



ДИСТРИБУТИВ
Turbo C++
Explorer

Никита Культин

ОСНОВЫ ПРОГРАММИРОВАНИЯ В TURBO C++



Среда разработки

Программирование графики,
мультимедиа и баз данных

Создание справочной системы
и установочного CD

+ CD



Никита Культин

**ОСНОВЫ
ПРОГРАММИРОВАНИЯ
В TURBO C++**

Санкт-Петербург
«БХВ-Петербург»
2007

УДК 681.3.068+800.92TurboC++
ББК 32.973.26-018.1
К90

Кульгин Н. Б.

К90 Основы программирования в Turbo C++. — СПб.: БХВ-Петербург, 2007. — 464 с.: ил. + CD-ROM
ISBN 978-5-9775-0062-3

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

Компакт-диск содержит дистрибутив Turbo C++ Explorer, размещенный с разрешения Borland Software Corporation, а также проекты, рассматриваемые в книге.

Для начинающих программистов

УДК 681.3.068+800.92TurboC++
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>Натальи Смирновой</i>
Корректор	<i>Наталья Перишакова</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 28.12.06.

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

Тираж 3000 экз. Заказ №

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

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

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

ISBN 978-5-9775-0062-3

© Кульгин Н. Б., 2007
© Оформление, издательство "БХВ-Петербург", 2007

Оглавление

Предисловие	1
Turbo C++ — что это?.....	1
Об этой книге.....	2
Часть I. Turbo C++	3
Глава 1. Среда разработки Turbo C++	5
Установка.....	5
Регистрация.....	6
Первое знакомство	7
Глава 2. Первый проект	12
Начало работы.....	12
Форма.....	13
Компоненты	18
Событие и функция обработки события.....	29
Редактор кода.....	34
Система подсказок.....	34
Шаблоны кода.....	36
Справочная информация	37
Сохранение проекта	38
Структура проекта	39
Компиляция.....	43
Компоновка	46
Запуск программы.....	47
Исключения.....	47
Обработка исключений	48
Внесение изменений.....	51
Настройка приложения.....	57
Установка приложения на другом компьютере.....	59
Глава 3. Базовые компоненты	61
<i>Label</i>	61
<i>Edit</i>	64
<i>Button</i>	68
<i>CheckBox</i>	72
<i>RadioButton</i>	75
<i>ComboBox</i>	78

<i>ListBox</i>	82
<i>Memo</i>	87
<i>Timer</i>	90
<i>Panel</i>	93
<i>ControlBar</i>	94
<i>SpeedButton</i>	95
<i>StatusBar</i>	99
<i>UpDown</i>	103
<i>TrayIcon</i>	108
<i>ProgressBar</i>	111
<i>Image</i>	114
<i>MainMenu</i>	120
<i>PopupMenu</i>	128
<i>OpenDialog</i>	130
<i>SaveDialog</i>	132

Часть II. ПРАКТИКУМ ПРОГРАММИРОВАНИЯ 137

Глава 4. Графика..... 139

Графическая поверхность.....	139
Карандаш и кисть.....	141
Графические примитивы.....	143
Текст.....	144
Линия.....	147
Ломаная линия.....	154
Прямоугольник.....	155
Многоугольник.....	160
Точка.....	166
Битовые образы.....	167
Загрузка из файла.....	168
Загрузка из ресурса.....	173
Анимация.....	176
Движение.....	176
Взаимодействие с пользователем.....	182
Использование битовых образов.....	189

Глава 5. Мультимедиа 197

Функция <i>PlaySound</i>	197
Компонент <i>MediaPlayer</i>	198
Простой mp3-плеер.....	202
Проигрыватель CD.....	208
Воспроизведение MIDI-музыки.....	216
Просмотр видеороликов.....	220
Компонент <i>Animate</i>	227
Создание анимации.....	230

Глава 6. Базы данных.....	236
База данных и СУБД.....	236
Локальные и удаленные базы данных.....	236
Структура базы данных.....	237
Механизмы доступа к данным.....	238
Компоненты доступа к данным.....	239
Создание базы данных.....	239
Программа работы с базой данных.....	239
Доступ к данным.....	240
Отображение данных.....	245
Выбор информации из базы данных.....	250
Работа с базой данных в режиме формы.....	256
Сервер InterBase.....	264
Установка.....	264
Запуск сервера.....	264
Утилиты gsec и isql.....	265
Подключение к серверу.....	265
Регистрация пользователя.....	266
Создание базы данных.....	267
Создание таблицы.....	268
Доступ к базе данных.....	268
Сценарии.....	270
Приложение работы с базой данных InterBase.....	271
Установка программы работы с базой данных на другой компьютер.....	279
Глава 7. Компонент программиста.....	280
Модуль компонента.....	281
Тестирование компонента.....	291
Установка компонента.....	295
Ресурсы компонента.....	295
Создание пакета.....	297
Установка пакета.....	298
Тестирование компонента.....	299
Установка программы на другом компьютере.....	302
Распространение компонента.....	303
Глава 8. Справочная информация.....	304
Справочная система HTML Help.....	304
Подготовка справочной информации.....	304
Создание chm-файла.....	308
Файл проекта.....	308
Файл контента (оглавление).....	311
Идентификаторы разделов.....	313
Компиляция.....	315
Отображение справочной информации.....	316

Глава 9. Создание установочного диска	319
Программа InstallShield Express	319
Начало работы.....	320
Структура	321
Выбор устанавливаемых компонентов	324
Конфигурирование системы пользователя	325
Настройка диалогов.....	327
Требования к системе.....	330
Создание образа установочного CD	331
Глава 10. Примеры программ	333
Экзаменатор	333
Требования к программе	334
Файл теста.....	334
Форма	337
Отображение иллюстраций.....	338
Доступ к файлу теста.....	338
Текст программы.....	340
Сапер	353
Правила и представление данных	353
Форма.....	356
Игровое поле	357
Начало игры	358
Игра	362
Справочная информация	366
Информация о программе	368
Текст программы.....	371
MP3-плеер	382
Форма	382
Регулятор громкости.....	385
Перемещение окна.....	386
Заключение	395
ПРИЛОЖЕНИЯ	397
Приложение 1	399
Форма.....	399
Базовые компоненты	401
<i>Animate</i>	401
<i>Button</i>	402
<i>CheckBox</i>	402
<i>ComboBox</i>	404
<i>Edit</i>	405

<i>Image</i>	406
<i>Label</i>	407
<i>ListBox</i>	408
<i>MediaPlayer</i>	409
<i>Memo</i>	410
<i>OpenDialog</i>	411
<i>RadioButton</i>	412
<i>SaveDialog</i>	413
<i>SpeedButton</i>	414
<i>StringGrid</i>	415
<i>Timer</i>	417
<i>UpDown</i>	417
Компоненты доступа-манипулирования данными	418
<i>ADOConnection</i>	418
<i>ADODataSet</i>	419
<i>ADOQuery</i>	420
<i>ADOTable</i>	421
<i>DataSource</i>	422
<i>DBEdit, DBMemo, DBText</i>	422
<i>DBGrid</i>	423
<i>DBNavigator</i>	424
Графика	426
<i>PaintBox</i>	426
<i>Canvas</i>	426
<i>Brush</i>	429
<i>Pen</i>	430
Цвет	430
Функции	431
Функции ввода и вывода	431
Математические функции	432
Функции преобразования	433
Функции манипулирования датами и временем	434
События	436
Исключения	436
Приложение 2. Описание компакт-диска	438
Рекомендуемая литература	444
Предметный указатель	445

Предисловие

Turbo C++ — что это?

Среда разработки Borland C++ Builder является одним из популярнейших инструментов разработки прикладных программ (приложений) для Windows. Она ориентирована на так называемую "быструю" разработку, в основе которой лежит технология визуального проектирования и событийного программирования (суть этой технологии, которую часто называют "компонентной", в том, что среда берет на себя большую часть рутины, оставляя программисту работу по конструированию окон и созданию функций обработки событий).

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. Здесь необходимо отметить, что существенным недостатком прекрасного во всех отношениях инструмента Borland Developer Studio является его высокая цена для рядового программиста (школьника, студента). Теперь программистам вновь стала доступна среда разработки, в которой в качестве языка программирования используется C++, и называется она Turbo C++ (фактически Turbo C++ — это C++ Builder, извлеченный из Borland Developer Studio).

Среда Turbo C++ доступна в двух вариантах: Turbo C++ Explorer и Turbo C++ Professional. Версия Turbo C++ Explorer предназначена исключительно для целей обучения и предоставляется фирмой Borland всем желающим бесплатно. Версия Turbo C++ Professional является полноценным средством разработки программного обеспечения со всеми вытекающими отсюда последствиями — необходимостью легального приобретения (покупки) и регистрации. Принципиальных отличий, с точки зрения процесса разработки, между пакетами Explorer и Professional нет. Разница состоит в наборе компонентов и функциональных возможностей, которые среда предоставляет программисту. Например, в версию Professional включены Rave Reports (средство подготовки отчетов) и Remote Debugger (отладчик приложений, работающих на другом компьютере), которых в версии Explorer нет. Также при работе в Turbo C++ Explorer программист может использовать только стандартные компоненты, добавить другие компоненты нельзя.

Turbo C++ может работать в среде операционных систем Microsoft Windows 2000 Professional, Microsoft Windows XP Professional или Microsoft Windows

Server 2003. Особых требований (по современным меркам) к ресурсам компьютера Turbo C++ не предъявляет: процессор должен быть класса Pentium III с тактовой частотой 1,4 ГГц (рекомендуется Pentium IV с частотой 2,0 ГГц и выше), оперативная память — 512 Мбайт (рекомендуется 1 Гбайт и больше), достаточное количество свободного дискового пространства (для установки Turbo C++ необходимо приблизительно 600 Мбайт и 750 Мбайт — для установки Microsoft .NET Framework).

Об этой книге

Книга, которую вы держите в руках, — это не описание среды разработки или языка программирования. Это пособие по разработке программ для Windows (Win32-приложений) в среде Turbo C++. В книге представлена технология визуального проектирования и событийного программирования, показаны возможности среды разработки, на конкретных примерах рассмотрен процесс создания программ — от разработки диалогового окна и функций обработки событий до создания справочной системы и установочного диска.

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

Научиться программировать можно, только программируя в процессе решения конкретных задач. Поэтому, чтобы получить максимальную пользу от книги, вы должны работать с ней активно. Изучайте листинги, старайтесь понять, как работают программы. Не бойтесь экспериментировать — вносите изменения в приведенные примеры, совершенствуйте их. Чем больше вы сделаете самостоятельно, тем большему научитесь!



ЧАСТЬ I

TURBO C++

Глава 1



Среда разработки Turbo C++

Установка

Turbo C++ можно установить с CD, прилагаемого к книге.

Перед тем как приступить к непосредственной установке Turbo C++, следует установить компоненты, необходимые для работы Turbo C++, а именно:

- Microsoft Internet Explorer 6.0 SP1;
- Microsoft .NET Framework v1.1;
- Microsoft .NET Framework SDK v1.1;
- Microsoft Visual J# .NET v1.1;
- Microsoft XML Core Services v4.0 SP2.

Дистрибутивы перечисленных выше компонентов есть на прилагаемом к книге CD (архив prerequisites.zip). Сначала надо распаковать архив prerequisites.zip во временный каталог (дистрибутив каждого компонента будет помещен в отдельный каталог), затем активизировать процесс установки (табл. 1.1). Обратите внимание, что компоненты следует устанавливать в том порядке, в котором они перечислены в таблице.

Таблица 1.1. Компоненты и программы, обеспечивающие установку

Компонент	Каталог дистрибутива	Программа
Microsoft Internet Explorer 6.0 SP1	ie60sp1	ie6setup.exe
Microsoft .NET Framework v1.1	dotNETRedist	dotnetfx.exe
Microsoft .NET Framework SDK v1.1	dotNETSDK	setup.exe

Таблица 1.1 (окончание)

Компонент	Каталог дистрибутива	Программа
Microsoft Visual J# .NET v1.1	dotNETJSharp	vjredist.exe
Microsoft XML Core Services v4.0 SP2	Msxml	msxml.msi

После того как подготовительная работа будет выполнена, можно приступить к установке Turbo C++, т. е. запустить с прилагаемого к книге CD файл TurboCPP.exe, который представляет собой самораспаковывающийся архив. После распаковки содержимого архива в каталог C:\Document and Settings\User\Local Settings\Temp\TurboCPP на экране появится окно **Borland Turbo Products**, в котором надо сделать щелчок на кнопке **Install**. На экране появится окно **Turbo C++ CD Browser**, в котором для активизации установки надо щелкнуть кнопку **Install Borland Turbo C++**. Необходимо отметить, что если по какой-либо причине процесс установки будет прерван (например, из-за того, что программа установки обнаружит, что на компьютере не установлен один из перечисленных в табл. 1.1 компонентов), то повторно активизировать процесс установки можно, запустив программу install.exe, которая находится в каталоге C:\Document and Settings\User\Local Settings\Temp\TurboCPP. Также следует обратить внимание, что на компьютер можно установить *только один* из продуктов линейки Turbo Explorer, другими словами, если на компьютере уже установлена какая-либо среда разработки Turbo (например, Turbo Delphi Explorer), то установить Turbo C++ Explorer уже не получится.

Регистрация

Borland требует регистрации всех своих продуктов, в т. ч. и предоставляемых бесплатно. Чтобы зарегистрировать Turbo C++, надо зайти на страницу www.borland.com/downloads и выбрать ссылку **Turbo**. На следующей странице в таблице **Keys Only** следует выбрать ссылку **Turbo C++ Explorer**. В результате описанных действий станет доступна форма **Membership services login**, в которой, для начала процесса регистрации, надо выбрать ссылку **Create new user account**. По окончании процесса регистрации на адрес, указанный в регистрационной форме, придет письмо с прикрепленным TXT-файлом. Этот файл надо поместить в каталог C:\Document and Settings\User, где User — зарегистрированное в Windows имя пользователя. На этом процесс установки Turbo C++ можно считать завершенным.

Первое знакомство

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

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

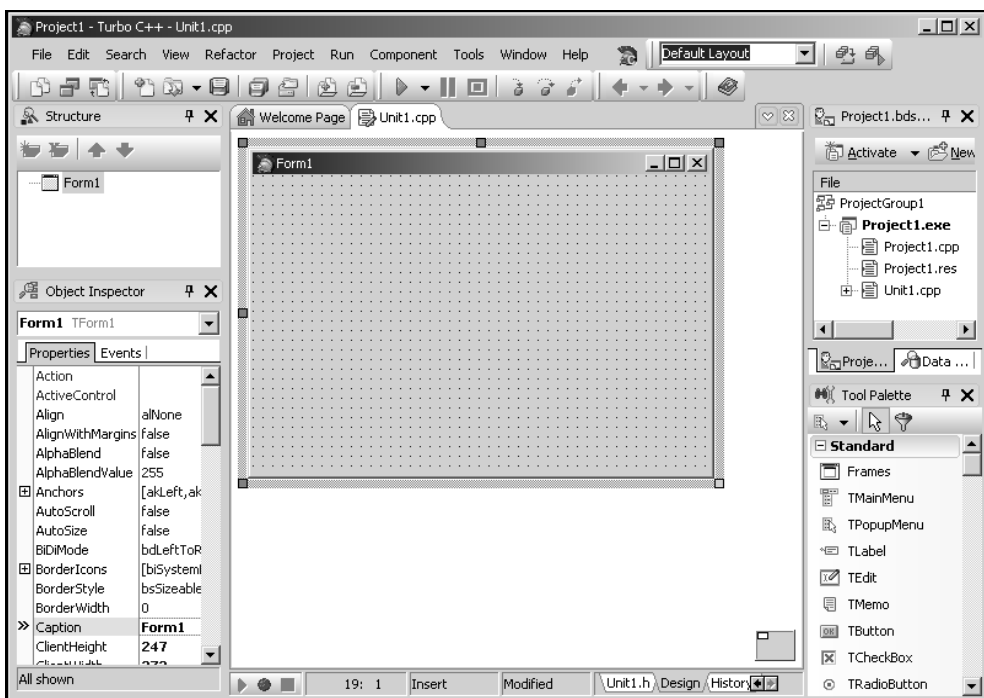


Рис. 1.1. Окно Turbo C++
в начале работы над новым проектом

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

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 ярлычке **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 ярлычке **Design** или нажать клавишу <F12>.

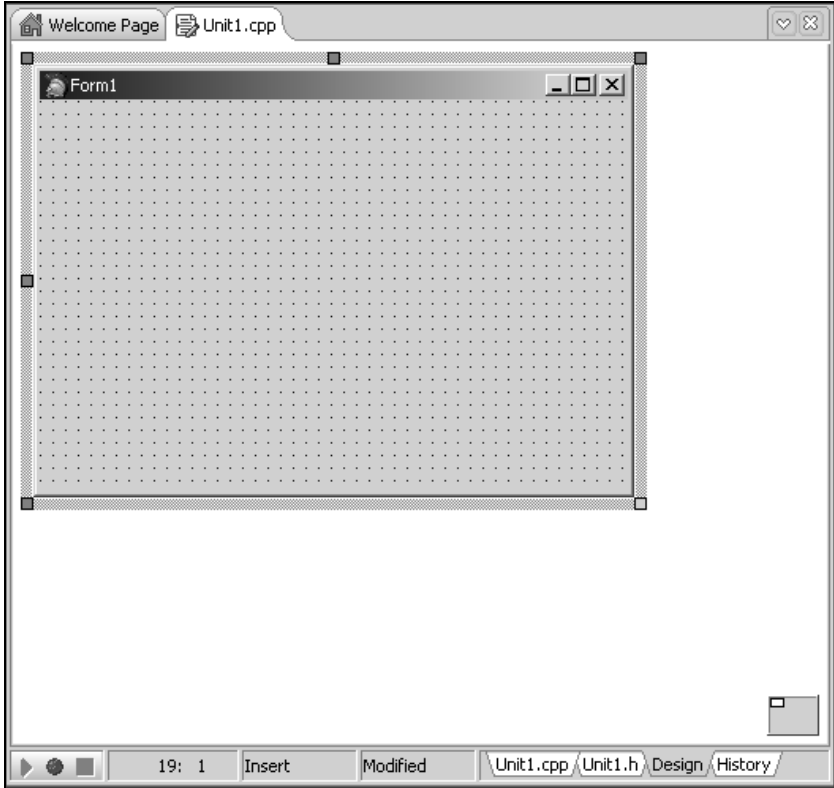


Рис. 1.2. Окно дизайнера формы

Слева от окна дизайнера формы находится окно **Object Inspector** (рис. 1.4). В верхней части окна указано имя объекта, *свойства* которого отображаются в окне **Object Inspector** в данный момент. Вкладка **Properties** (свойства) используется для редактирования (изменения) значений свойств объектов. Свойство — это характеристика *объекта* (формы, командной кнопки, поля редактирования и т. д.). Значения свойств определяют вид объекта, его положение относительно других объектов, а также поведение (реакцию на действия пользователя). Например, значение свойства `Caption` формы определяет текст, который отображается в заголовке формы, а

поле редактирования и др.), а на вкладке **dbGo** — компоненты, обеспечивающие доступ к базам данных.

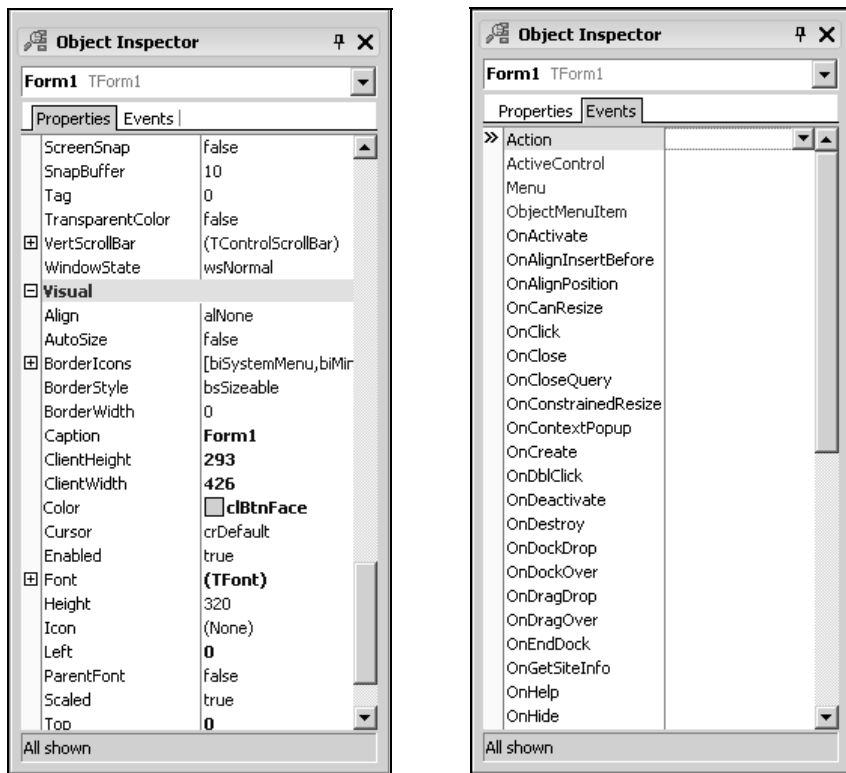


Рис. 1.4. В окне **Object Inspector**

на вкладке **Properties** перечислены свойства объекта, на вкладке **Events** — события, на которые объект может реагировать

В окне **Project Manager** (рис. 1.6) отображается структура приложения (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).

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

Иногда возникает необходимость восстановить исходное (стандартное) расположение окон. Чтобы это сделать, надо раскрыть список **Desktop speedsetting** (по умолчанию этот список находится в строке главного меню, после меню **Help**) и выбрать **Default Layout** (рис. 1.7).

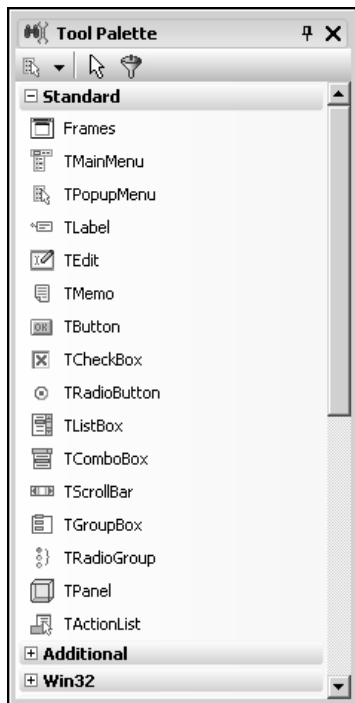


Рис. 1.5. Вкладка **Standard** содержит компоненты, обеспечивающие взаимодействие пользователя с программой

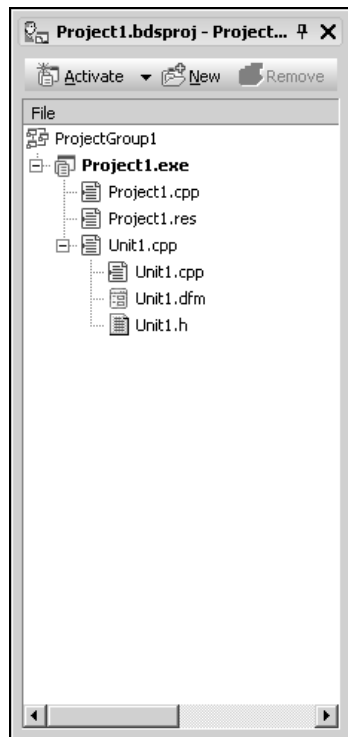


Рис. 1.6. В окне **Project Manager** отображается структура проекта

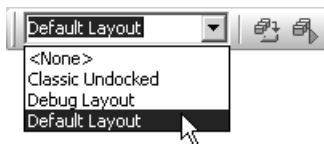


Рис. 1.7. Чтобы восстановить исходное расположение окон, выберите **Default Layout**

Глава 2



Первый проект

Процесс разработки программы в Turbo C++ 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.

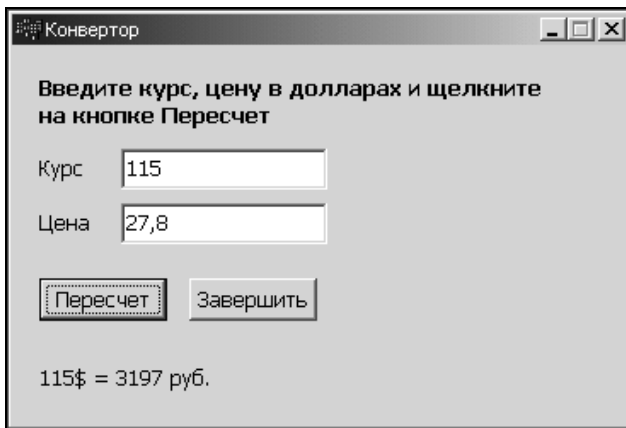


Рис. 2.1. Окно программы **Конвертор**

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 (Pro-

ject1.res). Форма, которая отображается в окне дизайнера формы (на вкладке **Design**), — это заготовка окна программы, которое появится на экране в результате ее запуска.

Форма

Работа над приложением начинается с настройки 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 формы
Caption	Текст заголовка
Width	Ширина формы
Height	Высота формы
Position	Положение окна в момент первого его появления на экране (poCenterScreen — в центре экрана; poOwnerFormCenter — в центре родительского окна; poDesigned — положение окна определяют значения свойств Top и Left)
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 (bsSizeable), тонкой (bsSingle) или отсутствовать (bsNone). 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
BorderIcons	Кнопки управления окном. Значение свойства определяет, какие кнопки управления окном будут доступны пользователю во время работы программы (задается путем присвоения значений уточняющим свойствам: biSystemMenu, biMinimize, biMaximize и biHelp).

Таблица 2.1 (окончание)

Свойство	Описание
	Свойство <code>biSystemMenu</code> определяет доступность кнопки системного меню (пиктограммы 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
<code>Font</code>	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. То есть компоненты наследуют свойство <code>Font</code> от формы (имеется возможность запретить наследование)

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 — заголовков, 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 надо 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 <Enter> (рис. 2.2).

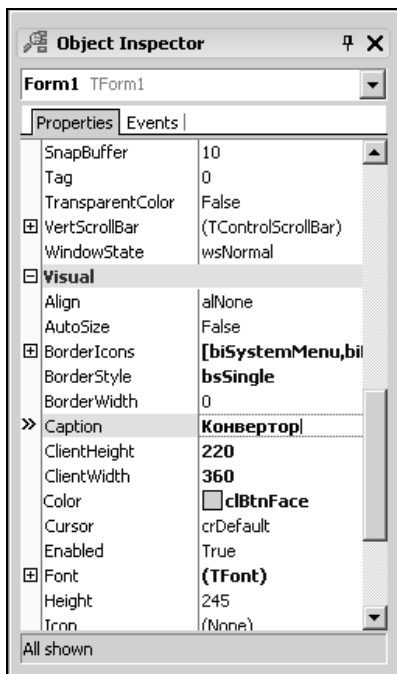


Рис. 2.2. Изменение значения свойства Caption путем ввода значения

Аналогичным образом можно установить значения других свойств, например, Height и Width, которые определяют размер (высоту и ширину) формы. Свойству Height надо присвоить значение 200, свойству Width — значение 330.

Форма — это обычное окно. Поэтому размер формы можно изменить 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 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 отображается значок раскрывающегося списка.

Очевидно, что значение таких свойств можно задать путем выбора из списка (рис. 2.3).

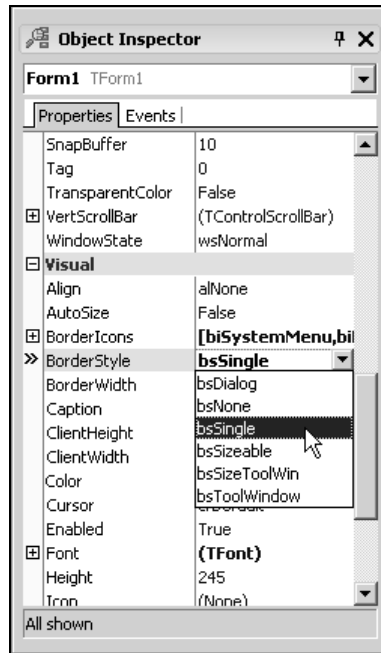


Рис. 2.3. Установка значения свойства путем выбора из списка

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. 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 на котором раскрывается список уточняющих свойств (рис. 2.4). Значение уточняющего свойства можно задать обычным образом (ввести значение в поле редактирования или выбрать из списка).

В результате выбора некоторых свойств (например, свойства `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).

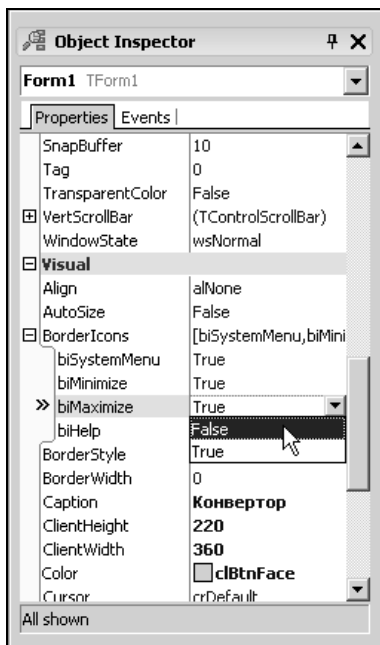


Рис. 2.4. Изменение значения уточняющего свойства

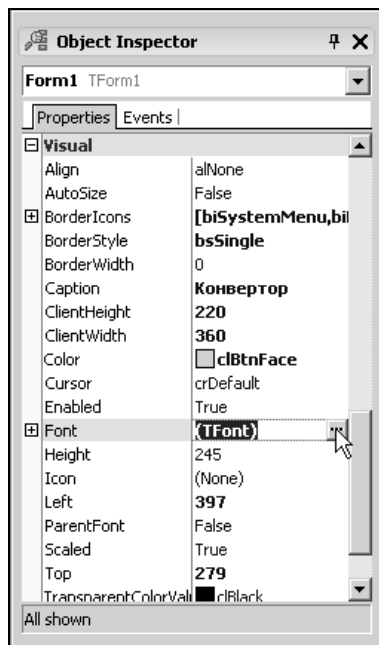


Рис. 2.5. Чтобы задать свойства шрифта, щелкните на кнопке с тремя точками

В табл. 2.2 приведены значения свойств стартовой формы программы **Конвертор** (значения других свойств оставлены без изменения и поэтому в таблице не представлены). 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	
Height	247	

Таблица 2.2 (окончание)

Свойство	Значение	Комментарий
Width	360	
BorderStyle	bsSingle	Тонкая граница. Во время работы программы пользователь не сможет изменить размер окна путем захвата и перемещения его границы
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	10	
Position	poDesktopCenter	В момент запуска программы окно появится в центре рабочего стола

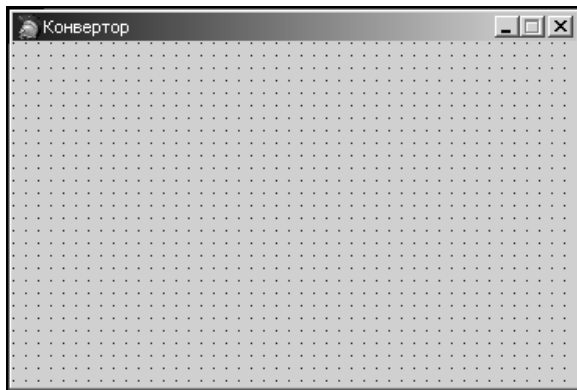


Рис. 2.6. Так должна выглядеть форма после ее настройки

Компоненты

Различают два типа компонентов: визуальные и невидимые.

Визуальный компонент — это элемент пользовательского интерфейса, например, поле редактирования (Edit), поле отображения текста (Label), кнопка (Button), список (ListBox), поле отображения иллюстрации (Image).

Визуальные компоненты отображаются как на форме (во время разработки программы), так и в окне программы во время ее работы. Визуальные компоненты образуют VCL-библиотеку (Visual Component Library), т. е. библиотеку визуальных компонентов (поэтому приложения, создаваемые в Turbo C++, называются VCL-приложениями).

Невизуальные компоненты отображаются только на форме во время разработки программы. Таймер (Timer) — типичный невизуальный 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.

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.

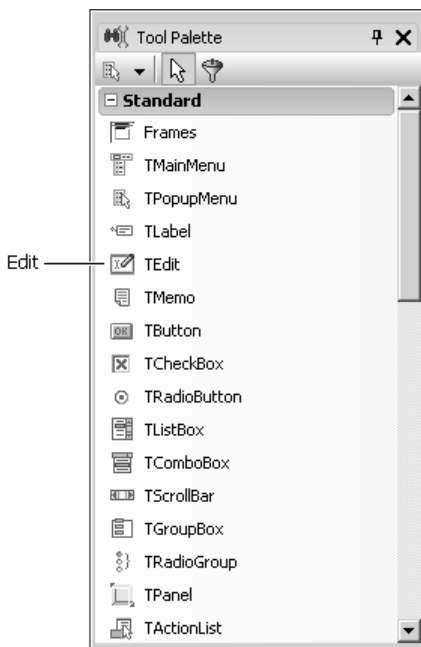


Рис. 2.7. Компонент Edit — поле редактирования

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

1. В палитре компонентов (окно **Tool Palette**) раскрыть вкладку **Standard**.
2. Сделать щелчок 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 действий на форме должен появиться компонент `Edit` — поле редактирования (рис. 2.8).

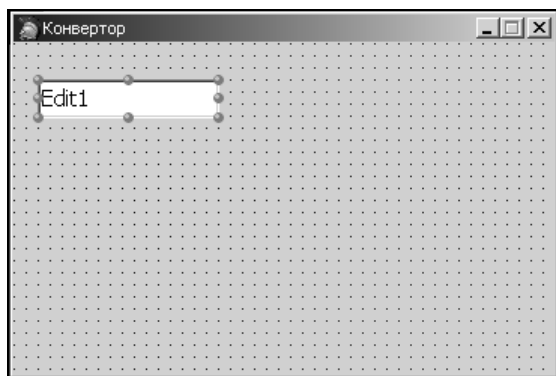


Рис. 2.8. Результат добавления на форму компонента `Edit`

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`. 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` (объект типа `TEdit`)

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

Таблица 2.3 (окончание)

Свойство	Описание
Height	Высота компонента
Width	Ширина компонента
Font	Шрифт, используемый для отображения текста в поле компонента
ParentFont	Признак наследования шрифта от формы. Если значение свойства равно <code>true</code> , то для отображения текста в поле компонента используется шрифт формы
MaxLength	Количество символов, которое можно ввести в поле редактирования. Если значение свойства равно нулю, то ограничения на количество символов нет

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

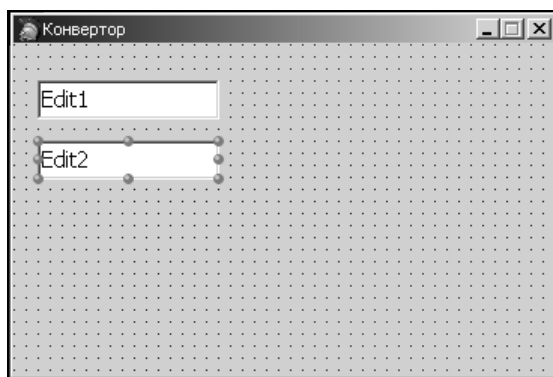


Рис. 2.9. Форма с двумя компонентами

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