

# Visual Basic на практике



Среда разработки Visual Basic  
Готовые решения для повседневных задач  
Работа с базами данных  
Взаимодействие с мультимедиа  
Разработка сетевых приложений  
Использование дополнительных компонентов  
Секреты и хитрости Visual Basic

**Решения и опыт  
профессионалов  
VBStreets.ru**

# **Visual Basic на практике**

Под общей редакцией  
Г. И. Магданурова

Санкт-Петербург  
«БХВ-Петербург»  
2008

УДК 681.3.068+800.92VisualBasic  
ББК 32.973.26-018.1  
В41

В41 Visual Basic на практике / Под общ. ред. Г. И. Магданурова. —  
СПб.: БХВ-Петербург, 2008. — 480 с.: ил. + CD-ROM  
ISBN 978-5-9775-0088-3

В книге обобщен многолетний опыт авторов популярного сайта VBStreets.ru. Рассмотрен широкий круг практических вопросов по программированию в Visual Basic 6: трюки и хитрости использования стандартных компонентов, готовые решения для повседневных задач, работа с базами данных, взаимодействие с мультимедиа, разработка приложений для Интернета, работа с оборудованием и многое другое. На компакт-диске содержатся исходные коды программ, рассмотренные в книге, а также дополнительные приложения и компоненты.

*Для программистов*

УДК 681.3.068+800.92VisualBasic  
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>Андрей Смышляев</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Татьяна Кошелева</i>
Дизайн обложки	<i>Инны Тачиной</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.10.07.

Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 38,7.

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

# Оглавление

<b>Введение</b> .....	<b>9</b>
<b>Глава 1. Среда разработки Visual Basic 6</b> .....	<b>11</b>
Знакомство с интегрированной средой разработки.....	11
Основные возможности среды разработки.....	12
Панель инструментов.....	14
Дизайнер форм.....	16
Редактор меню.....	18
Редактор кода.....	19
Клавиши быстрого доступа в редакторе кода.....	20
Настройка среды разработки.....	22
Вкладка <i>Editor</i> .....	23
Вкладка <i>Editor Format</i> .....	25
Вкладка <i>General</i> .....	26
Вкладка <i>Docking</i> .....	27
Вкладка <i>Environment</i> .....	28
Вкладка <i>Advanced</i> .....	28
Вспомогательные инструменты редактора кода.....	29
Утилита для создания заготовок процедур.....	29
Утилита для генерации заготовок кода классов и коллекций.....	30
Браузер объектов.....	32
Использование отладчика Visual Basic.....	33
Точки останова.....	34
Окно <i>Immediate</i> .....	36
Окно <i>Locals</i> .....	37
Окно <i>Watch</i> .....	38
Окно <i>Call Stack</i> .....	41
Visual Data Manager (Визуальный менеджер данных), инструмент для работы с базами данных.....	43
Поддерживаемые форматы баз данных.....	43

Соединение с базой данных с использованием ODBC .....	44
Работа с базой данных .....	48
<b>Глава 2. Графический интерфейс .....</b>	<b>51</b>
Поддержка интерфейса Windows XP .....	51
Элементы управления и изменение размеров формы .....	51
Использование системных иконок .....	53
Использование схемы оформления Windows XP для стандартных элементов управления Windows .....	55
Определение версии Microsoft Windows .....	59
Прозрачная форма.....	61
Изменение регистра символов, вводимых в текстовое окно.....	64
Мигающие заголовки окон.....	66
Надпись как гиперссылка.....	68
Динамическое создание элементов управления .....	71
Реализация Drag&Drop.....	79
Работа с мышью .....	80
Как установить время двойного нажатия клавиши мыши.....	84
Определение положения курсора мыши .....	84
Переместить курсор мыши в заданное место .....	85
Программная замена кнопок мыши.....	85
Скрыть/показать курсор мыши .....	85
Реализация автоматического дополнения выпадающего списка.....	86
<b>Глава 3. Работа с текстом .....</b>	<b>89</b>
Регулярные выражения.....	89
Простейшее регулярное выражение .....	90
Спецсимволы .....	91
Символы начала и конца строки.....	93
Символы повтора и группировки.....	93
Еще несколько спецсимволов .....	93
Символьные классы .....	94
Ответы на вопросы.....	96
Заключение .....	97
Конвертация текста из RTF в HTML.....	98
Проверка орфографии с помощью Microsoft Word.....	101
Определение позиции курсора ввода в текстовом поле .....	102
Повышение скорости замены в тексте .....	106
<b>Глава 4. Работа с сетью.....</b>	<b>115</b>
Использование компонента Winsock.....	115
Выбор протокола.....	115
Установка протокола .....	116
Определение имени компьютера .....	116

Основы TCP-соединения .....	116
Создание простого TCP-сервера.....	117
Создание клиентского TCP-приложения .....	118
Обработка более чем одного запроса на установление соединения.....	120
Основы UDP .....	121
Пример написания приложений "клиент-сервер" .....	124
Немного теории .....	124
Общее описание .....	125
Реализация сервера .....	128
Реализация клиента .....	136
Сложности и проблемы .....	140
Заключение .....	141
Электронная почта.....	141
Общие принципы работы почтового сервера .....	141
Работа с электронной почтой.....	142
Создание веб-сервера.....	145
Простейший веб-сервер.....	145

## **Глава 5. Мультимедиа .....**

<b>Глава 5. Мультимедиа .....</b>	<b>153</b>
Работа с трехмерной графикой .....	153
Подготовка к работе.....	154
Инициализация Direct3D .....	154
Рисуем плоские фигуры.....	157
ZBuffer.....	160
3D, перспектива и матрицы.....	162
Рисуем цилиндр. TriangleStrip.....	167
Текстурирование .....	169
Индексирование вертексов.....	172
Нормали и свет .....	175
Материал.....	178
Организация проекта .....	179
Мип-мэппинг, фильтрация текстур.....	183
Проверка совместимости. <i>D3DCAPS8</i> .....	186
Мультитекстурирование .....	187
Использование прозрачности.....	192
Использование моделей. <i>D3DXMesh</i> .....	196
Имитация отражения. Spherical Environment Mapping.....	199
Преобразование содержимого моделей .....	201
Приближение к реальности .....	204
Шейдер.....	209
Работа со звуком .....	210
Воспроизведение звука.....	210
Захват звука .....	214

<b>Глава 6. Работа с данными.....</b>	<b>221</b>
Создание и использование новой базы данных.....	221
Создание новой базы данных.....	221
Работа с базой данных.....	222
Асинхронная работа с базами данных.....	231
Подключение в фоновом режиме.....	232
Выполнение операций в фоновом режиме.....	235
Работа с данными в фоновом режиме.....	238
Работа с объектом <i>ADODB.Command</i> .....	242
Запрос с параметрами.....	246
Вызов хранимой процедуры.....	247
Подготовленные запросы.....	250
Использование потоков ADO.....	252
Количество измененных при выполнении запроса записей.....	253
Текстовый файл как база данных.....	253
<b>Глава 7. Разное.....</b>	<b>269</b>
Создание консольного приложения.....	269
Работа с файлами File System Object.....	274
Создание папки по несуществующему пути.....	279
Рекурсивный поиск папок.....	280
Получение пути к директории Windows.....	283
Открытие файла в приложении по умолчанию.....	284
Скачивание файла из сети Интернет.....	285
Создание ярлыка.....	285
Закрытие окна другой программы.....	285
Получение имени класса, которому принадлежит окно.....	286
Найти и обезвредить.....	287
Поиск окна и изменение его заголовка.....	288
Вызов метода по имени.....	289
Добавление иконок в меню.....	289
Создание всплывающего меню.....	291
Создание меню с кнопкой.....	292
Создание меню.....	293
Работа с реестром.....	294
Функция <i>SaveSetting</i> .....	294
Функция <i>DeleteSetting</i> .....	295
Функция <i>GetSetting</i> .....	295
Функция <i>GetAllSettings</i> .....	295
Работа с реестром с использованием API-функций.....	296
Работа с бинарным файлом ресурсов.....	301
Получение информации о текущих процессах.....	302
Использование <i>SendMessage</i> .....	304
Количество процессов с заданным именем.....	305

Изменение настроек дисплея .....	312
Пример использования .....	314
Получение установленных шрифтов .....	316
Оптимизация приложений.....	318
Создание библиотеки ресурсов.....	319
Оптимизация потребления ресурсов системы .....	319
Оптимизация скорости исполнения приложения .....	320
Установка горячих клавиш для активирования своей программы .....	322
Связывание расширения с программой .....	323
<b>Глава 8. Под капотом .....</b>	<b>327</b>
Тип данных <i>Double</i> .....	327
Сравнение числа со строкой .....	332
Устройство строкового типа .....	333
Теория .....	335
Исследование: возврат строк из API-функций и восстановление строки по указателю .....	336
Заключение .....	338
Функции для работы с указателями.....	339
Сущность массивов в Visual Basic.....	340
Вызов функций по указателю .....	345
Прямой вызов <i>CallWindowProc</i> .....	347
И в обход... .....	348
В объектно-ориентированный обход... .....	351
Сусанин отдыхает... .....	353
Применимость .....	356
И последнее .....	357
Работа с СОМ-объектами, не поддерживаемыми <i>IDispatch</i> .....	357
Работа с описателями .....	359
Обнаружение скрытых процессов .....	366
"Убиваем" антивирус Касперского.....	374
Занятые файлы: перечисление .....	377
Занятые файлы. Закрытие чужих описателей.....	380
Занятые файлы. Копирование .....	383
Доступ к безымянным параметрам.....	386
Работа с Рабочими столами и семафорами.....	387
Удаление приложения после завершения работы .....	390
Внедрение VB-кода в другие процессы .....	394
Использование фиберов в VB для одновременного выполнения нескольких задач..	401
Использование фиберов в перечислителях .....	404
Перечислители.....	404
Фиберы .....	406
Пример .....	407
Заключение .....	411



---

Использование фиберов для расщепления выполнения процедуры.....	411
Расщепление выполнения программы .....	411
Устройство фиберов .....	412
Расщепление фиберов.....	415
Приложение. Псевдокод функций Windows API, работающих с фиберами.....	420
Как пометить свой объект как безопасный для использования в сценариях.....	428
<b>Глава 9. Дополнительная полезная информация.....</b>	<b>429</b>
Сообщения об ошибках Visual Basic .....	429
Основные свойства формы.....	456
<b>Приложение. Описание компакт-диска.....</b>	<b>465</b>
<b>Предметный указатель .....</b>	<b>469</b>

# Введение

Дорогой читатель, перед вами сейчас уникальный труд — книга, ставшая результатом труда различных авторов, многие из которых даже никогда не были знакомы. Данная книга является сборником статей и материалов, опубликованных в разное время на сайте [www.VBStreets.ru](http://www.VBStreets.ru), специально доработанных для этой книги.

Мы постарались включить в книгу те материалы, которые могут быть полезны разработчикам на Visual Basic, обладающим разным опытом — от новичка, впервые запустившего среду разработки Visual Basic, до профессионала, жаждущего познать скрытые возможности Visual Basic. В связи с этим уровень материала, представленный в книге, достаточно разный и, мы искренне надеемся, что это должно добавить в книгу небольшую интригу.

Мы не ставили перед собой цель детально описать все возможности Visual Basic, ведь для этого уже существуют десятки книг — мы постарались включить лишь те материалы, которые являются ответами на наиболее часто задаваемые вопросы на форумах сайта [www.VBStreets.ru](http://www.VBStreets.ru). Это значит, что не имеет смысла читать книгу "от корки до корки", нужно выбрать интересующие вас раздел или статью, и воспользоваться представленной в ней информацией. Исключением, пожалуй, может быть лишь *глава 5*, которая посвящена мультимедиа и является достаточно подробным и целостным изложением материала об использовании возможностей DirectX.

Надеюсь, что вы не только извлечете пользу, но и получите удовольствие от прочтения данной книги, авторами статей в которой являются: Алибек Болатов, Сергей Гергерт, Михаил Ильин, Андрей Чурсин, Николай Амеличев, Александр Егоров, Евгений Куколь, Павел Дерманович, Игорь Шаронов, Данила Беляев, Артем Скробов, Гайдар Магдануров, Максим Павлов, Валерий Семчуков, Александр Кузнецов, Александр Иванов, Сергей Русаков, Сергей Никифоров. Мы намеренно не стали приводить список соотнесения статей и

их авторов, чтобы подчеркнуть, что данная книга является плодом труда сообщества разработчиков.

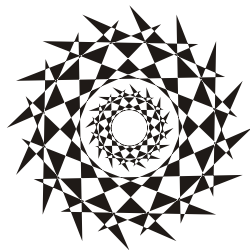
Как и сам сайт **www.VBStreets.ru**, данная книга является некоммерческим проектом, возможный доход от продаж которой будет использован на дальнейшее развитие сообщества Visual Basic-программистов. Поэтому, покупая эту книгу, вы помогаете своим коллегам-разработчикам. Согласитесь, благородная цель?

Не станем выделять благодарности в отдельный раздел, как это принято в современной технической литературе, а просто скажем большое спасибо всем авторам, а также читателям книги. Спасибо вам!

*Успехов в программировании!*

Гайдар Магдануров,  
главный редактор VBStreets.ru

# ГЛАВА 1



## Среда разработки Visual Basic 6

Здесь собраны материалы, которые будут полезны для программистов, начинающих осваивать *интегрированную среду разработки* (IDE, Integrated Development Environment) Visual Basic 6 и входящие в ее состав различные полезные инструменты.

### Знакомство с интегрированной средой разработки

Прежде, чем приступать к серьезной работе с тем или иным инструментом разработки программного обеспечения, так же как и при работе с любым другим многофункциональным программным продуктом, необходимо детально изучить возможности самого инструмента. Без этого работа не будет удовольствием, а превратится в регулярное "изобретение велосипеда", поиск вспомогательных инструментов или пытку, связанную с нарастающим недовольством несовершенным программным продуктом, возможности которого вы не удосужились изучить.

Согласитесь, потратив несколько часов на изучение инструмента, вы будете знать наилучший способ выполнения поставленной задачи. А если учесть, что изучать возможности среды разработки придется лишь один раз, а использовать возможности этой среды разработки придется многократно, можно предположить, что знание инструмента принесет значительную пользу и огромную экономию рабочего времени в будущем.

Постарайтесь выделить время на то, чтобы "поиграть" установками рабочего окружения среды разработки и понять, какие настройки комфортны именно для вас и удобны для вашей текущей работы. Ведь установки среды разра-

ботки по умолчанию могут быть очень удобны для вашего коллеги, но неудобны для вас. Зачем мучиться, если можно настроить все по своему вкусу.

Надеюсь, что вы убедились в необходимости полноценного знания среды разработки и с радостью приступите к ее детальному изучению. В этой статье приведено описание основных возможностей среды разработки и наиболее важных диалоговых окон, предоставляющих возможности настройки и изменения параметров рабочего окружения.

## Основные возможности среды разработки

При первом запуске среды разработки и создании **Standard EXE** (Обычное приложение), вы обнаружите несколько основных окон:

- ❑ **Toolbox** (Панель инструментов), расположенная слева (рис. 1.1);
- ❑ **Form Designer** (Дизайнер форм) — окно с созданной для вас формой **Form1**, занимающее центральное место в среде разработки (рис. 1.2);



Рис. 1.1. Панель инструментов Visual Basic

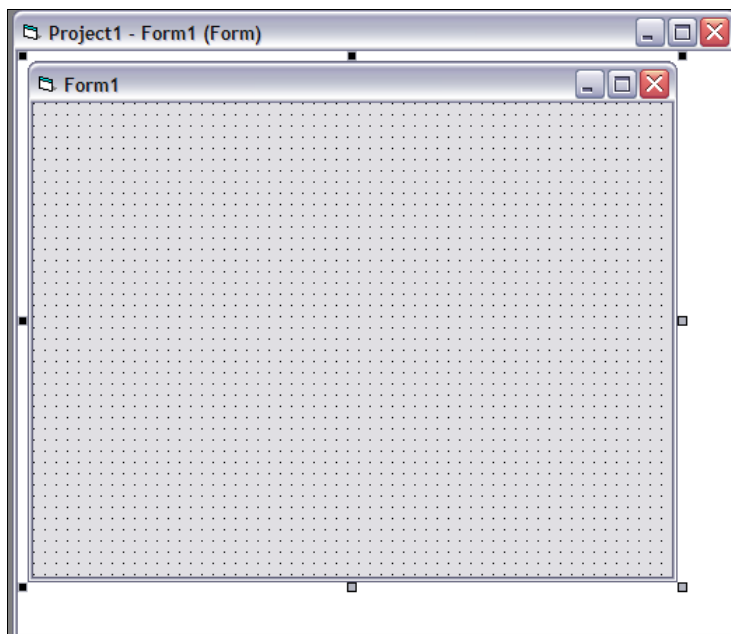


Рис. 1.2. Дизайнер форм

- ❑ **Project Explorer** (Дерево проекта) — список составляющих компонентов проекта (форм, модулей, классов и др.), находящийся в верхнем правом углу (рис. 1.3);

- ❑ **Properties Window** (Окно свойств), расположенное справа (рис. 1.4);
- ❑ **Form Layout** (Расположение формы) — окно выбора положения формы на экране, находящееся в правом нижнем углу (рис. 1.5).

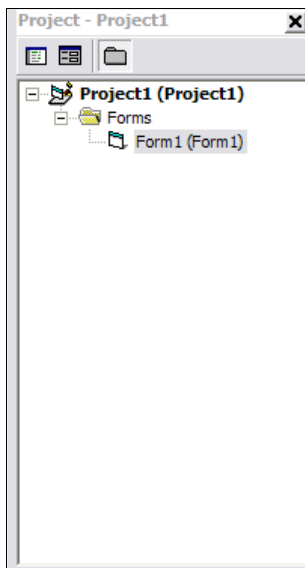


Рис. 1.3. Список составляющих компонентов проекта

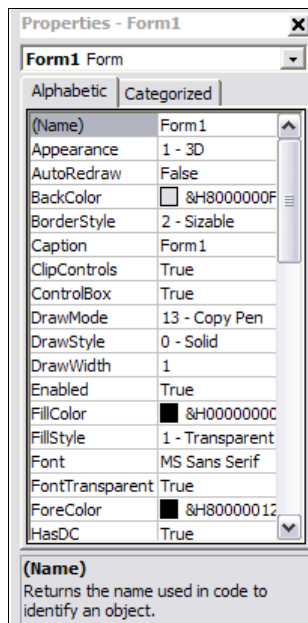


Рис. 1.4. Окно свойств

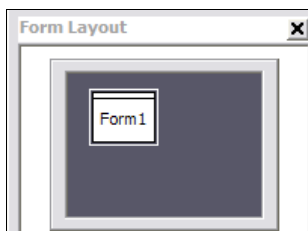


Рис. 1.5. Окно выбора положения формы на экране

Среда разработки Visual Basic позволяет визуально создавать дизайн форм, используемых в приложении путем помещения компонентов, называемых *элементами управления*, иконки которых расположены на панели инструментов, на форму, открытую в дизайнера форм.

Выбрав форму или элемент управления в окне свойств можно посмотреть доступные свойства и их значения. Окно свойств имеет два возможных вари-

анта отображения списка — **Alphabetic** (По алфавиту) и **Categorized** (По категориям). Если вы только начинаете работать со средой Visual Basic, более удобным для вас может быть список по категориям, поскольку он позволяет легче находить свойства, относящиеся ко внешнему виду или поведению элемента управления.

С помощью окна для установления положения формы на экране можно визуально, путем перетаскивания изображения окна, выбрать, в каком месте экрана будет отображаться форма при появлении.

В окне составляющих компонентов проекта можно выбрать форму, модуль или класс, с которым будет вестись работа. С помощью кнопок **View Code** (Просмотр кода) и **View Object** (Просмотр объекта) в верхней части окна можно перейти к редактированию кода или визуальному редактированию компонента в дизайнера форм.

Далее мы более детально рассмотрим возможности панели инструментов, дизайнера форм и редактора кода Visual Basic.

## Панель инструментов

**Toolbox** (Панель инструментов) по умолчанию содержит лишь список основных и наиболее часто используемых компонентов. Список компонентов может быть расширен: на панель инструментов могут быть добавлены дополнительные элементы управления. Для этого нужно щелкнуть правой кнопкой мыши по панели инструментов и выбрать в контекстном меню пункт **Components** (Компоненты) или выбрать пункт **Components** (Компоненты) в меню **View** (Вид), или нажать комбинацию клавиш <Ctrl>+<T>.

### Примечание

Настоятельно рекомендуется запомнить наиболее полезные комбинации клавиш, поскольку использование клавиатуры при разработке проектов значительно ускоряет работу, по сравнению с использованием мыши для поиска нужных кнопок или пунктов меню.

На рис. 1.6 представлен вид окна выбора компонентов — **Components** (Компоненты). Для того чтобы добавить компоненты на панель инструментов, нужно отметить соответствующие названия компонентов в списке, либо, если требуемые названия в списке отсутствуют, щелкнуть по кнопке **Browse** (Выбрать) и указать библиотеку, компоненты из которой необходимо добавить в список.

Наиболее полезными для большинства приложений являются компоненты Microsoft Windows Common Controls, которые содержат такие компоненты,

как: список изображений (ImageList), дерево (TreeView), строка состояния (StatusBar), панель инструментов (Toolbar), индикатор выполнения процесса (ProgressBar), вкладки (TabStrip) и др.

Для удобства работы компоненты можно группировать на разных вкладках, на панели инструментов. Чтобы добавить новую вкладку щелкните правой кнопкой по панели инструментов и выберите **Add Tab...** (Добавить вкладку...). Введите название вкладки и нажмите **ОК** в появившемся диалоговом окне. После этого щелкните по появившемуся заголовку вкладки на панели инструментов и повторите процедуру с добавлением компонентов. В результате, вновь отмеченные компоненты будут появляться на выбранной вкладке (рис. 1.7).

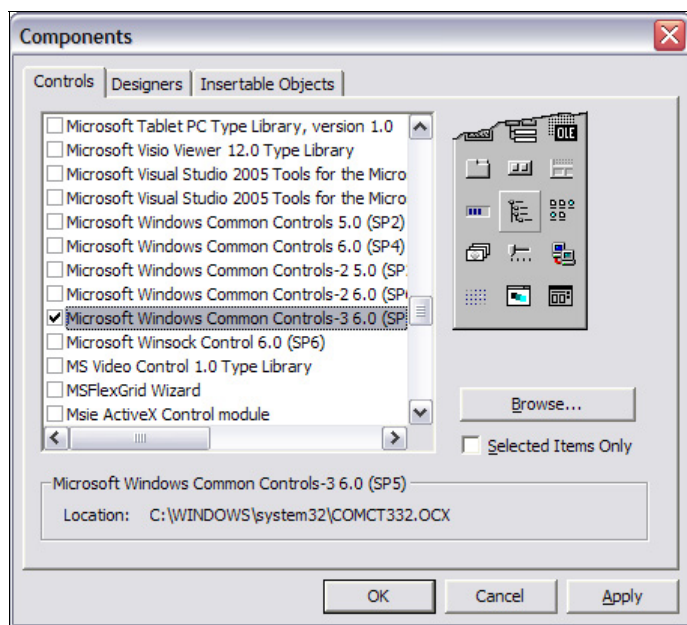


Рис. 1.6. Окно выбора компонентов



Рис. 1.7. Панель инструментов с добавленной вкладкой ComC

Набор компонентов, представленных на панели инструментов Visual Basic привязывается к конкретному проекту, открытому в среде разработки. Таким образом вы можете для каждого конкретного проекта добавить лишь те компоненты, которые используются на формах этого конкретного проекта, и среда Visual Basic будет показывать собственный список компонентов для каждого проекта.



## Дизайнер форм

Задачей дизайнера форм является удобное размещение элементов управления и визуальное редактирование свойств этих компонентов с помощью окна свойств. Основная функциональность по взаимному размещению элементов дизайнера форм доступна из меню **Format** (Формат). Описание команд приведено в табл. 1.1.

**Таблица 1.1.** Команды меню **Format** (Формат)

Пункт меню	Подпункт меню	Описание
<b>Align</b> (Выравнивание)	<b>Lefts</b> (Слева)	Выравнивание выделенных элементов управления по левой границе
	<b>Centers</b> (По центру)	Выравнивание выделенных элементов управления относительно центров элементов по горизонтали
	<b>Rights</b> (Справа)	Выравнивание выделенных элементов управления по правой границе
	<b>Tops</b> (Вверху)	Выравнивание выделенных элементов управления по верхней границе
	<b>Middles</b> (Посередине)	Выравнивание выделенных элементов управления относительно центра элементов по вертикали
	<b>Bottoms</b> (Внизу)	Выравнивание выделенных элементов управления по нижней границе
	<b>To Grid</b> (По сетке)	Выравнивание соответствии с сеткой (Grid) на форме
<b>Make same size</b> (Выравнивание по размерам)	<b>Width</b> (По ширине)	Уравнивание выделенных элементов управления по ширине
	<b>Height</b> (По высоте)	Уравнивание выделенных элементов управления по высоте
	<b>Both</b> (По ширине и высоте)	Уравнивание выделенных элементов управления по ширине и высоте
<b>Size to Grid</b> (Размеры по сетке)		"Подгонка" размеров элементов управления под сетку
<b>Horizontal Spacing</b> (Промежутки по горизонтали)	<b>Make Equal</b> (Сделать одинаковыми)	Уравнивание промежутков между выделенными элементами управления по горизонтали
	<b>Increase</b> (Увеличить)	Увеличение расстояния между выделенными элементами управления по горизонтали

Таблица 1.1 (окончание)

Пункт меню	Подпункт меню	Описание
	<b>Decrease</b> (Уменьшить)	Уменьшение расстояния между выделенными элементами управления по горизонтали
	<b>Remove</b> (Удалить)	Совмещение выделенных элементов управления "граница в границу" по горизонтали
<b>Vertical Spacing</b> (Промежутки по вертикали)	<b>Make Equal</b> (Сделать одинаковыми)	Уравнивание промежутков между выделенными элементами управления по вертикали
	<b>Increase</b> (Увеличить)	Увеличение расстояния между выделенными элементами управления по вертикали
	<b>Decrease</b> (Уменьшить)	Уменьшение расстояния между выделенными элементами управления по вертикали
	<b>Remove</b> (Удалить)	Совмещение элементов управления "граница в границу" по вертикали
<b>Center in Form</b> (Отцентрировать в форме)	<b>Horizontally</b> (По горизонтали)	Размещение выделенного элемента управления по центру формы по горизонтали
	<b>Vertically</b> (По вертикали)	Размещение выделенного элемента управления по центру формы по вертикали
<b>Order</b> (Порядок)	<b>Sent to Back</b> (Поместить вниз)	Размещение выделенного элемента под остальными
	<b>Bring to Front</b> (Поместить наверх)	Размещение выделенного элемента над остальными
<b>Lock Controls</b> (Заблокировать элементы управления)		Элементы управления на форме блокируются

При использовании команд из меню **Format** стоит учесть, что за основу для большинства команд берется первый выделенный элемент или расстояние между первыми двумя элементами, в зависимости от типа выполняемой команды. Если выделяются несколько элементов в столбце или строке и выбирается команда для выравнивания расстояния между элементами, то будет усреднено расстояние на основании расстояния между первым и последним элементом.

Использование возможностей, представленных в меню **Format**, позволяет просто и быстро создавать удобные для использования формы. Нет необхо-

димости долго выверять размеры элементов и тщательно прицеливаться курсором мыши для того, чтобы правильно расположить элементы управления. Достаточно поместить несколько элементов на форму и, с помощью команд из меню **Format**, расположить элементы в соответствии с вашим планом.

## Редактор меню

Для визуального редактирования меню можно выбрать пункт **Menu Editor** (Редактор меню) из меню **Tools** либо щелкнуть правой кнопкой мыши по форме и выбрать пункт **Menu Editor** из выпадающего меню, либо нажать комбинацию клавиш <Ctrl>+<E>. В результате откроется окно редактора меню **Menu Editor**, показанное на рис. 1.8.

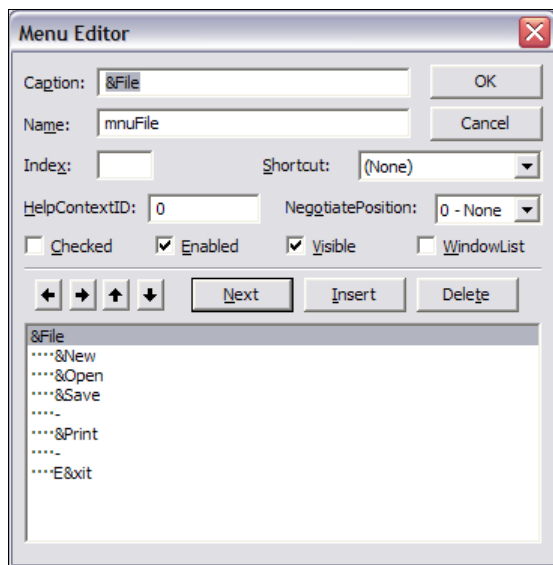


Рис. 1.8. Редактор меню

Пункты меню в редакторе располагаются в виде дерева — пункты первого уровня будут расположены в строке меню формы, вложенные пункты меню разного уровня создаются путем добавления новых пунктов и установления вложенности с помощью кнопок-стрелок в редакторе меню, как это показано на рис. 1.8.

Для того чтобы пункт меню можно было вызвать с клавиатуры с помощью нажатия комбинация клавиш <Alt>+*буква* нужно вставить в название пункта меню символ & перед той буквой, которая будет использована для доступа к пункту меню с клавиатуры. Например, если на форме создано меню, изобра-

женное на рис. 1.8, то для вызова пункта меню **Exit** (Выход) нужно нажать последовательно <Alt>, затем <F>, затем <X>. Сначала будет развернуто меню **File** (Файл), поскольку буква F отмечена символом &, затем пункт меню **Exit**, поскольку за символом & в слове Exit следует буква X. Стоит отметить, что на одном уровне меню не должно быть нескольких повторяющихся букв, отмеченных символом &.

Для того чтобы использовать символ & в названии пункта меню, необходимо вставить символ & два раза. Например, если вы хотите создать меню с названием M&M, то необходимо задать название в редакторе меню как M&&M.

Для назначения комбинации клавиш быстрого доступа нужно установить свойство **Shortcut** (Ярлык) в незанятую комбинацию клавиш.

## Редактор кода

Редактор кода Visual Basic обладает возможностями подсветки синтаксиса, автоматического выделения неверного с синтаксической точки зрения кода, быстрой навигации между определенными в коде методами и свойствами,

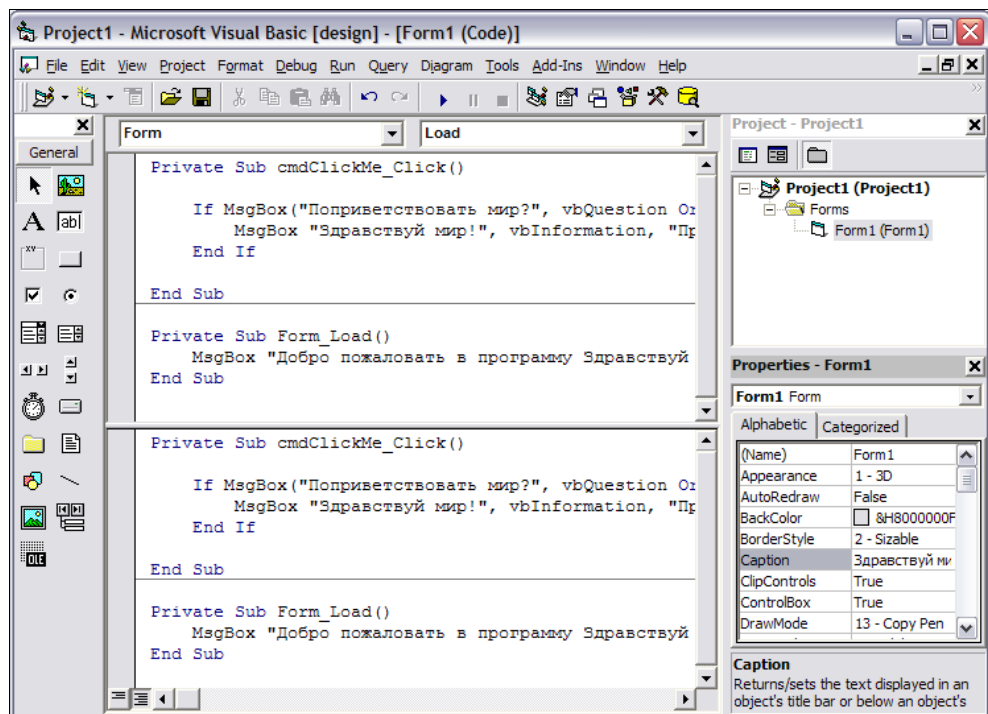


Рис. 1.9. Разделение окна редактора кода

перечисления доступных методов и свойств (всплывающая подсказка IntelliSense) и автоматического дополнения кода.

### Примечание

Если вы используете мышь с колесиком, то прокручивание кода может не работать. В этом случае вам необходимо выполнить действия, описанные в статье, расположенной по адресу <http://support.microsoft.com/kb/837910/ru>.

Весьма удобной возможностью редактора кода является разделение окна на две части так, как это показано на рис. 1.9.

Для того чтобы перевести редактор кода в этот режим, необходимо дважды щелкнуть по элементу полосы прокрутки, показанному на рис. 1.10, либо, нажав на этот элемент, перетащить разделительную полосу в удобное для вас положение. Также можно выбрать пункт **Split** (Разделить) в меню **Window** (Окно).

Для автоматического создания заготовок обработчиков событий в редакторе кода используются выпадающие списки (рис. 1.11). Слева располагается список элементов управления, а справа — список обработчиков событий. Реализованные обработчики выделяются с помощью полужирного шрифта.



Рис. 1.10. Фрагмент полосы прокрутки для разделения окна редактора кода

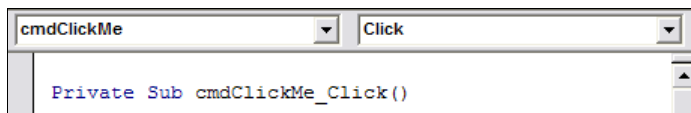


Рис. 1.11. Выпадающие списки с именами элементов управления и обработчиков событий

## Клавиши быстрого доступа в редакторе кода

Для работы с текстом в редакторе кода существует набор комбинаций клавиш быстрого доступа, приведенный в табл. 1.2.

Таблица 1.2. Основные клавиши быстрого доступа

Комбинация клавиш	Описание
<Alt>+<F8>	Выравнивание выделенного участка кода
<Alt>+<Shift>+<T>	Взаимная замена предыдущей и последующей строк кода
<Ctrl>+<J>	Переход к закрывающей скобке
<Ctrl>+<Backspace>	Удаление слова слева от положения курсора

Таблица 1.2 (продолжение)

Комбинация клавиш	Описание
<Ctrl>+<Delete>	Удаление слова справа от положения курсора
<Ctrl>+<End>	Перемещение курсора в конец документа или списка
<Ctrl>+<F>	Поиск в тексте
<Ctrl>+<F1>	Показать справку по выделенному идентификатору (ключевому слову, имени объекта и т. п.)
<Ctrl>+<H>	Поиск и замена в тексте
<Ctrl>+<Home>	Переход в начало документа или списка
<Ctrl>+<I>	Показать информацию о методе или свойстве
<Ctrl>+<J>	Показать список членов класса
<Ctrl>+<Shift>+<F2>	Переход к предыдущей позиции курсора
<Ctrl>+<Shift>+<]>	Выделение текста до соответствующей закрывающей скобки
<Ctrl>+<Shift>+<End>	Выделение текста до конца документа
CTRL+<Shift>+<Home>	Выделение текста до начала документа
<Ctrl>+<Shift>+<I>	Отобразить информацию о параметрах вызываемого метода
<Ctrl>+<Shift>+<Влево>	Выделение слова слева от курсора
<Ctrl>+<Shift>+<Вправо>	Выделение слова справа от курсора
<Ctrl>+<Shift>+<T>	Взаимная замена предыдущего и последующего слов либо замена слов по разные стороны от оператора, например: a = b превращается в b = a
<Ctrl>+<Shift>+<U>	Приведение выделенного текста к верхнему регистру
<Ctrl>+<Shift>+<W>	Выделение слова в позиции курсора
<Ctrl>+<Пробел>	Автоматическое дополнение введенного идентификатора в коде
<Ctrl>+<T>	Взаимная замена предыдущего и последующего символов в позиции курсора
<Ctrl>+<U>	Приведение выделенного текста к нижнему регистру
<Ctrl>+<Вверх>	Прокрутка текста на один экран вверх
<F1>	Отобразить контекстную справку
<F3>	Поиск следующего совпадения с тем фрагментом, что был ранее задан в диалоговом окне поиска
<Shift>+<Вниз>	Расширить выделение на одну строку текста вниз

Таблица 1.2 (окончание)

Комбинация клавиш	Описание
<Shift>+<End>	Расширить выделение до конца текущей строки
<Shift>+<F2>	Переход к определению функции или свойства, на идентификаторе которого установлен курсор ввода
<Shift>+<F3>	Поиск предыдущего совпадения с тем фрагментом, что был ранее задан в диалоговом окне поиска
<Shift>+<F7>	Переход в дизайнер формы для текущего открытого файла исходного кода
<Shift>+<Home>	Расширить выделение до начала текущей строки
<Shift>+<PageDown>	Расширить выделение до конца следующего экрана
<Shift>+<PageUp>	Расширить выделение до начала следующего экрана
<Shift>+<Tab>	Смещение выделенного текста на один символ табуляции
<Shift>+<Вверх>	Расширить выделение на одну строку текста вверх

Список возможных методов и свойств для объекта появляется автоматически, при вводе точки после идентификатора объекта, а чтобы вызвать список — нужно нажать комбинацию клавиш <Ctrl>+<J> или <Ctrl>+<Пробел>. Для отображения списка констант нужно установить курсор в ту позицию, где должна быть константа, и нажать <Ctrl>+<Shift>+<I>. Для отображения подсказки для метода или свойства нужно установить курсор на имя этого метода или свойства и нажать <Ctrl>+<I>.

Для смещения блока кода на символ табуляции нужно выделить желаемый фрагмент кода и нажать <Tab>, для того чтобы убрать один символ табуляции из выделения, нужно нажать комбинацию клавиш <Shift>+<Tab>. Стоит отметить, что это не работает для одной строки, т. е. если вы выделите только одну строку и нажмете <Tab> или <Shift>+<Tab>, то строка будет заменена на символ табуляции.

## Настройка среды разработки

Все основные настройки среды разработки Visual Basic располагаются в окне **Options** (Опции), которое можно вызвать, выбрав пункт **Options** в меню **Tools**. На рис. 1.12 представлено окно **Options**.

Большинство параметров настройки среды разработки, которые вам могут понадобиться, для того чтобы настроить среду "под себя", находятся в окне **Options**, на шести вкладках, описание которых будет приведено далее.

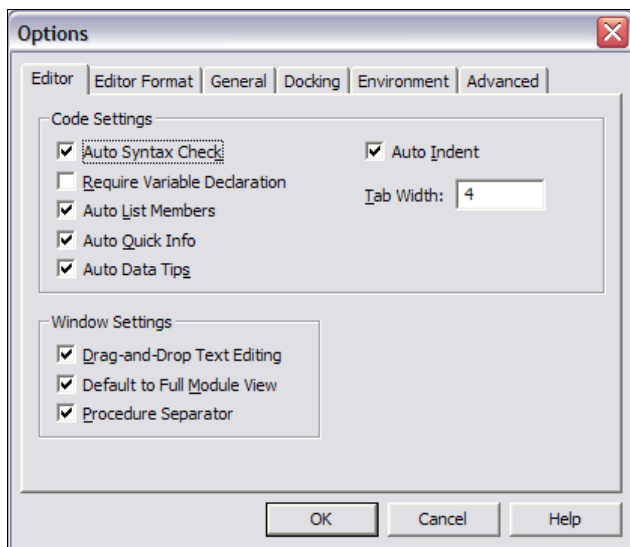


Рис. 1.12. Окно Options

## Вкладка Editor

На этой вкладке **Editor** (Редактор) собраны настройки, влияющие непосредственно на поведение редактора кода. Поскольку редактор кода является основной частью среды разработки, с которой вам придется работать, то имеет смысл настроить его так, чтобы работа была продуктивной и максимально комфортной. По статистике, даже когда над проектом работает всего один разработчик, на написание и редактирование кода уходит до 80% времени и лишь примерно 20% уходит на редактирование дизайна форм и работу с другими утилитами, интегрированными в среду разработки.

Поведение редактора при работе с кодом определяется группой настроек **Code Settings** (Установки кода):

- Auto Syntax Check** (Автоматическая проверка синтаксиса) — если этот флажок установлен, то редактор будет автоматически проверять допустимость введенного кода и немедленно сообщать об ошибках. На начальных этапах работы в Visual Basic имеет смысл включить эту возможность, поскольку это приучит вас не допускать грубых ошибок при вводе кода. В дальнейшей работе разумным может быть отключение этой функции, поскольку если вы профессионально работаете с кодом, то иногда бывает удобно написать лишь часть выражения, перейти к другой строке и потом вернуться к недописанной строке. Кроме того, даже при выключенной функции **Auto Syntax Check** редактор будет подсвечивать неправильные



строки кода красным цветом. В любом случае, использовать ли эту функцию, зависит полностью от вас и вашего стиля кодирования;

- ❑ **Require Variable Declaration** (Требовать объявление переменной) — данная функция заставляет редактор кода при создании новых файлов автоматически вставлять инструкцию `Option Explicit`, которая указывает, что все используемые переменные должны быть объявлены. Стоит всегда использовать эту функцию, т. к. строгое объявление типа переменных приводит к более логичному программному коду и повышает производительность программ. Поскольку использование переменных без объявления означает, что переменная будет объявлена компилятором Visual Basic с типом `Variant`, который является наиболее ресурсоемким из всех встроенных типов переменных в Visual Basic;
- ❑ **Auto List Members** (Выводить список членов) — данная функция редактора выводит список возможных членов класса или список допустимых значений перечислимого во время ввода кода;
- ❑ **Auto Quick Info** (Автоподсказка) — опция включает автоматическую подсказку, содержащую описание функции и принимаемых ею параметров. Особенно эта возможность полезна начинающим программистам на Visual Basic, поскольку, не имея обширной практики программирования на Visual Basic, запомнить списки аргументов функций и порядок их следования очень сложно;
- ❑ **Auto Data Tips** (Подсказки о данных) — эта функция позволяет посмотреть значение переменной во время отладки путем наведения курсора мыши на ее идентификатор. Используя эту функцию при отладке, можно значительно сэкономить время на поиск ошибок в программной логике;
- ❑ **Auto Indent** (Авторасстановка отступов) — функция редактора, расставляющая отступы в программном коде. При написании больших функций и методов категорически необходимо использовать отступы для повышения читаемости кода. Помните, код значительно чаще читают и модифицируют, чем пишут, поэтому код должен быть написан с учетом возможности быстрого прочтения и анализа человеком в будущем. В текстовом поле **Tab Width** (Ширина отступа) задается число пробелов, которые редактор будет автоматически вставлять при нажатии клавиши `<Tab>`. Почему вместо символа табуляции используются пробелы? Дело в том, что код должен оставаться в читабельном виде при работе в любом редакторе, не только в редакторе кода среды разработки Visual Basic, а ширина смещения при использовании символов табуляции может сильно различаться в разных редакторах. Символ пробела практически всегда занимает место, сопоставимое с размером символа.

Общее поведение редактора, не связанное непосредственно с написанием кода, задается группой настроек **Window Settings** (Настройки для окон):

- ❑ **Drag-and-Drop Text Editing** (Разрешить Drag&Drop для текста) — если этот флажок установлен, то вы можете выделить фрагмент текста и перетащить его мышью. Поскольку точность перетаскивания очень сильно зависит от времени суток, количества прошедших праздников и усталости организма разработчика, то значительно удобнее перемещать блоки текста, предварительно вырезав нужный блок в буфер обмена (<Ctrl>+<X>) и затем вставив его в нужное положение (<Ctrl>+<V>);
- ❑ **Default to Full Module View** (Отображать полный код по умолчанию) — при установленном флажке в редакторе кода всегда будет отображаться полный код модуля (формы, класс и т. п.). Если вам удобнее работать с кодом отдельных функций и свойств, то уберите этот флажок. Также для включения этой возможности редактора можно использовать переключатель, показанный на рис. 1.13. Если включен режим просмотра по одной функции или свойству, то для перемещения между разными функциями и свойствами нужно использовать выпадающие списки, показанные на рис. 1.11;
- ❑ **Procedure Separator** (Разделитель между процедурами) — включает горизонтальную разделительную полосу между методами и свойствами в редакторе кода. Это удобно, когда вы работаете в режиме отображения полного исходного кода модуля, поскольку так визуальнее проще перемещаться между разными функциями.



Рис. 1.13. Переключатель отображения кода всего модуля/текущей функции

## Вкладка **Editor Format**

С помощью вкладки **Editor Format** (Формат редактора) вы можете настроить цветовую схему редактора кода Visual Basic. Поскольку восприятие цвета и отношение к разным цветам сугубо индивидуальны для каждого человека, настройка цветовой схемы может значительно повысить комфорт работы в среде разработки Visual Basic. Также без настройки цветовой гаммы тяжело обойтись в том случае, если вы работали в другой среде разработки и привыкли к другой цветовой схеме.

Для изменения параметров достаточно выбрать требуемый пункт из списка. После того как нужный пункт выбран, следует установить желаемый шрифт (Font), размер шрифта (Size), наличие поля слева от редактора кода (Margin Indicator Bar), цвет текста (Foreground), цвет фона (Background) и цвет инди-

катора на поле слева от редактора кода, если таковой применим для выбранного стиля оформления.

## Вкладка *General*

На вкладке **General** (Общие) собраны те настройки среды разработки, которые не удостоились места на отдельных вкладках, но это не значит, что эти настройки не нужны. Более того, на вкладке **General** (Общие) располагаются весьма полезные опции.

Группа **Form Grid Settings** (Настройки сетки форм) содержит настройки сетки, используемой в дизайнерах форм.

- ❑ **Show Grid** (Показать сетку) — при установленном флажке в дизайнерах форм будет нарисована сетка, расстояние между линиями в которой задается параметрами **Width** (Ширина) и **Height** (Высота). Эти параметры могут принимать значения от 24 до 1188, но установленные по умолчанию значения 120 достаточно удобны для работы в редакторе. По неизвестной причине начинающих программистов сильно раздражает сетка на формах, поэтому они предпочитают отключать эту функцию. Однако эта функция может быть полезна, поскольку позволяет легко оценить расположение элементов управления. Согласитесь, иногда кажется, что форма выглядит "как-то не так", но вы не можете понять, почему, — зачастую причина в том, что равнозначные элементы управления чуть-чуть различаются по размерам или смещены с одной линии. При включенной сетке заметить такие расхождения очень просто даже невооруженным глазом.
- ❑ **Align Controls to Grid** (Выводить элементы по сетке) — эта функция заставляет дизайнера форм размещать элементы управления точно в соответствии с сеткой. Вы можете провести небольшой эксперимент — сделать значения параметров **Width** и **Height** сетки достаточно большими и при включенной функции **Align Controls to Grid** разместить несколько кнопок разного размера на форме. Как вы сможете убедиться, функция **Align Controls to Grid** не позволит вам расположить кнопки на форме абы как. Если функция **Align Controls to Grid** отключена, всегда можно заставить дизайнера форм выровнять элементы управления в соответствии с сеткой с помощью меню **Format** и пунктов **Size to Grid** (Размер по сетке) и **Align (Выводить) | to Grid** (По сетке).

Под группой **Form Grid Settings** (Настройки сетки форм) расположены две опции, не выделенные в отдельную группу:

- ❑ **Show ToolTips** (Показывать всплывающие подсказки) — если эта функция включена, то при наведении курсора мыши на идентификатор объекта, метода или свойства будет выведено краткое описание на всплывающей подсказке (Tool Tip).

- ❑ **Collapse Proj. Hides Windows** (Прятать окна при сворачивании проекта) — данная опция сообщает редактору кода о том, что при сворачивании списка файлов проекта в окне составляющих компонентов проекта (см. рис. 1.3.) все открытые окна редактора кода и дизайнера форм, имеющие отношение к этому проекту, должны быть свернуты автоматически.

Группа **Error Trapping** (Организация отладочных ловушек) позволяет установить поведение отладчика среды разработки при возникновении ошибки времени выполнения программы. Установленный флажок **Break on All Errors** (Прерывать при любой ошибке) заставляет отладчик прекращать выполнение программы при возникновении любой ошибки. При установленном флажке **Break in Class Module** (Прерывать при ошибке в классе) программа будет остановлена при возникновении ошибки в коде класса, формы или модуля. Флажок **Break on Unhandled Errors** (Прерывать при необработанных ошибках) заставит отладчик остановить выполнение только при возникновении ошибки, для которой не написан специальный обработчик.

Группа **Compile** (Компиляция) позволяет управлять компиляцией проекта:

- ❑ **Compile on Demand** (Компиляция по требованию) — если этот флажок установлен, то при запуске приложения в отладчике будет происходить последовательная компиляция кода. В этом случае код будет компилироваться непосредственно во время выполнения и будет скомпилирован только тот код, который был выполнен во время работы с запущенным приложением. В общем случае, для небольших проектов, эта опция может быть только вредна, поскольку компиляция происходит достаточно быстро.
- ❑ **Background Compile** (Фоновая компиляция) — эта опция позволяет продолжать работу над проектом во время компиляции, которая будет происходить в фоновом режиме.

## Вкладка **Docking**

На вкладке **Docking** (Прилепить) можно установить — какие из вспомогательных окон в среде разработки могут "прилипнуть" к краям окна среды разработки. Функция прилипания очень полезна для окон, с которыми вы постоянно работаете, но может быть абсолютно бесполезна для редко используемых окон, которые открываются лишь время от времени.

Выбор окон, которые "имеют право" на прилипание и, собственно, определяют внешний вид среды разработки, полностью зависит от ваших потребностей. Очень удобно всегда держать на экране окна составляющих компонентов проекта, свойств и панель инструментов "прилипшими к краям", поскольку без этих окон трудно обойтись при частом перемещении между разными файлами проекта и работе в дизайнера форм.

## Вкладка *Environment*

На вкладке **Environment** (Окружение) расположены свойства, особенно важные для разработчика, поскольку влияют на запуск среды разработки и поведение среды при запуске проекта на выполнение или отладку.

Настройка **When Visual Basic starts** (При запуске Visual Basic) влияет на загрузку среды разработки (при условии, что среда запущена не двойным щелчком по файлу проекта — в этом случае будет открыт соответствующий проект). Опция **Prompt for project** (Запросить проект) сообщает среде разработки, что она должна запросить вас о том, какой проект вы хотели бы создать. Опция **Create default project** (Создавать проект по умолчанию) заставляет среду разработки автоматически создавать тип проекта **Standard EXE** (Стандартное приложение).

Настройка **When a program starts** (При запуске программы) определяет, должен ли текущий проект быть сохранен при запуске программы в отладчике среды разработки. Опция **Save Changes** (Сохранить изменения) определяет, что все файлы проекта должны быть сохранены автоматически. Опция **Prompt to Save Changes** (Подтверждение сохранения изменений) предполагает запрос к пользователю о необходимости сохранения, а опция **Don't Save Changes** (Не сохранять изменения) сообщает среде разработки, что она может расслабиться и не делать ничего. Для начинающих программистов лучшим выбором будет **Prompt to Save Changes**, поскольку всегда можно отказаться от сохранения внесенных изменений, в то же время при сохранении файлов проекта разработчик получает уверенность, что даже сбой среды разработки во время отладки приложения не приведет к потере внесенных в код изменений.

В текстовом поле **Templates Directory** (Папка шаблонов) задается путь к директории, содержащей шаблоны файлов (форм, классов, модулей), которые предлагает создать Visual Basic при добавлении нового компонента в проект.

Какие из типов проектов среда разработки должна предлагать создать, можно задать в группе опций **Show Templates For** (Показать шаблоны для), где перечислены типы проектов, поддерживаемые Visual Basic.

## Вкладка *Advanced*

На вкладке **Advanced** (Расширенные настройки) собраны настройки, которые редко используются при работе со средой разработки Visual Basic.

**Background Project Load** (Фоновая загрузка проекта) — при этой включенной опции среда разработки Visual Basic не будет заблокирована во время загрузки проекта. По большому счету данная опция не представляет интереса, поскольку редкий разработчик приступает к работе над проек-

том, не дождавшись его полной загрузки. Кроме того, при мощностях современных компьютеров загрузка даже очень больших проектов происходит крайне быстро.

- **Notify when changing shared project items** (Уведомление об изменении разделяемых файлов проекта) — данная функция полезна когда один или несколько файлов используются более чем в одном проекте. В этом случае среда Visual Basic будет сообщать об изменениях в общих файлах.
- **SDI Development Environment** (Режим SDI среды разработки) — эта опция добавлена для обратной совместимости со средой разработки Visual Basic ранних версий, когда вместо многооконного интерфейса (MDI, multi-document interface) использовался однооконный интерфейс SDI (SDI, single-document interface). В случае SDI окна проекта размещаются на экране независимо, в отличие от MDI, когда окна проекта располагаются внутри основного окна среды разработки. Опция **SDI Development Environment** может быть полезна для программистов, ранее работавших с Borland Delphi, поскольку в среде разработки Delphi используется SDI интерфейс.

В поле **External HTML Editor** (Внешний редактор HTML) можно указать путь к приложению, которое вы хотели бы использовать для редактирования кода HTML-страниц вне среды разработки Visual Basic. Например, вы можете использовать программу Microsoft Front Page, предоставляющую больше возможностей для быстрого редактирования HTML-документов.

## Вспомогательные инструменты редактора кода

В среде разработки Visual Basic существует несколько вспомогательных инструментов для ускорения разработки: утилиты для создания заготовок кода процедур, генераторы заготовок кода классов и коллекций и, пожалуй, самый полезный инструмент для программиста на Visual Basic, встроенный в среду разработки, — **Object Browser** (Браузер объектов).

### Утилита для создания заготовок процедур

Если во время редактирования кода вам потребуется создать новую процедуру, свойство или функцию, то можно набрать код заготовки функции вручную, а можно воспользоваться утилитой для создания заготовок, для вызова которой нужно выбрать пункт **Add Procedure...** (Добавить процедуру...) в меню **Tools**. Окно утилиты выглядит, как показано на рис. 1.14.

Для создания заготовки кода процедуры необходимо ввести имя процедуры, указать модификатор доступа (**Private** или **Public**) и тип: **Sub** (Процедура), не возвращающая значение, **Function** (Функция), **Property** (Свойство) или **Event**