

Никита Культин

# Visual Basic

ОСВОЙ САМОСТОЯТЕЛЬНО  
2-е издание

Санкт-Петербург  
«БХВ-Петербург»  
2009

УДК 681.3.06  
ББК 32.973.26-018.2  
К90

**Культин Н. Б.**

К90 Visual Basic. Освой самостоятельно. —  
2-е изд., перераб. и доп. — СПб.: БХВ-Петербург,  
2009. — 496 с.: ил. + CD-ROM

ISBN 978-5-9775-0365-5

Показан процесс создания программы в Visual Basic, в доступной форме изложены принципы визуального проектирования и событийного программирования. Примеры и задачи различной сложности — от простейших расчетных до программ работы с графикой, мультимедиа и базами данных — демонстрируют назначение базовых компонентов, раскрывают тонкости разработки в Visual Basic. Уделено внимание вопросам создания справочной системы и инсталлятора. Справочник содержит описания базовых компонентов, процедур и функций. Второе издание дополнено новыми примерами. На прилагаемом компакт-диске находятся приведенные в книге программы и их исходные тексты.

*Для начинающих программистов*

УДК 681.3.06  
ББК 32.973.26-018.2

#### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Владимир Красовский</i>
Компьютерная верстка	<i>Натали Смирновой</i>
Корректор	<i>Наталия Першакова</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 29.12.08.

Формат 60×90<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 31.

Тираж 2500 экз. Заказ №

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

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.003650.04.08 от 14.04.2008 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 978-5-9775-0365-5

© Культин Н. Б., 2009  
© Оформление, издательство "БХВ-Петербург", 2009

# Оглавление

<b>Предисловие</b> .....	<b>1</b>
Visual Basic — что это? .....	1
Об этой книге .....	2
<b>ГЛАВА 1. Visual Basic</b> .....	<b>7</b>
Установка .....	7
Первое знакомство .....	9
<b>ГЛАВА 2. Первый проект</b> .....	<b>17</b>
Начало работы .....	17
Форма.....	18
Компоненты .....	23
Событие и процедура обработки события .....	34
Редактор кода .....	40
Запись инструкций .....	43
Справочная информация.....	46
Сохранение проекта .....	46
Запуск программы .....	49
Исключения.....	52
Обработка исключений .....	53
Создание EXE-файла .....	56
Завершение работы .....	57
Внесение изменений .....	57
Значок приложения.....	62
Окончательная настройка приложения .....	66
Установка приложения на другой компьютер .....	68

<b>ГЛАВА 3. Базовые компоненты.....</b>	<b>70</b>
<i>Label</i> .....	70
<i>TextBox</i> .....	78
<i>CommandButton</i> .....	82
<i>CheckBox</i> .....	87
<i>OptionButton</i> .....	91
<i>ComboBox</i> .....	94
<i>Timer</i> .....	100
<i>PictureBox</i> .....	103
<i>Image</i> .....	112
<b>ГЛАВА 4. Дополнительные компоненты.....</b>	<b>118</b>
Компонент <i>CommonDialog</i> .....	120
Компонент <i>StatusBar</i> .....	126
Компонент <i>ProgressBar</i> .....	131
Компонент <i>UpDown</i> .....	135
Меню.....	139
<b>ГЛАВА 5. Графика.....</b>	<b>153</b>
Графическая поверхность .....	153
Графические примитивы.....	155
Точка .....	156
Линия.....	157
Прямоугольник.....	161
Окружность и круг .....	165
Дуга и сектор .....	168
Эллипс .....	173
Текст.....	176
Иллюстрации.....	182
Битовые образы.....	196
Мультипликация.....	201
Загрузка битового образа из ресурса .....	214
Создание файла ресурсов .....	214
Доступ к файлу ресурсов.....	216
Загрузка ресурса.....	217

<b>ГЛАВА 6. Мультимедиа .....</b>	<b>221</b>
Функция <i>PlaySound</i> .....	221
Компонент <i>MMControl</i> .....	225
MP3-плеер.....	229
MIDI .....	238
CD-плеер.....	245
Регулятор громкости.....	251
Регулировка громкости MIDI.....	263
Просмотр видеороликов.....	267
Установка программы на другой компьютер .....	273
<b>ГЛАВА 7. Базы данных .....</b>	<b>274</b>
База данных и СУБД .....	274
Локальные и удаленные базы данных .....	275
Структура базы данных .....	276
Механизмы доступа к данным.....	276
Компоненты доступа и отображения данных .....	277
Строка соединения .....	278
Приложение работы с базой данных .....	278
Создание базы данных .....	279
Работа с базой данных в режиме таблицы .....	279
Выбор информации из базы данных.....	288
Работа с базой данных в режиме формы.....	294
Еще раз о создании базы данных .....	308
Создание файла базы данных .....	308
Создание таблицы.....	309
Добавление информации.....	310
Удаление таблицы.....	310
Создание базы данных .....	311
Установка программы работы с базой данных на другой компьютер .....	314
<b>ГЛАВА 8. Справочная система .....</b>	<b>315</b>
Справочная система HTML Help .....	316
Подготовка справочной информации .....	317
Подготовка справочной информации в Microsoft Word .....	318

Подготовка справочной информации в HTML Help Workshop.....	319
Создание СНМ-файла.....	324
Организация разделов справочной системы.....	324
Формирование вкладки <i>Содержание</i> .....	329
Компиляция.....	333
Отображение справочной информации.....	334
<b>ГЛАВА 9. Программа установки.....</b>	<b>341</b>
Утилита IExpress.....	342
Создание программы установки .....	344
Шаг 1. Тип установки .....	344
Шаг 2. Заголовок окна.....	345
Шаг 3. Подтверждение.....	345
Шаг 4. Файл лицензионного соглашения.....	346
Шаг 5. Устанавливаемые файлы .....	347
Шаг 6. Окно программы установки .....	348
Шаг 7. Сообщение после установки .....	348
Шаг 8. Имя файла программы установки.....	349
Шаг 9. Файл проекта программы установки.....	349
Установка программы.....	351
Библиотеки времени выполнения.....	353
<b>ГЛАВА 10. Примеры программ.....</b>	<b>354</b>
Система проверки знаний.....	354
Требования к программе.....	354
Файл теста.....	356
Форма приложения .....	359
Отображение иллюстрации.....	360
Доступ к файлу теста .....	361
Текст программы .....	363
Запуск программы.....	375
Игра "Сапер" .....	377
Правила и представление данных .....	378
Форма.....	380

Начало работы программы .....	382
Новая игра.....	383
Игра .....	388
Справочная информация .....	391
Информация о программе .....	392
Текст программы .....	397
<b>ГЛАВА 11. Краткий справочник.....</b>	<b>411</b>
Форма.....	411
Компоненты .....	414
<i>CheckBox</i> .....	414
<i>ComboBox</i> .....	415
<i>CommandButton</i> .....	417
<i>CommonDialog</i> .....	419
<i>DirListBox</i> .....	420
<i>DriveListBox</i> .....	422
<i>FileListBox</i> .....	423
<i>Image</i> .....	425
<i>Label</i> .....	427
<i>Line</i> .....	428
<i>ListBox</i> .....	430
<i>MMControl</i> .....	432
<i>OptionButton</i> .....	433
<i>PictureBox</i> .....	434
<i>ProgressBar</i> .....	437
<i>Shape</i> .....	438
<i>StatusBar</i> .....	441
<i>TextBox</i> .....	443
<i>Timer</i> .....	444
<i>UpDown</i> .....	445
Графика.....	447
<i>Circle</i> .....	448
<i>Line</i> .....	449
<i>LoadPicture</i> .....	450
<i>LoadResPicture</i> .....	450
<i>PaintPicture</i> .....	451

---

<i>Print</i> .....	451
<i>PSet</i> .....	452
<i>RGB</i> .....	452
Функции.....	457
Ввод и вывод .....	457
<i>InputBox</i> .....	457
<i>MsgBox</i> .....	458
Математические функции .....	459
Преобразование данных .....	461
Работа со строками .....	461
Работа с датами и временем.....	465
Работа с файлами и каталогами.....	467
<b>Приложение. Описание компакт-диска.....</b>	<b>472</b>
<b>Предметный указатель .....</b>	<b>479</b>



# Предисловие

## Visual Basic — что это?

В последнее время возрос интерес к программированию. Это связано с развитием и внедрением в повседневную жизнь информационно-коммуникационных технологий. Если кто-то 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 популярна операционная система Windows и, естественно, что тот, кто собирается программировать, стремится писать программы, которые будут работать в ней.

Раньше рядовому программисту оставалось только мечтать о создании собственных программ, работающих в среде Windows, так как 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 созданию диалоговых окон и функций обработки событий. Производительность программиста при работе в RAD-системе — фантастическая!

Microsoft Visual Basic — это среда быстрой разработки, в которой в качестве языка программирования используется Visual Basic.

В настоящее время, несмотря на появление новых версий Visual Basic, широко используется шестая версия пакета — Microsoft Visual Basic 6.0, которая стала "классикой".

В Visual Basic 6.0 можно создавать программы различного назначения: от простейших однооконных приложений до программ, работающих с графикой, мультимедиа и базами данных.

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

## Об этой книге

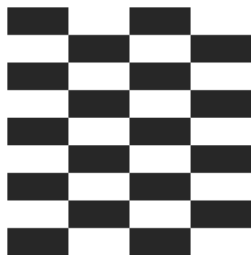
В книге, которая посвящена программированию в конкретной среде разработки, необходим баланс между тремя линиями: язык программирования, технология программирования (программирование как таковое) и среда разработки. Уже при первом знакомстве со средой разработки, представлении ее возможностей у автора возникает проблема: чтобы описать процесс разработки программы, объяснить, как работает программа, 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? Очевидно, что это не лучший вариант решения. Поэтому при изложении материала принят подход, который можно назвать "от задачи". Суть его 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 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, то есть создавать законченные программы различного назначения: от простых однооконных приложений до программ работы с базами данных.

Научиться программировать можно, только решая конкретные задачи. 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 вы сделаете самостоятельно, тем большему вы научитесь!

# ГЛАВА 1



## Visual Basic

### Установка

Устанавливается Microsoft Visual Basic обычным образом. Установщик, который стартует автоматически, после того как установочный CD будет помещен в дисковод, после ввода регистрационной информации предлагает выбрать вариант установки (рис. 1.1):

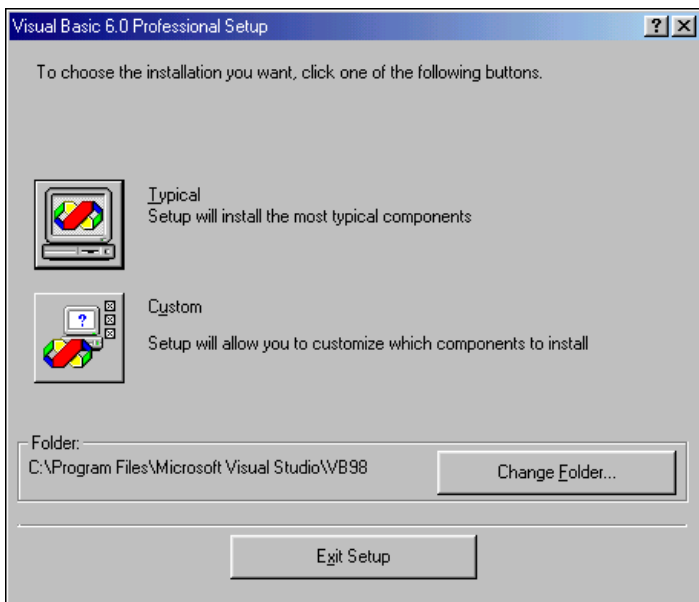
- ❑ **Typical** (Типовой). С установочного CD на жесткий диск компьютера будут скопированы основные компоненты Microsoft Visual Basic. Если на диске компьютера достаточно свободного места, то лучше выбрать именно этот вариант установки;
- ❑ **Custom** (Выборочный). Позволяет программисту указать компоненты, которые надо установить. Этот вариант можно выбрать, если на диске компьютера недостаточно свободного места для полной установки.

По окончании процесса копирования файлов установщик предлагает перезапустить компьютер и, 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** (Типовой). На жесткий диск компьютера будет скопировано ядро справочной системы (порядка 60 Мбайт), а вся

остальная информация будет загружаться с компакт-диска по мере необходимости;

- Custom** (Выборочный). 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 Library находится в дисковом де;
- Full** (Полный). Вся справочная информация (порядка 800 Мбайт) будет скопирована на жесткий диск.



**Рис. 1.1.** При установке Microsoft Visual Basic рекомендуется выбрать вариант **Typical**

Если программист предполагает работать только с Visual Basic, то рекомендуется выбрать выборочный (**Custom**) вариант установки MSDN Library (рис. 1.2) и затем — разделы: **VB Documen-**

**tion** (Документация Visual Basic), **VB Product Samples** (Примеры Visual Basic) и **VS Shared Documentation** (Общая документация Visual Studio).

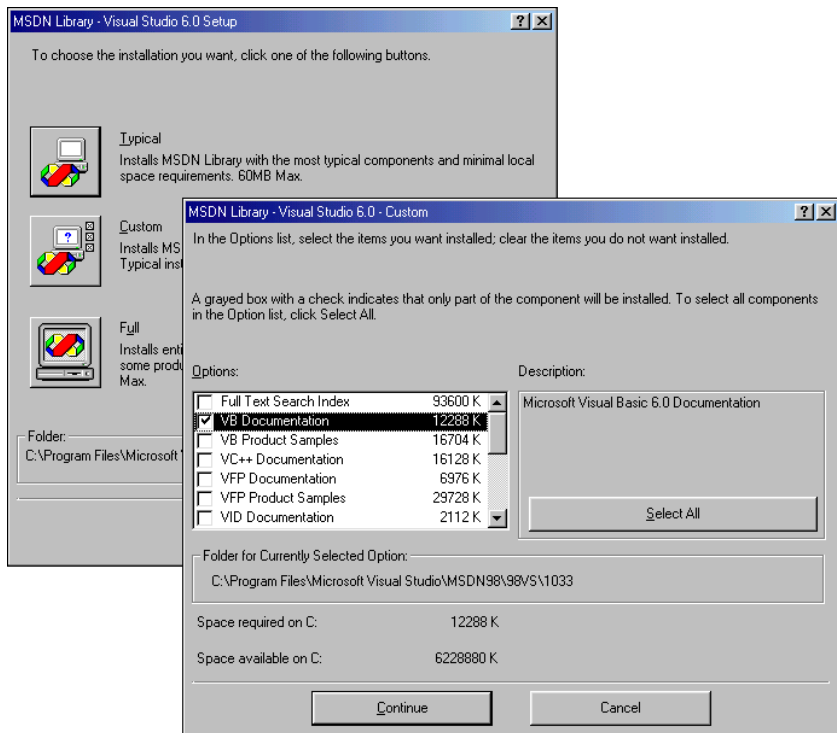


Рис. 1.2. Установка справочной системы MSDN Library

## Первое знакомство

Запускается Visual Basic обычным образом, то есть выбором в меню Пуск команды Программы ► Microsoft Visual Basic 6.0 ► Microsoft Visual Basic 6.0 (рис. 1.3).

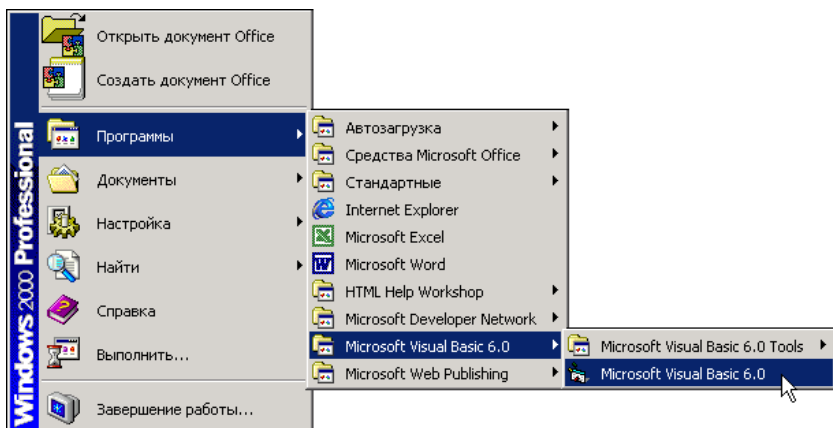


Рис. 1.3. Запуск Visual Basic

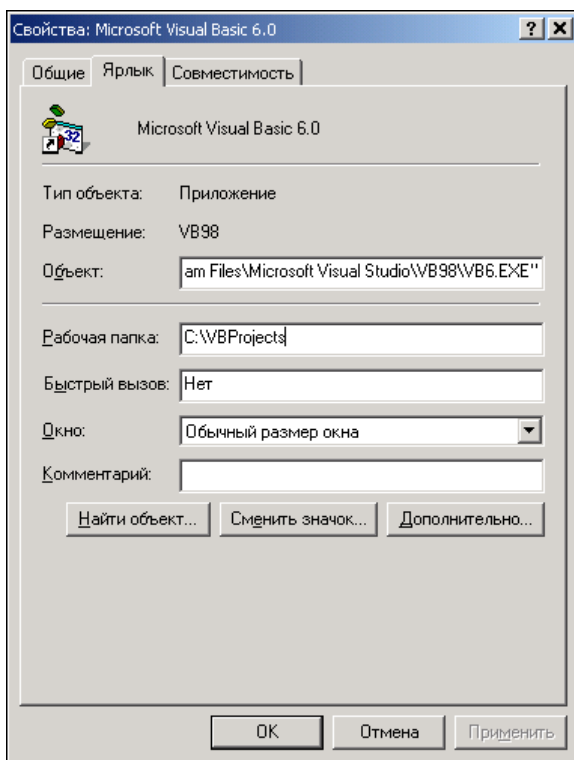
Перед тем как запустить Visual Basic первый раз, рекомендуется создать папку для проектов (программ) Visual Basic и указать ее в качестве рабочей. Папка (назвать ее можно, например, VBProjects) создается обычным образом. Чтобы указать, что папка является рабочей, надо раскрыть окно свойств команды запуска Visual Basic (сделать щелчок *правой* кнопкой мыши на команде запуска Visual Basic) и ввести ее имя в поле **Рабочая папка** (рис. 1.4).

Если Visual Basic запущен сразу после установки, то на фоне главного окна отображается окно **New Project** (рис. 1.5). В этом окне на вкладке **New** перечислены типы проектов (программ), которые можно создать в Visual Basic.

Чтобы приступить к работе над новой программой, или, как принято говорить, *приложением*, надо выбрать **Standard EXE** и сделать щелчок на кнопке **Открыть**.

### ЗАМЕЧАНИЕ

Если после запуска Visual Basic окно **New Project** на экране не отображается, то, для того чтобы начать работу над новой программой, надо в меню **File** выбрать команду **New Project**.



**Рис. 1.4.** Имя папки, предназначенной для проектов Visual Basic, надо ввести в поле **Рабочая папка**

Окно Visual Basic в начале работы над новой программой приведено на рис. 1.6. В верхней части окна находится строка меню и панель инструментов, слева — *палитра компонентов*, в центре — *окно конструктора формы*, справа — *окно проекта*, *окно свойств* и *окно отображения положения формы*.

На стандартной панели инструментов (рис. 1.7) находятся кнопки активизации наиболее часто используемых команд. Там же находятся кнопки, используя которые можно быстро сделать доступным окно палитры компонентов, менеджера проектов, свойств и др.



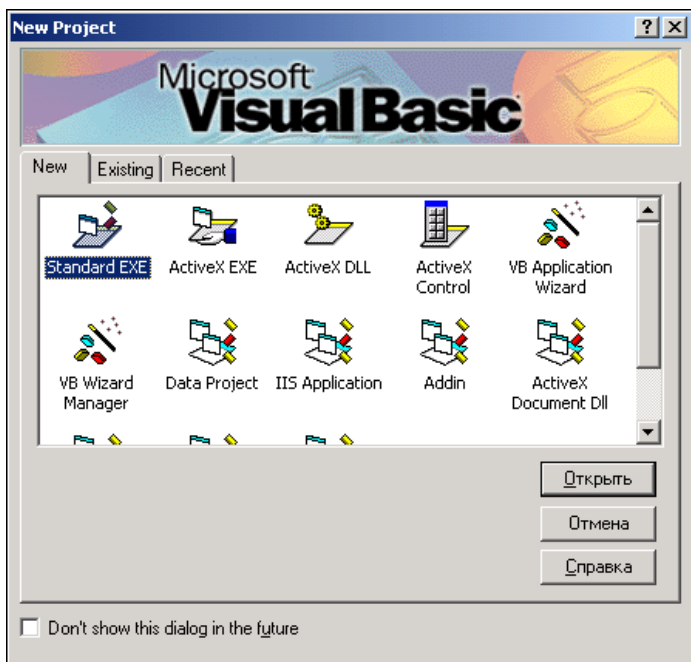


Рис. 1.5. Начало работы над новой программой (приложением)

В окне конструктора формы (рис. 1.8) находится *форма* — заготовка окна разрабатываемого приложения.

### **ЗАМЕЧАНИЕ**

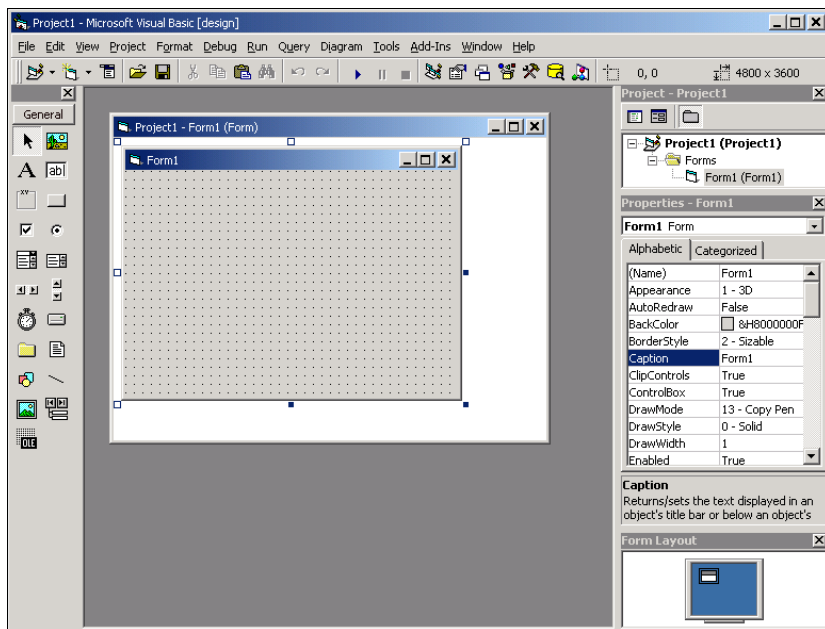
Окно программы во время ее разработки принято называть формой.

В палитре компонентов (рис. 1.9) отображаются значки компонентов, которые программист может поместить на форму.

### **ЗАМЕЧАНИЕ**

Если палитра компонентов не отображается, то, для того чтобы она стала доступной, надо в меню **View** выбрать команду

**Toolbox** или сделать щелчок на соответствующей кнопке панели инструментов.



**Рис. 1.6.** Окно Visual Basic  
в начале работы над новой программой



**Рис. 1.7.** Стандартная панель инструментов

В окне проекта (рис. 1.10) отображается структура (состав) проекта, над которым в данный момент идет работа.

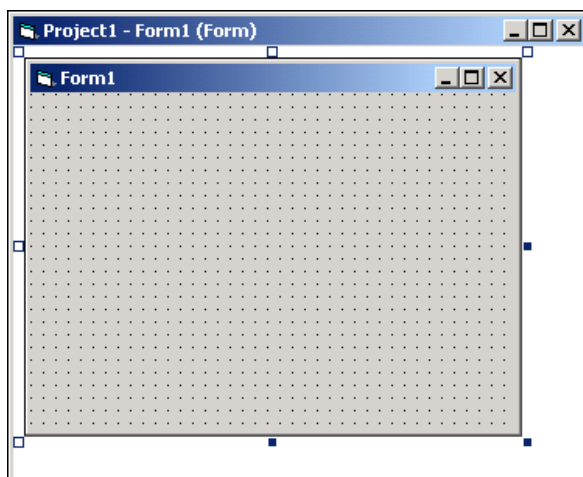


Рис. 1.8. Окно конструктора формы

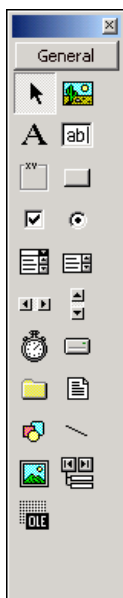
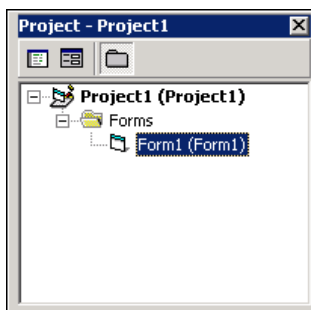


Рис. 1.9. Палитра компонентов (окно **Toolbox**)

### ЗАМЕЧАНИЕ

Если окно проекта не отображается, то, для того чтобы оно стало доступным, надо в меню **View** выбрать команду **Project Explorer** или сделать щелчок на соответствующей кнопке панели инструментов.



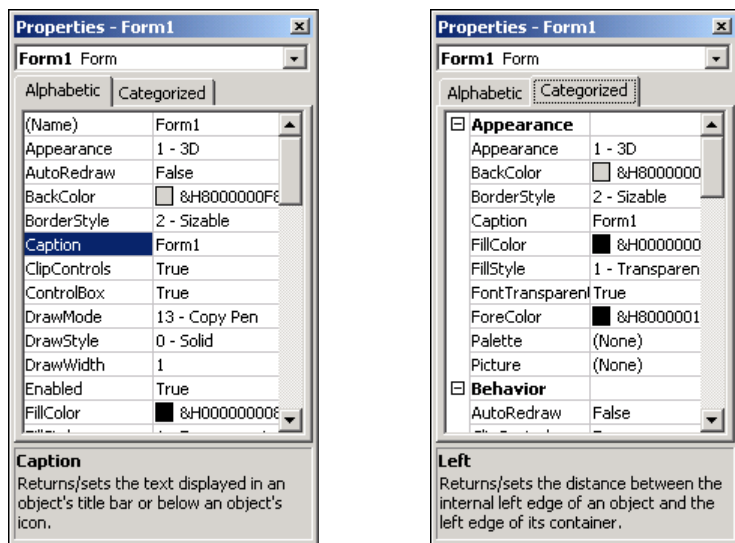
**Рис. 1.10.** В окне **Project** отображается структура (состав) проекта

Окно **Properties** (рис. 1.11) предназначено для редактирования значений свойств объектов. В нем отображаются свойства выбранного в данный момент объекта (в начале работы над новой программой — формы). На вкладке **Alphabetic** свойства отображаются в алфавитном порядке, а на вкладке **Categorized** свойства сгруппированы по функциональному признаку. Например, в группу **Appearance** объединены свойства, которые определяют вид объекта, а в группу **Position** — его размер и положение на экране (для формы) или поверхности формы (другие компоненты).

### ЗАМЕЧАНИЕ

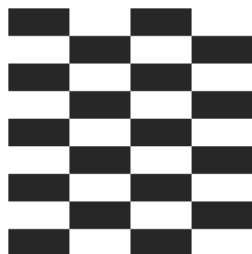
В терминологии визуального проектирования *объект* — это диалоговое окно или элемент интерфейса пользователя (поле ввода, командная кнопка, переключатель и др.). *Свойство* — это характеристика, которая определяет (задает) внешний вид

объекта. Например, значение свойства `Caption` задает заголовок формы, а свойств `Width` и `Height` — ее размер.



**Рис. 1.11.** В окне **Properties** перечислены свойства объекта и указаны их значения

## ГЛАВА 2



# Первый проект

Процесс создания программы в Visual Basic рассмотрим на примере. Создадим приложение, которое позволяет пересчитать цену из долларов в рубли, — Конвертор (рис. 2.1).

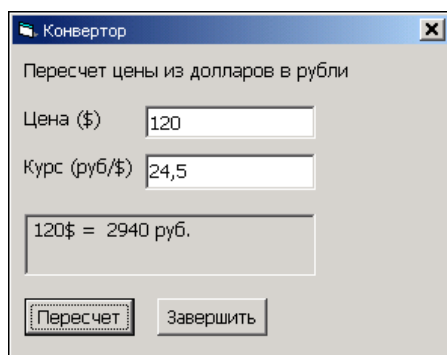


Рис. 2.1. Окно программы пересчета цены из долларов в рубли

## Начало работы

Чтобы начать работу над новой программой, или, как принято говорить, проектом, надо сначала в меню **File** выбрать команду **New Project**, затем в открывшемся окне **New Project** указать тип

создаваемого приложения (**Standard EXE**) и сделать щелчок на кнопке **ОК**.

## Форма

Работа над новой программой начинается с создания стартовой (главной) формы.

Стартовая форма создается путем изменения значений ее свойств (табл. 2.1) и добавления к форме необходимых компонентов (полей ввода, отображения текста, командных кнопок и т. д.).

*Таблица 2.1. Свойства формы*

<b>Свойство</b>	<b>Описание</b>
Name	Имя формы. Используется для доступа к форме и ее компонентам
Caption	Текст заголовка
Width	Ширина формы
Height	Высота формы
StartPosition	Положение формы при первом ее появлении на экране. Форма может располагаться в центре экрана (CenterScreen), в центре родительской формы (CenterOwner). Положение формы могут определять также 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 (окончание)

Свойство	Описание
Left	Расстояние от левой границы формы до левой границы экрана или до левой границы родительской формы
Icon	Значок (картинка) в заголовке
MaxButton	Признак наличия в заголовке окна кнопки <b>Развернуть</b>
MinButton	Признак наличия в заголовке окна кнопки <b>Свернуть</b>
BorderStyle	Стиль (вид) границы. Граница может быть обычной ( <i>Sizable</i> ), тонкой ( <i>FixedSingle</i> ) (в этом случае изменить размер окна путем перемещения границы мышью нельзя) или вообще отсутствовать ( <i>None</i> ). Если значение свойства равно <i>FixedDialog</i> , то граница окна тонкая и кнопки <b>Развернуть</b> и <b>Свернуть</b> в заголовке не отображаются
BackColor	Цвет формы. Цвет можно задать, выбрав 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
ScaleMode	Единица измерения размеров и координат компонентов, находящихся на форме. Размер и координаты компонентов могут измеряться в твипах ( <i>Twip</i> ), пикселях ( <i>Pixel</i> ), миллиметрах ( <i>Millimeter</i> ) и других единицах
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)



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

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).

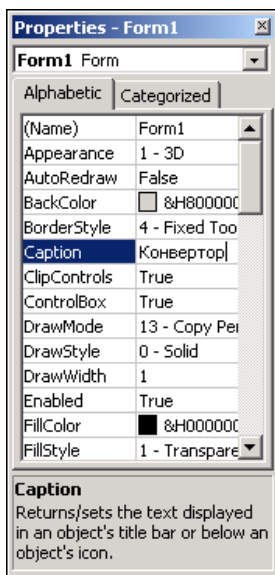
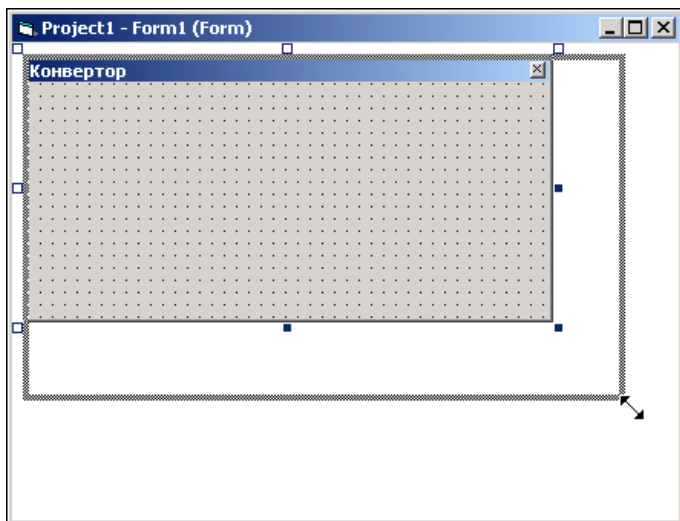


Рис. 2.2. Изменение значения свойства путем ввода строки

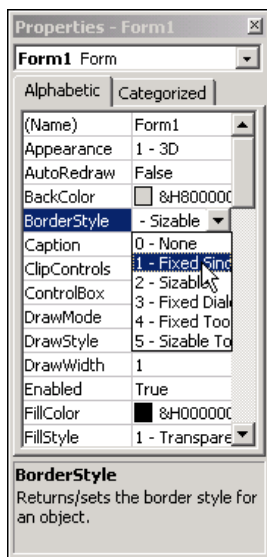
Следует обратить внимание, что ширина и высота формы изменяются в специальных единицах — твипах. Задавать значения свойств `Width` и `Height` в твипах не удобно. Проще захватить

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, численное значение константы `FixedSingle` равно единице, а символьное значение — `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 ICO-файл — 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 формы

Свойство	Значение
Caption	Конвертор

Таблица 2.2 (окончание)

Свойство	Значение
Width	4425
Height	3480
BorderStyle	FixedSingle
MaxButton	False
MinButton	False
StartPosition	CenterScreen
ScaleMode	Pixel
Font	Tahoma, обычный, 10 pt

## Компоненты

Программа пересчета цены из долларов в рубли должна получить от пользователя исходные данные — цену в долларах и курс (соотношение рубля к доллару). Ввод данных с клавиатуры обеспечивает компонент `TextBox` — поле редактирования. Поэтому в форму надо добавить два компонента `TextBox`.

Чтобы добавить в форму компонент `TextBox`, нужно в палитре компонентов сделать щелчок на значке компонента (рис. 2.5), установить указатель мыши в ту точку формы, в которой должен быть левый верхний угол компонента, нажать левую кнопку мыши и переместить указатель мыши в ту точку, в которой должен быть правый нижний угол компонента. После того как кнопка мыши будет отпущена, на форме появится компонент.

Каждому компоненту, добавленному в форму, автоматически присваивается имя, которое формируется из стандартного имени компонента и его порядкового номера. Например, первый компонент `TextBox` получает имя `Text1`, второй — `Text2`. Программист,

путем изменения значения свойства `Name`, может изменить имя компонента. В простых программах имена компонентов, как правило, не меняют.

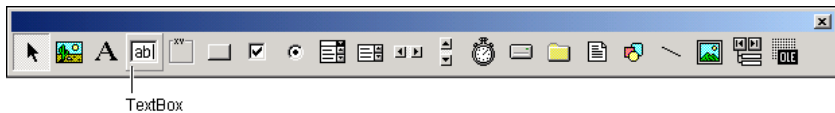


Рис. 2.5. Значок компонента `TextBox`

На рис. 2.6 приведен вид формы программы "Конвертор" после добавления двух компонентов `TextBox`, предназначенных для ввода исходных данных. Один из компонентов выделен. Свойства этого (выделенного) компонента отображаются в окне **Properties**. Чтобы увидеть свойства другого компонента, надо щелкнуть левой кнопкой мыши на его изображении в форме или выбрать имя нужного компонента в раскрывающемся списке, который находится в верхней части окна **Properties**.

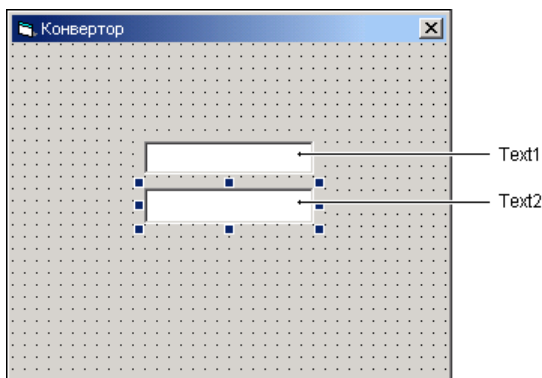


Рис. 2.6. Форма после добавления компонентов `TextBox`

В табл. 2.3 перечислены основные свойства компонента `TextBox`.

Таблица 2.3. Свойства компонента `TextBox`

Свойство	Описание
Name	Имя компонента. Используется для доступа к компоненту и его свойствам
Text	Текст, находящийся в поле редактирования
Left	Расстояние от левой границы компонента до левой границы формы
Top	Расстояние от верхней границы компонента до верхней границы формы
Width	Ширина поля редактирования
Height	Высота поля редактирования
Font	Шрифт, используемый для отображения текста в поле компонента
ForeColor	Цвет текста
Locked	Блокировка. Используется для ограничения возможности изменения текста в поле редактирования. Если значение свойства равно <code>True</code> , то текст в поле редактирования изменить нельзя
MultiLine	Разрешает ( <code>True</code> ) отображение текста в несколько строк
ScrollBars	Управляет отображением полос прокрутки. У компонента может быть полоса вертикальной прокрутки ( <code>Vertical</code> ), горизонтальной ( <code>Horizontal</code> ) или обе полосы прокрутки ( <code>Both</code> ). Если значение свойства равно <code>None</code> , то полосы прокрутки не отображаются
Visible	Позволяет скрыть компонент ( <code>False</code> ) или сделать его видимым ( <code>True</code> )

Изменить размер и положение компонента на форме можно присвоив нужные значения соответственно свойствам `Width`, `Height`, `Left` и `Top` или с помощью мыши.