>Width</code>	Ширина компонента
<code>Font</code>	Шрифт, используемый для отображения текста в поле компонента
<code>ParentFont</code>	Признак наследования шрифта от формы. Если значения свойства равно <code>true</code> , то для отображения текста в поле компонента используется шрифт формы
<code>MaxLength</code>	Количество символов, которое можно ввести в поле редактирования. Если значение свойства равно нулю, ограничения на количество символов нет

На рис. 2.9 приведен вид формы после добавления двух полей редактирования. Один из компонентов *выбран* (выделен), помечен восьмью 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 имя компонента 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).

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

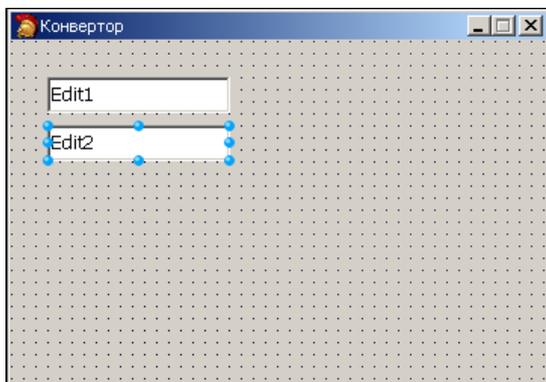


Рис. 2.9. Форма с двумя компонентами

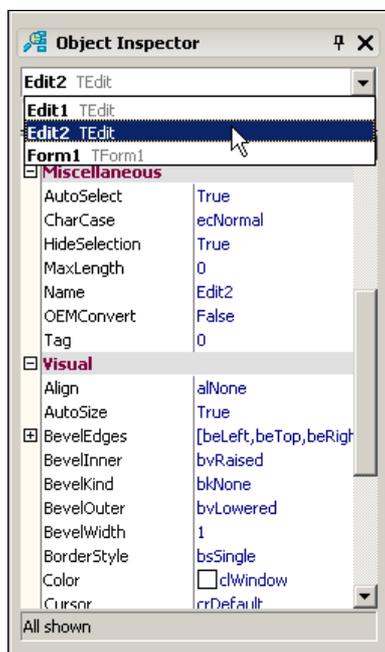


Рис. 2.10. Выбор компонента в окне Object Inspector

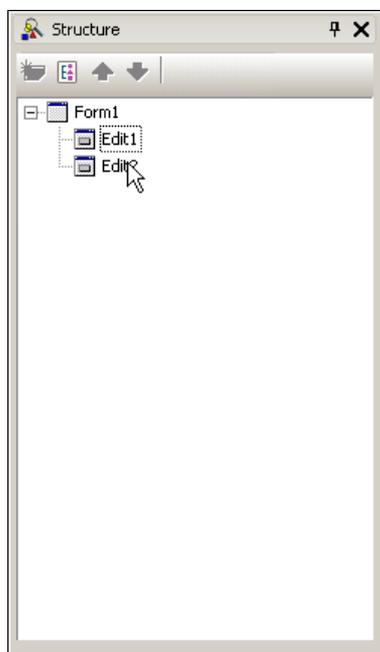


Рис. 2.11. Выбор компонента в окне Structure

Для того 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`).