

НИКИТА КУЛЬТИН

Visual Basic

ОСВОЙ САМОСТОЯТЕЛЬНО



+CD



bhv®

Никита Культин

Visual Basic

ОСВОЙ САМОСТОЯТЕЛЬНО

Санкт-Петербург
«БХВ-Петербург»
2005

УДК 681.3.068+800.92VB
ББК 32.973.26-018.1
K90

Культин Н. Б.

K90 Visual Basic. Освой самостоятельно. — СПб.:
БХВ-Петербург, 2005. — 480 с.: ил.

ISBN 5-94157-516-5

Рассмотрен язык программирования Visual Basic. Изложены принципы визуального проектирования и событийного программирования. На конкретных примерах показаны возможности среды разработки, методика создания программы и тонкости процесса программирования. Большое внимание уделено программированию графики, баз данных, игр. Приведено описание базовых компонентов и наиболее часто используемых функций. На прилагаемом компакт-диске находятся исходные тексты программ.

Для начинающих программистов

УДК 681.3.068+80.92VB
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Римма Смоляк</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.05.05.

Формат 60×90^{1/16}. Печать офсетная. Усл. печ. л. 30.

Тираж 4000 экз. Заказ №

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

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

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

ISBN 5-94157-516-5

© Культин Н. Б., 2005
© Оформление, издательство "БХВ-Петербург", 2005

Оглавление

Предисловие	1
Visual Basic — что это?.....	1
Об этой книге	2
 ЧАСТЬ I. СРЕДА РАЗРАБОТКИ MICROSOFT VISUAL BASIC	 5
 Глава 1. Установка	 7
Начало работы	9
 Глава 2. Первый проект	 15
Форма	16
Компоненты.....	21
Событие и процедура обработки события.....	31
Редактор кода	37
Запись инструкций.....	39
Справочная информация.....	42
Сохранение проекта.....	43
Запуск программы.....	45
Создание exe-файла	47
Завершение работы	48
Внесение изменений.....	49
Значок приложения	54
Окончательная настройка приложения	57
Перенос приложения на другой компьютер	60

Глава 3. Базовые компоненты	61
Компонент <i>Label</i>	61
Компонент <i>TextBox</i>	65
Простой редактор текста	71
Компонент <i>CommandButton</i>	76
Компонент <i>CheckBox</i>	80
Компонент <i>OptionButton</i>	84
Компонент <i>ComboBox</i>	87
Компонент <i>Timer</i>	93
Компонент <i>PictureBox</i>	100
Компонент <i>Image</i>	107
 Глава 4. Дополнительные компоненты.....	 112
Компонент <i>CommonDialog</i>	114
Компонент <i>StatusBar</i>	119
Компонент <i>ProgressBar</i>	122
Компонент <i>UpDown</i>	126
Компонент <i>Menu</i>	132
 ЧАСТЬ II. ПРАКТИКУМ ПРОГРАММИРОВАНИЯ	 141
 Глава 5. Графика.....	 143
Графическая поверхность.....	143
Графические примитивы	145
Точка	146
Линия	147
Прямоугольник	150
Окружность и круг	155
Дуга и сектор.....	157
Эллипс	160
Текст	163
Иллюстрации	167
Битовые образы	179
Мультипликация	184
Использование битовых образов	193
Загрузка битового образа из ресурса программы	201
Создание файла ресурсов	202
Доступ к файлу ресурсов	204
Загрузка ресурса.....	205

Глава 6. Мультимедиа	211
Функция <i>PlaySound</i>	211
Компонент <i>MMControl</i>	215
Воспроизведение звука	219
MIDI	226
CD-плеер	231
Регулятор громкости	237
Просмотр видеороликов	246
Установка программы на компьютере пользователя	251
Глава 7. Базы данных	253
База данных и СУБД	253
Локальные и удаленные базы данных	254
Структура базы данных	254
Механизмы доступа к данным	255
Компоненты доступа и отображения данных	255
Технология ODBC	257
Строка соединения	258
Программа работы с базой данных	259
Создание базы данных	259
Регистрация базы данных	259
Работа с базой данных в режиме таблицы	261
Выбор информации из базы данных	269
Работа с базой данных в режиме формы	275
Еще раз о создании базы данных	288
Создание файла базы данных	289
Создание таблицы	290
Добавление информации	291
Удаление таблицы	291
Создание базы данных	292
Установка программы работы с базой данных на компьютере пользователя	295
Глава 8. Справочная система	296
Справочная система HTML Help 1.x	299
Подготовка справочной информации	299
Создание справочной системы	306
Отображение справочной информации	315

Глава 9. Программа установки	322
Утилита IExpress	323
Создание программы установки	325
Установка программы	332
Глава 10. Примеры программ	335
Система проверки знаний	335
Требования к программе	335
Файл теста	336
Форма приложения	340
Отображение иллюстраций	341
Доступ к файлу теста	341
Текст программы	343
Запуск программы тестирования	355
Игра "Сапер"	356
Правила игры и представление данных	357
Форма приложения	360
Начало работы программы	361
Новая игра	363
Игра	368
Справочная информация	370
Информация о программе	371
Текст программы	376
ЧАСТЬ III. СПРАВОЧНИК	387
Глава 11. Краткий справочник	389
Компоненты	389
Форма	389
<i>CheckBox</i>	392
<i>ComboBox</i>	393
<i>CommandButton</i>	395
<i>CommonDialog</i>	397
<i>DirListBox</i>	398
<i>DriveListBox</i>	400
<i>FileListBox</i>	401
<i>Image</i>	403
<i>Label</i>	404

<i>Line</i>	406
<i>ListBox</i>	407
<i>MMControl</i>	409
<i>OptionButton</i>	411
<i>PictureBox</i>	412
<i>ProgressBar</i>	415
<i>Shape</i>	416
<i>StatusBar</i>	418
<i>TextBox</i>	420
<i>Timer</i>	422
<i>UpDown</i>	423
Графика	425
<i>Circle</i>	425
<i>Line</i>	427
<i>LoadPicture</i>	428
<i>LoadResPicture</i>	428
<i>PaintPicture</i>	428
<i>Print</i>	429
<i>PSet</i>	429
<i>RGB</i>	430
Функции	434
Ввод и вывод	434
Математические функции	435
Преобразование данных	437
Работа со строками	438
Работа с датами и временем	442
Работа с файлами	444
Приложение. Описание прилагаемого компакт-диска	450
Предметный указатель	459

Предисловие

Visual Basic — что это?

В последнее время возрос интерес к программированию, что связано с развитием и внедрением в повседневную жизнь информационно-коммуникационных технологий. Если человек имеет дело с компьютером, то рано или поздно у него возникает желание, а иногда и необходимость программировать.

Среди пользователей персональных компьютеров в настоящее время наиболее популярно семейство операционных систем Windows и, естественно, что тот, кто собирается программировать, стремится писать программы, которые будут работать в этих системах.

Раньше рядовому программисту оставалось только мечтать о создании собственных программ, работающих в среде Windows, т. к. средства разработки были явно ориентированы на профессионалов, обладающих серьезными знаниями и опытом.

Бурное развитие вычислительной техники, потребность в эффективных средствах разработки программного обеспечения привели к появлению систем программирования, ориентированных на так называемую "быструю разработку", пионером среди которых был пакет Microsoft Visual Basic.

В основе систем быстрой разработки (RAD-систем, Rapid Application Development — среда быстрой разработки приложений) лежит технология визуального проектирования и событийного программирования. Суть этой технологии заключается в том, что среда разработки берет на себя большую часть рутинной работы, оставляя программисту работу по конструированию диалоговых окон и функций обработки событий. Производительность программиста при использовании RAD-систем — фантастическая!

Microsoft Visual Basic — это среда быстрой разработки, в которой в качестве языка программирования используется Visual Basic. В основе идеологии Visual Basic лежит технология визуального проектирования и методология объектно-ориентированного событийного программирования.

В настоящее время наиболее широко используется шестая версия пакета — Microsoft Visual Basic 6.0, которая позволяет создавать самые различные программы: от простейших однооконных приложений до программ управления распределенными базами данных.

Microsoft Visual Basic может работать в среде операционных систем от Windows 98 до Windows XP. Особых требований, по современным меркам, к ресурсам компьютера пакет не предъявляет.

Об этой книге

В книге, которая посвящена программированию в конкретной среде, необходим баланс между тремя линиями:

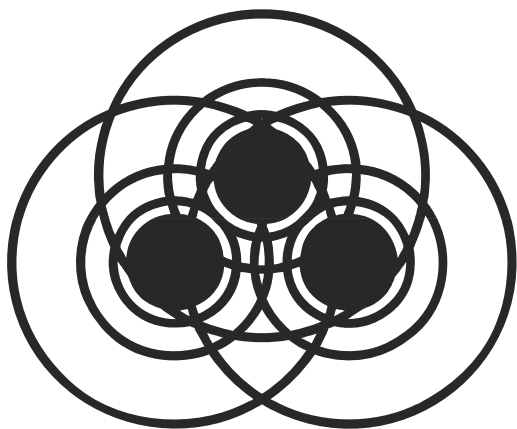
- язык программирования;
- техника и технология программирования (программирование как таковое);
- среда разработки.

Чтобы описать процесс разработки программы, объяснить, как она работает, нужно оперировать такими терминами, как *объект*, *событие*, *свойство*, понимание которых на начальном этапе изучения программирования весьма проблематично. Как поступить? Сначала дать описание языка, а затем приступить к описанию среды разработки и процесса программирования в Visual Basic? Очевидно, что это не лучший вариант. Поэтому при изложении материала принят подход (метод), который можно назвать "от задачи". Суть метода заключается в том, что берется конкретная задача и на ее примере рассматриваются определенная технология, возможности среды разработки и особенности языка программирования, необходимые для решения этой конкретной задачи.

Книга, которую вы держите в руках, — это не описание языка программирования и среды разработки Visual Basic. Это руководство (учебное пособие) по программированию в Microsoft Visual Basic. В нем рассмотрена вся цепочка, весь процесс создания программы: от разработки диалогового окна и процедур обработки событий до создания справочной системы.

Цель этой книги — научить программировать в Microsoft Visual Basic 6.0, т. е. создавать законченные программы различного назначения: от простых однооконных приложений до вполне профессиональных программ работы с базами данных.

Научиться программировать можно только в процессе решения конкретных задач. Поэтому, чтобы получить максимальную пользу от книги, вы должны работать с ней активно. Не занимайтесь просто чтением примеров, реализуйте их с помощью вашего компьютера. Не бойтесь экспериментировать, вносите изменения в программы — чем больше сделаете самостоятельно, тем большему вы научитесь!

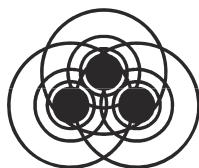


ЧАСТЬ I

**СРЕДА РАЗРАБОТКИ
MICROSOFT VISUAL BASIC**

Первая часть книги знакомит читателя со средой разработки Microsoft Visual Basic. На примере программы **Конвертор**, которая позволяет пересчитать цену из долларов в рубли, демонстрируется процесс создания программы (технология визуального проектирования и событийного программирования), вводятся основные понятия и термины объектно-ориентированного программирования. Также в первой части приведено описание базовых и дополнительных компонентов.

Глава 1



Установка

Устанавливается Microsoft Visual Basic обычным образом. Программа установки стартует автоматически.

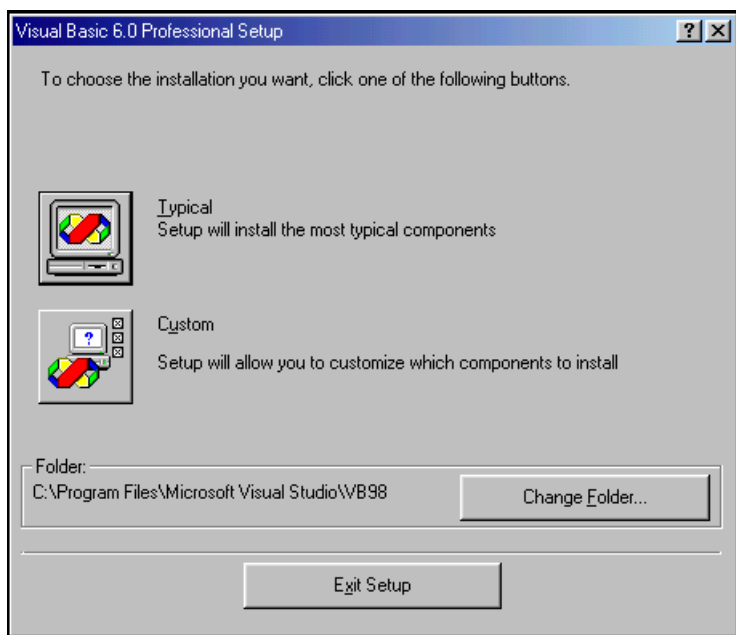


Рис. 1.1. Диалоговое окно выбора варианта установки

После того как установочный CD будет помещен в дисковод, надо ввести регистрационную информацию и выбрать каталог, куда предполагается установить Visual Basic, а затем выбрать вариант установки (рис. 1.1):

- ☐ **Typical** (Обычный). В этом случае с установочного CD на жесткий диск компьютера будут скопированы все основные компоненты Microsoft Visual Basic (лучше выбрать именно этот вариант установки);
- ☐ **Custom** (Выборочный, определяемый программистом). Этот вариант позволяет программисту самому выбрать компоненты, которые надо установить. Рекомендуется в том случае, когда на диске компьютера недостаточно свободного места для полной установки.

По окончании процесса копирования файлов программа установки предлагает перезапустить компьютер, а после перезагрузки поместить в CD-дисковод компакт-диск, на котором находится MSDN Library (Microsoft Developer Network — справочная система Microsoft Visual Studio). Возможны три варианта установки MSDN:

- ☐ **Typical** (Обычный). Предполагается, что на жесткий диск компьютера будет скопировано ядро справочной системы (порядка 60 Мбайт), а вся остальная информация будет загружаться по мере необходимости с компакт-диска;
- ☐ **Custom** (Выборочный). Этот вариант позволяет программисту выбрать разделы справочной информации, которые будут скопированы на диск компьютера, остальные разделы также будут доступны, но только тогда, когда компакт-диск с MSDN находится в дисковом диске;
- ☐ **Full** (Полный). В этом случае вся справочная информация (порядка 800 Мбайт) будет скопирована на жесткий диск.

Если программист предполагает работать только с Visual Basic, рекомендуется выбрать выборочный (**Custom**) вариант установки MSDN (рис. 1.2), а затем разделы: **VB Documentation** (Документация Visual Basic), **VB Product Samples** (Примеры Visual Basic) и **VS Shared Documentation** (Общая документация Visual Studio).

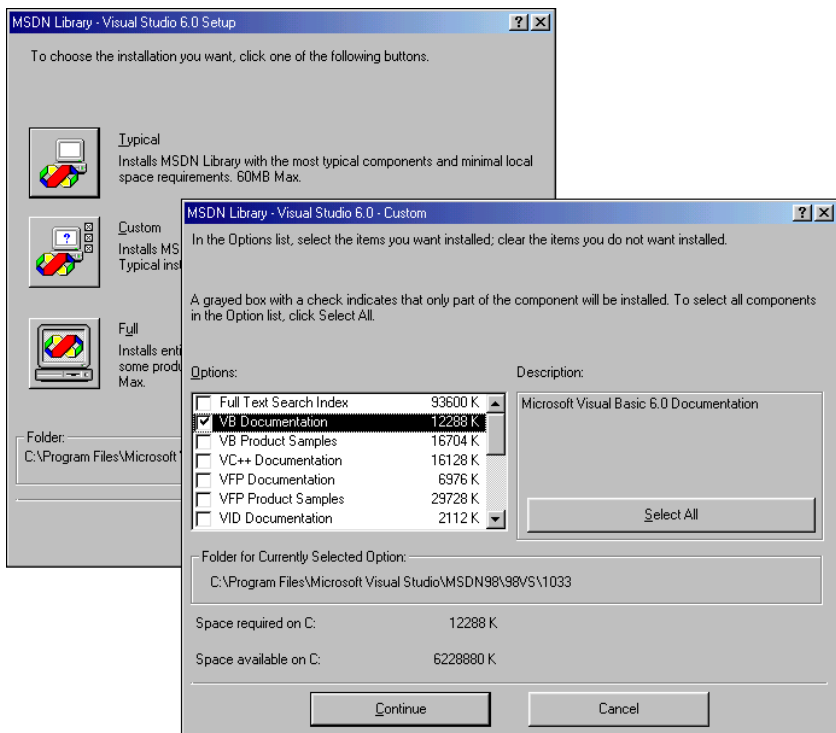


Рис. 1.2. Установка MSDN-справочной системы

Начало работы

Запускается Visual Basic обычным образом, т. е. выбором команды **Пуск** ► **Программы** ► **Microsoft Visual Basic 6.0** ► **Microsoft Visual Basic 6.0** (рис. 1.3).

Если Visual Basic запущен сразу после установки, то на фоне главного окна отображается окно **New Project** (рис. 1.4). В этом окне на вкладке **New** перечислены виды программ (проектов), которые можно создать в Visual Basic. Чтобы приступить к работе над новым *Windows-приложением*, надо выбрать **Standard EXE** и щелкнуть на кнопке **Открыть**. Если после запуска Visual Basic окна **New Project** на экране не отображается, то для того чтобы оно стало доступным, надо в меню **File** выбрать команду **New Project**.

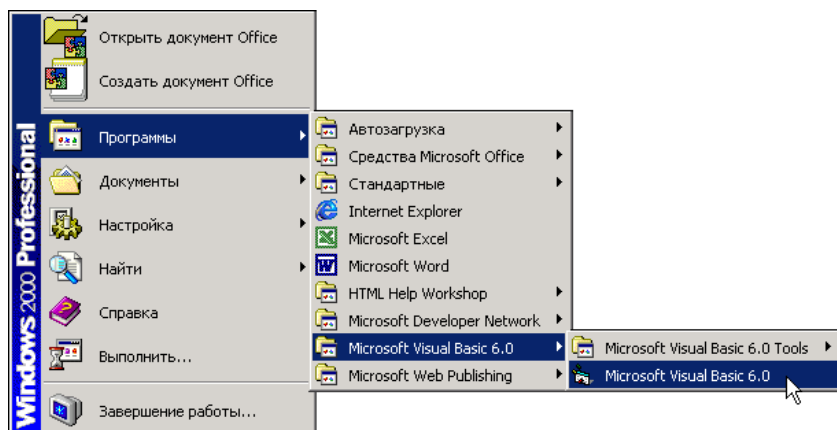


Рис. 1.3. Запуск Visual Basic



Рис. 1.4. Начало работы над новой программой (Windows-приложением)

Вид окна Visual Basic в начале работы над стандартным Windows-приложением приведен на рис. 1.5. В верхней части окна находится строка меню и стандартная панель инструментов, слева — *палитра компонентов*, в центре — *окно конструктора формы*, справа — *окно менеджера проекта*, *окно свойств* и *окно отображения положения формы*.

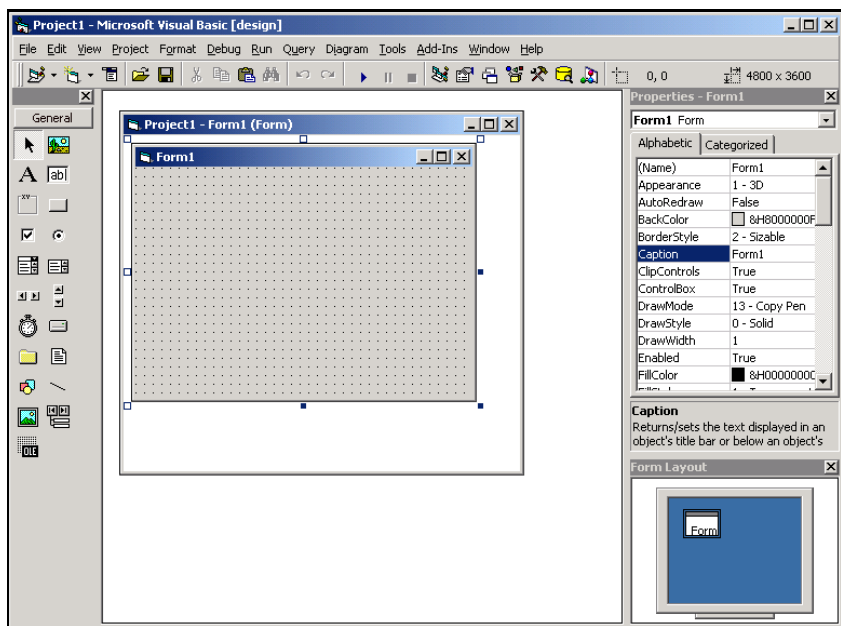


Рис. 1.5. Окно Visual Basic
в начале работы над новой программой

На стандартной панели инструментов (рис. 1.6) находятся кнопки, соответствующие наиболее часто используемым командам. Здесь же располагаются кнопки, с помощью которых можно быстро сделать доступными окна палитры компонентов, менеджера проектов, свойств и др.

В окне конструктора формы (рис. 1.7) находится *форма*, которая представляет собой заготовку окна разрабатываемого приложения.



Рис. 1.6. Стандартная панель инструментов

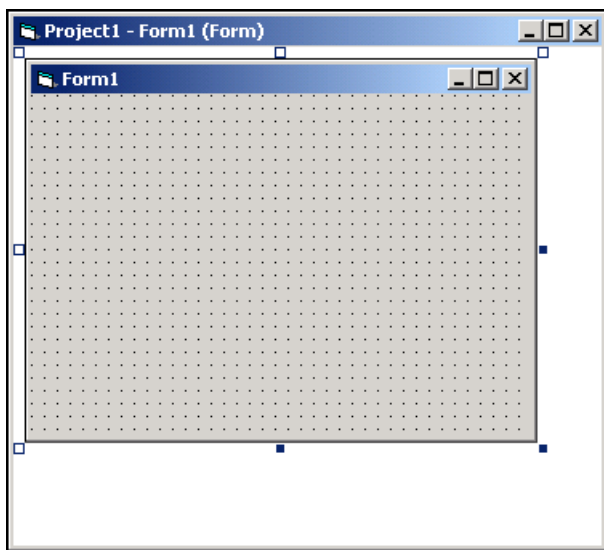


Рис. 1.7. Окно конструктора формы

В палитре компонентов (рис. 1.8) находятся значки, обозначающие *компоненты* — элементы интерфейса пользователя, которые программист может поместить на поверхность формы. Если палитра компонентов не отображается, то для того чтобы она стала доступной, надо выбрать команду **View ► Toolbox** или щелкнуть на соответствующей кнопке панели инструментов.

В окне менеджера проекта (рис. 1.9) отображается структура (состав) проекта, над которым в данный момент идет работа. Если окно менеджера проекта не отображается, то для того чтобы оно стало доступным, надо выбрать команду **View ► Project Explorer** или щелкнуть на соответствующей кнопке панели инструментов.



Рис. 1.8. Палитра компонентов
(окно **Toolbox**)

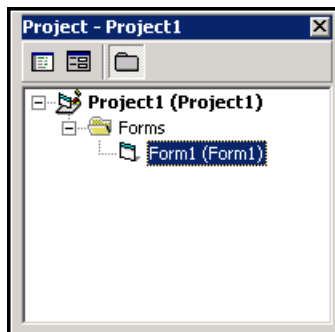


Рис. 1.9. В окне **Project**
отображается структура
(состав) проекта

Окно свойств (рис. 1.10) предназначено для отображения и редактирования значений свойств объектов. В терминологии визуального проектирования *объект* — это диалоговое окно или элемент интерфейса пользователя (поле ввода, командная кнопка, переключатель и др.).

Свойство — это характеристика, которая определяет (задает) внешний вид объекта. Например, значение свойства *Caption* задает заголовок формы, а свойств *Width* и *Height* — ее размер. Свойства на вкладке **Alphabetic** отображаются в алфавитном порядке, а на вкладке **Categorized** сгруппированы. Например, в группу **Appearance** объединены свойства, которые определяют вид формы (заголовок, фоновый рисунок, тип границы и т. д.), а в группу **Position** — свойства, определяющие размер и положение на экране.

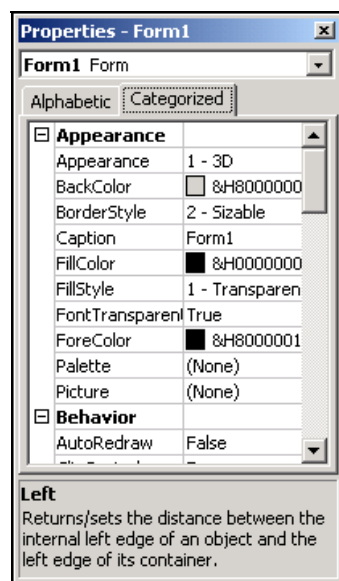
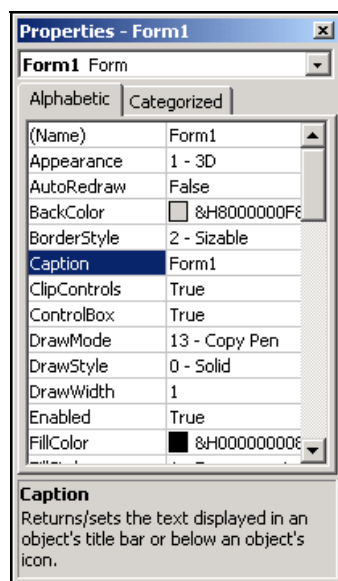
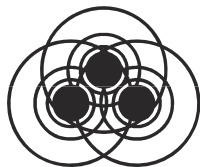


Рис. 1.10. В окне **Properties** перечислены свойства объекта и указаны их значения

Глава 2



Первый проект

Процесс создания программы (Windows-приложения) в Visual Basic рассмотрим на примере. Пусть надо создать программу, которая пересчитывает цену из долларов в рубли. Вид окна программы во время ее работы после ввода исходных данных и щелчка на кнопке **Пересчет** приведен на рис. 2.1.

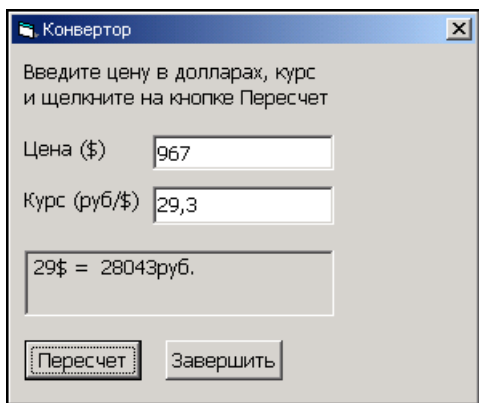


Рис. 2.1. Окно программы пересчета цены из долларов в рубли

Чтобы начать работу над новой программой или, как принято говорить, над новым проектом, запустите Visual Basic, на вкладке **New** окна **New Project** выберите **Standard EXE** и щелкните на кнопке **Открыть**. Если после запуска Visual Basic окно **New Project** не отображается, то выберите команду **File ► New Project**.

Форма

Работа над новым *проектом* (так принято называть программу (приложение), над которой идет работа) начинается с создания или, как иногда говорят, с настройки стартовой формы.

Стартовая форма создается путем изменения значений свойств, находящихся в окне дизайнера формы **Form1**, и добавления к этой форме необходимых компонентов (полей ввода, отображения текста, командных кнопок и т. д.).

Для просмотра и изменения значений свойств формы и ее компонентов используется окно **Properties**. В верхней части этого окна указано имя объекта, значения свойств которого отображаются в данный момент. В левой колонке окна перечислены свойства объекта, в правой — указаны значения свойств.

Некоторые свойства формы приведены в табл. 2.1.

Таблица 2.1. Свойства формы (объекта **Form**)

Свойство	Описание
Name	Имя формы. Используется для доступа к форме и ее компонентам
Caption	Текст заголовка
Width	Ширина формы. Задается в твипах
Height	Высота формы. Задается в твипах
StartPosition	Положение формы при ее появлении на экране. Форма может располагаться в центре экрана (Center Screen), в центре родительской формы (Center Owner). Положение формы могут определять значения свойств Top и Left (в этом случае значение свойства StartUpPosition должно быть равно Manual)
Top	Расстояние от верхней границы формы до верхней границы экрана или до верхней границы родительской формы

Таблица 2.1 (продолжение)

Свойство	Описание
Left	Расстояние от левой границы формы до левой границы экрана или до левой границы родительской формы
Icon	Значок (картинка) в заголовке, обозначающая системное меню
MaxButton	Признак наличия в заголовке окна кнопки Развернуть
MinButton	Признак наличия в заголовке окна кнопки Свернуть
BorderStyle	<p>Стиль (вид) границы.</p> <p>Граница может быть:</p> <ul style="list-style-type: none"> • обычной (Sizable); • тонкой (Fixed Single) (в этом случае изменить размер окна путем перемещения границы мышью нельзя); • вообще отсутствовать (None). <p>Если значение свойства равно Fixed Dialog, то граница окна тонкая и кнопки Развернуть и Свернуть в заголовке не отображаются</p>
BackColor	<p>Цвет формы.</p> <p>Цвет можно задать, выбрав его из палитры или указав привязку к элементу цветовой схемы операционной системы. Во втором случае цвет определяется текущей цветовой схемой и выбранным компонентом привязки, и меняется при изменении цветовой схемы операционной системы</p>
ScaleMode	Определяет (задает) единицу измерения размеров компонентов, которые находятся на поверхности формы. Размер компонентов может измеряться в твипах (Twip), пикселах (Pixel) и других единицах

Таблица 2.1 (окончание)

Свойство	Описание
Font	Шрифт, который по умолчанию используется находящимися на поверхности формы компонентами для отображения текста (например, надпись на командной кнопке, текст в поле редактирования или в поле отображения текста)

Сначала надо задать заголовок формы — изменить значение свойства `Caption` с `Form1` на `Конвертор`. Чтобы это сделать, нужно в окне **Properties** выбрать свойство `Caption` и щелкнуть мышью в поле значения свойства. В результате этих действий в поле значения свойства (после слова `Form1`) появится курсор и можно будет ввести значение свойства (рис. 2.2).

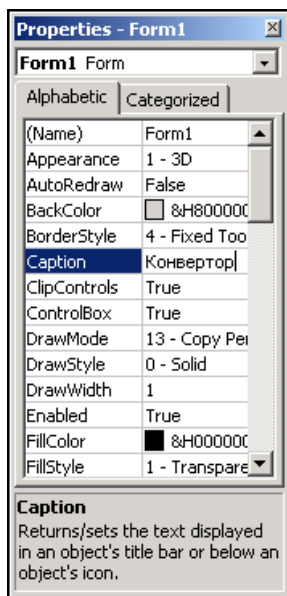


Рис. 2.2. Изменение значения свойства путем ввода строки

Следует обратить внимание, что ширина и высота формы измеряются в специальных единицах — твипах. Задавать значения свойств `Width` и `Height` в твипах неудобно. Гораздо проще захватить один из находящихся на границе формы черных квадратиков и переместить границу (вертикальную, горизонтальную или обе сразу) в нужном направлении (рис. 2.3). По окончании перемещения границы значения свойств `Width` и `Height` автоматически изменятся и будут соответствовать установленному размеру формы.

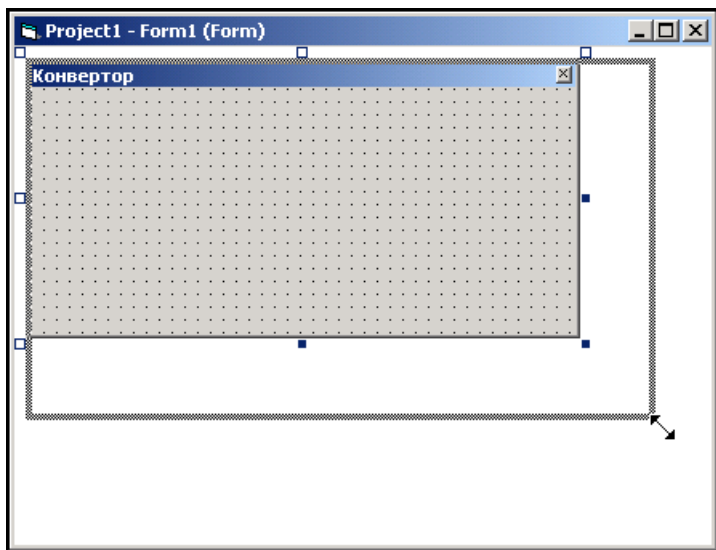


Рис. 2.3. Изменение размера формы путем перемещения границы

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. 2.4). Здесь следует обратить внимание на то, что в списке сначала указывается числовое значение константы, а затем — ее название. При этом не следует путать название константы и ее обозначение. Например, чис-

ленное значение константы `Fixed Single` равно единице, а символьное значение равно `vbFixedSingle`.

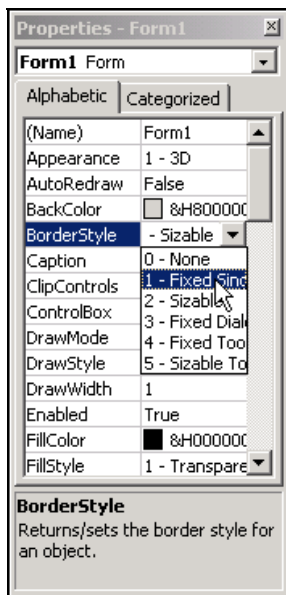


Рис. 2.4. Изменение значения свойства путем выбора из списка

Рядом со значениями некоторых свойств отображается командная кнопка с тремя точками. Это значит, что для изменения значения свойства используется дополнительное диалоговое окно. Например, в результате щелчка на кнопке с тремя точками в строке свойства `Icon` открывается окно **Load Icon**, в котором можно открыть один из каталогов компьютера и выбрать иконку (картинку, которая будет изображать системное меню в заголовке формы).

В табл. 2.2 приведены значения свойств стартовой формы разрабатываемой программы. Остальные свойства оставлены без изменения и в таблице не приведены.

Таблица 2.2. Значения свойств стартовой формы

Свойство	Значение
Caption	Конвертор
Width	4425
Height	3705
BorderStyle	Fixed Single
MaxButton	False
MinButton	False
StartPosition	CenterScreen
ScaleMode	Pixel
Font	Tahoma, обычный, 10

Компоненты

Программа пересчета цены из долларов в рубли должна получить от пользователя исходные данные — курс (соотношение рубля к доллару) и цену в долларах. В подобных программах данные с клавиатуры вводят в поля редактирования. Поэтому в форму разрабатываемого приложения надо добавить два поля редактирования — два компонента `TextBox`.

Для того чтобы добавить в форму компонент `TextBox` или другой *компонент*, необходимо:

1. В окне **ToolBox** выбрать этот компонент (щелкнуть на значке компонента) (рис. 2.5).
2. Установить указатель мыши в ту точку формы, в которой должен быть левый верхний угол компонента, и нажать левую кнопку мыши.
3. Переместить указатель мыши в ту точку, в которой должен быть правый нижний угол компонента, и отпустить кнопку мыши.

В результате этих действий на форме появится компонент.