Глава 52

Использование редактора Visual Basic

В этой главе...

- Использование редактора Visual Basic
- Оптимальное размещение окон на экране
- Использование Project Manager для решения общих задач
- Работа с отдельным проектом при помощи Object Browser
- Маленькие хитрости и отладка программ
- Использование окна свойств для настройки объектов VBA

У плотника есть мастерская, у пекаря — кухня, а у вас... Что ж, сейчас у вас есть редактор Visual Basic. Редактор Visual Basic (Visual Basic Editor) — это служебное название встроенной среды разработки VBA, настоящей цифровой студии, в которой создают программные шедевры. Это неотъемлемая часть пяти основных приложений Office.

В настоящей главе вы научитесь использовать возможности редактора Visual Basic, который часто называют просто “редактором”. Вначале рассмотрим компоненты редактора Visual Basic, включая панели инструментов, меню и окна. А затем изучим отдельные окна и их уникальные возможности.

Работа с пользовательским интерфейсом редактора Visual Basic

Пользовательский интерфейс редактора Visual Basic соответствует стандарту Microsoft: меню, панели инструментов и комбинации клавиш выглядят и работают так же, как в любом приложении Office. Чувствуйте себя, как дома!

Приглашение на ленч с меню

Конечно, вы уже знаете, как пользоваться меню. Что вас может интересовать в такой ситуации, так это — где отыскать определенные команды меню редактора Visual Basic.

Команды редактора, в основном, организованы логически. Например, если необходимо вывести на экран какое-либо окно, проблемы с поиском нужной команды возникнуть не должно. Конечно же, она находится в меню View (Вид). Однако положение некоторых пунктов меню не столь очевидно. Следующая небольшая таблица расскажет, где искать некоторые спрятанные команды редактора:

Если вы хотите:	*Используйте команду меню:*	*Комментарии*
Задать установки всего проекта	Tools⇨Project Properties (Сервис⇨Свойства проекта)	Странное положение для команды, определяющей свойства всего проекта (я рассчитывал ее найти в меню File (Файл))
Настроить редактор Visual Basic	View⇨Toolbars⇨Customize (Вид⇨Панели инструментов⇨Настройка) или Tools⇨Options (Сервис⇨Параметры)	
Включить режим конструктора	Run⇨Design Mode (Выполнить⇨Режим конструктора)	Название команды Design Mode (режим конструктора) звучит так, что вы ожидаете увидеть макет формы. Так почему же она в меню запуска? Ответ прост: она останавливает все запущенные программы (зачем точно она нужна, я до сих пор не понял)

Настройка панелей инструментов

Если вы хорошо знакомы с Word, Excel или PowerPoint, работа панели инструментов редактора Visual Basic покажется удобной. Если же нет — нескольких полезных замечаний будет достаточно.

Основные панели инструментов

В редакторе находятся такие основные панели инструментов.

- **Debug** (Отладка). Эта панель инструментов содержит кнопки для вызова команд, которые понадобятся при отслеживании ошибок в программах.
- **Edit** (Редактирование). Кнопки этой панели удобно использовать в процессе редактирования кода программы. Они повторяют команды, содержащиеся в меню Edit (Правка).
- **Standard** (Стандартная). Это единственная панель инструментов, которая отображается на экране при первом запуске редактора. Она содержит кнопки с широким спектром функций, включая сохранение, вставку новых форм и модулей, редактирование и запуск программ.
- **UserForms** (Пользовательские формы). Данная панель используется при конструировании форм. Большинство кнопок повторяют команды меню Format (Формат) для выравнивания, упорядочения и группирования элементов управления на форме.

Жонглирование меню, панелями инструментов и кнопками

Аналогично панелям инструментов в Office, панели редактора могут находиться в одном из трех состояний: скрытом (hidden), прикрепленном (docked) или плавающем (floating). Для отображения на экране, скрытия или перемещения панелей инструментов, для их закрепления или открепления

используются методы, описанные в главе 4. Более того, меню редактора (включая контекстные меню), панели инструментов и их отдельные кнопки, можно настроить, используя стандартные методы Office, описанные в главе 5.

Комбинации клавиш

Комбинации клавиш, доступные в редакторе Visual Basic, приведены в табл. 52.1. Помимо них, можно использовать стандартные функциональные клавиши Windows для управления курсором и редактирования текста. Не забывайте также, что <Shift+F10> открывает контекстное меню для окна или другого активного элемента, действуя в точности, как при щелчке правой кнопкой мыши на элементе.

Таблица 52.1. Комбинации клавиш в редакторе Visual Basic

Действие	*Комбинация клавиш*
Отображение окон	
Отобразить окно программы для выделенной формы или элемента управления	<F7>
Отобразить форму или элемент управления, соответствующую активному окну программы	<Shift+F7>
Перейти в следующее окно программы или формы	<Ctrl+Tab>
Активизировать окно Object Browser (Просмотр объектов)	<F2>
Активизировать окно свойств	<F4>
Активизировать окно Immediate (Проверка)	<Ctrl+G>
Активизировать окно Call Stack (запущенных процедур)	<Ctrl+L>
Работа с кодом программы	
Перейти на то определение элемента, на котором находится курсор	<Shift+F12>
Отобразить диалоговые окна поиска	<Ctrl+F>
Найти далее (найти следующее вхождение искомого текста)	<F3>
Найти предыдущее	<Shift+F3>
Заменить	<Ctrl+H>
Перейти на последнюю редактированную строку	<Ctrl+Shift+F2>
Отменить последнее действие	<Ctrl+Z>
Отобразить список свойств/методов	<Ctrl+J>
Отобразить список констант	<Ctrl+Shift+J>
Отобразить краткие сведения о переменной или объекте, на котором установлен курсор	<Ctrl+I>
Вывести информацию о параметрах функции, на которой	<Ctrl+Shift+I>

установлен курсор	
Автоматическое дополнение вводимого слова	<Ctrl+"Пробел">
Работа со свойствами	
В окне свойств выполнить переход на ближайшее свойство в списке, начинающееся с заданной буквы	<Ctrl+Shift+"буква">
Запуск программ	
Запустить процедуру или пользовательскую форму в активном окне	<F5>
Приостановить выполнение программы и перейти в режим останова	<Ctrl+Break>
Отладка	
Построчное выполнение программы	<F8>
Построчное выполнение инструкций с обходом процедур	<Shift+F8>
Запуск с остановом в месте положения курсора	<Ctrl+F8>
Задание инструкции, следующей на выполнение	<Ctrl+F9>
Запуск процедуры обработки ошибки или возврат ошибки в вызывающую процедуру	<Alt+F5>
Шаг с заходом в процедуру обработки ошибки или возврат ошибки в вызывающую процедуру	<Alt+F8>
Переключение между курсором и точкой останова в строке программы	<F9>
Снятие всех точек останова	<Ctrl+Shift+F9>
Добавление контрольного выражения в месте положения курсора	<Shift+F9>

Управление окнами

Если у вас небольшой экран монитора, придется тратить довольно много времени на размещение окон в редакторе Visual Basic. Эти окна не просто занимают место на экране — каждое из них обладает важным качеством, помогающим в работе. Вся проблема в том, что держать все окна открытыми непрактично — просто не останется места для программы VBA и форм.

Между прочим, в этой главе будет идти речь именно о механизме работы с окнами, а не о функциях самих окон. Позже об отдельных окнах будет рассказано в следующих главах.

Какие окна любят одиночество, а какие — компанию?

Что касается окон редактора Visual Basic, самым важным фактом является то, что вы можете открыть столько окон Code и UserForm, сколько нужно, но всех остальных — только по одному окну каждого типа. Возможно, данный факт совершенно очевиден, но мне потребовалось некоторое время, чтобы понять, как все это работает.

Несколько окон Code и UserForm оказываются полезными, поскольку вы, скорее всего, захотите создать несколько форм и модулей VBA, а имея окно для каждого модуля и каждой формы, получаете быстрый доступ к каждому.

Некоторые окна — Properties и Locals (Локальные значения) — меняют содержимое автоматически при переходе из одного окна Code и UserForm к другому. Другие (окно Object Browser, Project Explorer и Immediate) относятся ко всему, что вы делаете в редакторе, поэтому достаточно и одного (каждого типа).

Отображение и сокрытие окон

Большинство окон редактора имеют комбинации клавиш для быстрого вызова с клавиатуры. В табл. 52.1 содержится список необходимых комбинаций клавиш. Если их сложно запомнить, отобразить любое окно можно при помощи меню View.

К сожалению, комбинации клавиш не являются переключателями. Если окно уже отображено, повторное нажатие клавиш не уберет его с экрана. Чтобы отправить окно отдохнуть, нажмите кнопку закрытия (самая правая кнопка в строке заголовка).

Для отображение открытого окна, которое спрятано под другими окнами, выберите его имя в меню Windows (Окно). Правда, в этом меню перечислены только незакрепленные окна.

Как во многих приложениях Windows, для перехода от одного окна к следующему существуют две комбинации клавиш: <Ctrl+Tab> и <Ctrl+F6>. Однако в редакторе Visual Basic переключаться с помощью этих клавиш можно только между незакрепленными окнами. Что же такое незакрепленное окно? Это тема следующего раздела.

Закрепленные и плавающие окна

Как и панели инструментов, большинство окон в редакторе являются *закрепляемыми* — их можно прикрепить вдоль любой из четырех сторон рабочего пространства, где они не будут перекрывать другие окна. Конечно, закрепление окон делает рабочее пространство меньше. На рис. 52.1 показано, как выглядит редактор, когда все видимые окна закреплены.

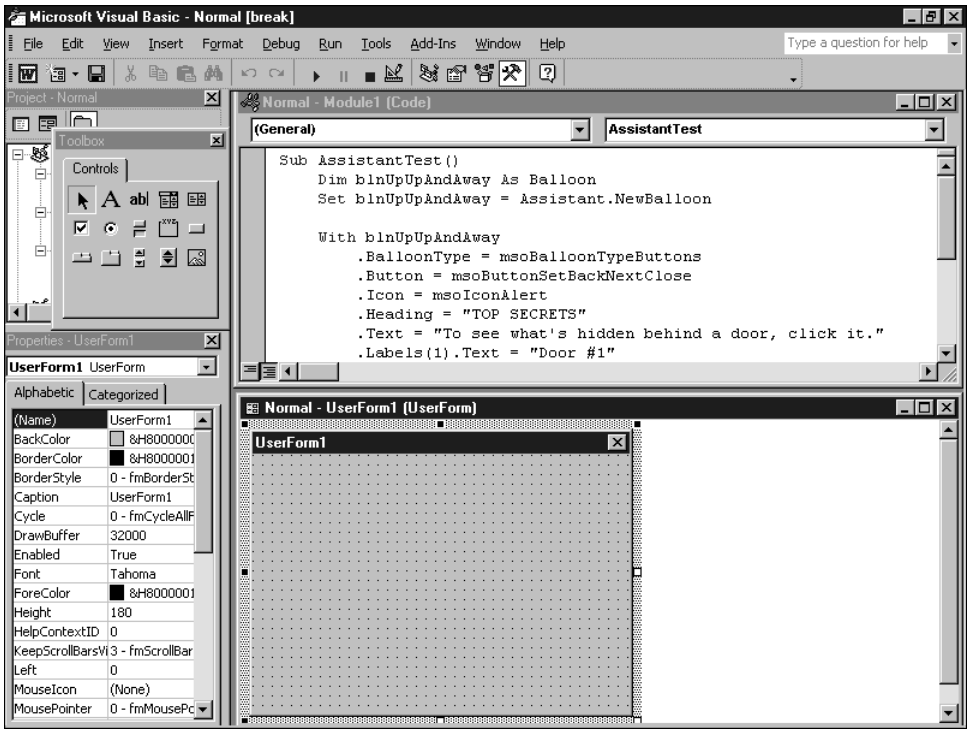


Рис. 52.1. При закреплении окон по краям окна редактора Visual Basic уменьшается пространство, отведенное для редактирования и конструирования форм

И наоборот, можно отправить окна свободно “плавать” по экрану. Плавающие окна оставляют больше свободного пространства для написания программ и разработки форм, но часто закрывают другие окна. При наличии большого монитора размеры окна редактора Visual Basic можно уменьшить, для того чтобы некоторые плавающие окна разместить за его пределами.

Закрепляемость

Некоторые окна не могут быть прикреплены, например окно Code и UserForm. К тому же, можно закрепить любое другое окно. Чтобы задать статус закрепленности для всех типов окон (конечно, исключая окна Code и UserForm), выполните команду Tools⇨Options (Сервис⇨Параметры) и выберите вкладку Docking (Закрепление).

Сделать закрепляемое окно незакрепляемым или наоборот можно, щелкнув правой кнопкой мыши на основной части окна и выбрав команду Dockable в контекстном меню.

Еще немного о закрепляемости

Существуют еще некоторые неочевидные истины о прикрепленных окнах.

- Окна программы, формы или любые другие плавающие окна могут быть развернуты, свернуты или восстановлены (термин от Microsoft, означающий “возвращение к размеру, определенному пользователем”). Плавающие окна имеют стандартные кнопки для этих функций на правой стороне строки заголовка.
- Определить, является ли окно прикрепленным, можно по его заголовку. Прикрепленные окна содержат только кнопку закрытия. У плавающих окон есть кнопки “свернуть” и “развернуть” (или “восстановить”).

- В процессе разворачивания плавающего окна оно займет все свободное пространство, незанятое прикрепленными окнами.
- Для перемещения плавающего окна к краю рабочей области без его прикрепления сначала сделайте его незакрепляемым.
- Все открытые незакрепленные окна доступны через последовательное нажатие комбинации клавиш <Ctrl+Tab> (они также перечислены в меню Window).

Сохранение макета экрана

Редактор Visual Basic автоматически сохраняет расположение содержимого экрана, с которым вы работали перед выходом. Размещение окон, меню и панелей инструментов переносится с одного сеанса работы на следующий. Расположение остается тем же даже при переходе с одного приложения Office в другое. Правда, для этого нужно закрыть первое приложение перед запуском следующего.

Использование Project Explorer для управления проектами

В VBA *проектом* называют все программы и формы, которые относятся к одному документу, вместе с самим документом. В редакторе Visual Basic окно Project Explorer используется для отображения общего вида всех проектов, открытых на данный момент в приложении, и, что более важно, для быстрого перемещения в окно программы или формы, с которыми вы хотите работать.

Для сохранения проекта, с которым вы работаете в данный момент и который является выделенным в Project Explorer (как описано позже в этом разделе), щелкните на кнопке Save (Сохранить) на стандартной панели инструментов редактора Visual Basic. Для сохранения документа и относящегося к нему кода нет необходимости возвращаться в приложение VBA.

Открытие окна Project Explorer

Окно Project Explorer появляется на экране при первом запуске редактора Visual Basic. Если его нет, используйте для его отображения один из следующих способов.

- Нажмите <Ctrl+R>.
- Щелкните на кнопке Project Explorer стандартной панели инструментов.
- Выберите команду View⇨Project Explorer (Вид⇨Проводник проекта).

Структура проводника

Если вы работали с проводником Windows для управления дисками и файлами (ой, *документами*), то с Project Explorer будете себя чувствовать свободно. Project Explorer отображает древовидную иерархическую структуру открытых проектов (рис. 52.2).

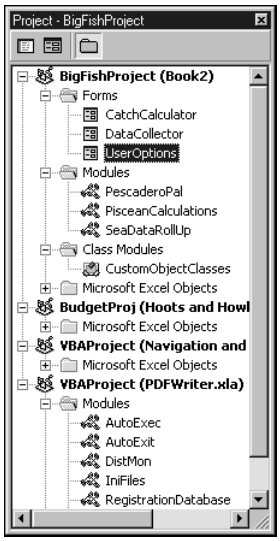


Рис. 52.2. Это Project Explorer в Excel, отображающий несколько открытых одновременно проектов

Вверху иерархии находятся отдельные проекты. Они размещены ближе всех к левому краю окна Project Explorer.

Для каждого документа, открытого в приложении, автоматически создается проект. Это происходит даже если для него не создается никаких форм, никаких программ. По умолчанию проект имеет тоже имя, что и документ, но его можно изменить в диалоговом окне Project Explorer или окне Properties (см. “Переименование проекта или модуля” далее в этой главе).

Следующий уровень иерархии занимают группы связанных объектов: формы, модули программы, *ссылки* на библиотеки других объектов и объекты из основного приложения. Обычно Project Explorer отображает папку для каждой из этих групп. Внизу иерархии находятся отдельные объекты.

Навигация в окне Project Explorer

Project Explorer предоставляет возможность самого быстрого поиска и активизации модулей, форм и других объектов, с которыми вы хотите работать. Так же, как проводник Windows, рядом с каждым элементом или *вершиной* дерева Project Explorer отображает значок. Когда является видимым только проект, а его содержимое спрятано, значок (*индикатор вложенности*) содержит знак “плюс”. Если вершины более низких уровней иерархии видимы, индикатор вложенности отображает знак “минус”.

Для работы с Project Explorer используйте следующую технику.

- **Чтобы развернуть вершину и отобразить элементы, находящиеся ниже по иерархии**, щелкните на индикаторе вложенности, который содержит знак “плюс” или выделите вершину и нажмите клавишу со стрелкой вправо.
- **Чтобы свернуть вершину и скрыть подчиненные элементы**, щелкните на индикаторе вложенности, который содержит знак “минус” или выделите вершину и нажмите клавишу со стрелкой влево.
- **Для открытия окна формы или программы, соответствующей в Project Explorer элементу модуля, формы или класса модулей** дважды щелкните на элементе или выделите его название в списке и нажмите <Enter>.
- **Для активизации окна программы формы** выделите название формы и нажмите <Shift+Enter>.

- **Для использование кнопок, которые находятся вверху окна Project Explorer** поместите указатель мыши над одной из них под названием окна. Они работают следующим образом:
  - кнопка View Code отображает окно программы (Code) для выделенного элемента;
  - кнопка View Object отображает сам выделенный элемент. Если выделенный элемент — форма, вы увидите ее в окне формы. Если элемент — документ, вы перейдете назад в основное приложение VBA, а данный документ останется активным;
  - кнопка переключения папок Toggle Folders включает или выключает средний уровень иерархии проекта. Обычно Project Explorer выделяет такие объекты, как формы, программные модули и документы в отдельные папки. При щелчке на данной кнопке эти папки исчезают, а отдельные объекты отображаются в каждом проекте в алфавитном порядке. Щелкните на кнопке еще раз, и папки появятся опять.

Использование контекстного меню Project Explorer

Контекстное меню Project Explorer, показанное на рис. 52.3, сделает вас еще ближе к элементам проекта. Данные команды доступны через меню, но если при работе вы используете Project Explorer, самый быстрый доступ обеспечивает контекстное меню.

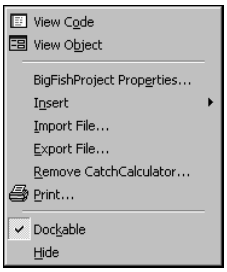


Рис. 52.3. Контекстное меню Project Explorer

Задание свойств проекта

Диалоговое окно Project Properties (Свойства проекта) позволяет изменять имя проекта, добавлять краткое описание, присоединять пользовательский файл справки и защищать проект от несанкционированного доступа и внесения нежелательных изменений. Хотя извлечь пользу из большинства этих параметров может только продвинутый VBA-разработчик.

Но в любом случае, вы будете знать, что такое диалоговое окно существует. Для его отображения выберите команду Tools⇨*Имя проекта* Properties (Сервис⇨Свойства *Название проекта*) или щелкните правой кнопкой мыши в области окна Project Explorer и выберите команду *Имя проекта* Properties из контекстного меню (рис. 52.4).

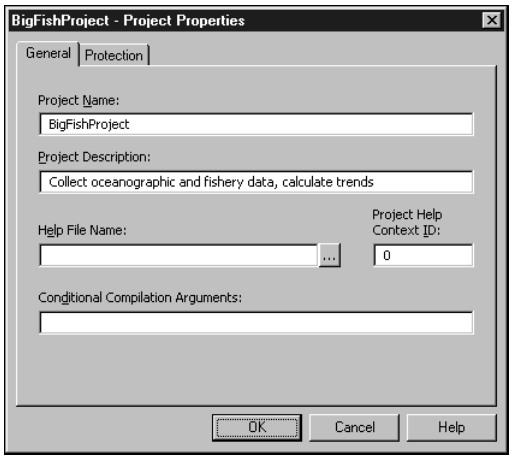


Рис. 52.4. Вкладка General (Общие) диалогового окна свойств проекта Project Properties

Переименование проекта

Project Explorer и Object Browser содержат список проектов в алфавитном порядке. Это значит, что переместить проект в любом из этих окон можно, изменив его имя. (Конечно, можно изменить имя и по чисто художественным соображениям.)

Самый простой способ изменить имя проекта, сгенерированное самим VBA, — через окно свойств. Когда проект выделен в окне Project Explorer, окно Properties отображает его имя, которое можно редактировать. Также имя проекта можно редактировать во вкладке General диалогового окна Project Properties, если оно по каким-то причинам открыто.

Защита проекта

Если существует вероятность того, что кто-нибудь сядет за ваш компьютер и внесет неразбериху в программы, можете защитить свои VBA-проекты паролем. Для возведения такой защитной ограды отобразите диалоговое окно Project Properties и перейдите на вкладку Protection (Защита) (рис. 52.5).

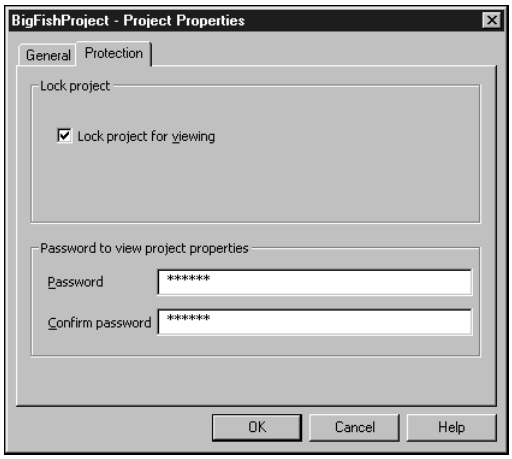


Рис. 52.5. Вкладка Protection диалогового окна Project Properties

Настройки во вкладке Protection работают следующим образом.

- **Lock project for viewnig** (Защита проекта от просмотра). Установка флажка рядом с этим полем блокирует любые действия того, кто не знает пароля. Без пароля невозможно даже просмотреть проект. Если флажок не установлен, пароль не допустит открытия

диалогового окна Project Properties и, следовательно, предотвратит блокировку вашей работы от вас самого.

- **Password** (Пароль). Введите пароль, который вы выбрали для вашего проекта.
- **Confirm password** (Подтвердите пароль). Введите его еще раз — это застрахует от неправильного ввода пароля.

Если вы не обладаете феноменальной памятью, запишите пароль и положите там, где никто не сможет найти. VBA хранит проект достаточно надежно. Забудете пароль — и придется заново создавать проект "с нуля".

Чтобы сохранить свое время, не создавать себе проблем и не заниматься затем самобичеванием, сохраните копию проекта в надежном месте перед тем, как защитить паролем.

Использование Object Browser

Несмотря на отличия во внешнем виде и наличие множества ненужных свойств, окно Object Browser имеет много общего с Project Manager. Как и Project Manager, оно позволяет быстро перемещаться в иерархии объектов, доступных для программ VBA.

Помимо внешних различий, между ними существует разница, которая заключается в том, что Object Browser отображает только один проект в определенный момент, но предоставляет доступ ко *всем* объектам, которые могут быть использованы этим проектом, а не только к тем, что принадлежат самому проекту. Другими словами, помимо программных модулей и форм проекта, можно просмотреть объекты, предоставляемые приложением, самим VBA и другими открытыми библиотеками объектов.

Другое большое преимущество Object Browser — то, что с его помощью можно просмотреть все процедуры, методы, события, свойства и прочее из *любой* из этих библиотек объектов.

Запуск Object Browser

Клавиша <F2> — команда для быстрого отображения окна Object Browser (рис. 52.6). А можно щелкнуть на кнопке Object Browser стандартной панели инструментов или выбрать команду View⇨Object Browser.

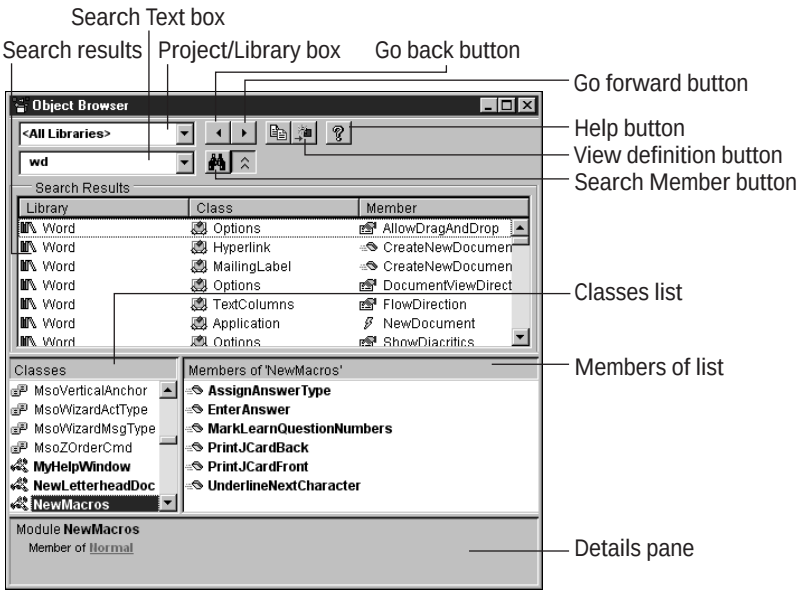


Рис. 52.6. Окно просмотра объектов — Object Browser

Просмотр объектов

Перед тем, как начать серьезную работу с Object Browser, убедитесь, что в нем находится нужный проект. Все, что необходимо сделать — выбрать его в Project Explorer.

Приняв вышеописанную меру предосторожности, щелкните на поле со списком Project/Library (Проект/Библиотека) в левом верхнем углу, где перечислены библиотеки, доступные для использования в выбранном проекте. (Проект должен быть в этом списке, если его нет — вернитесь в Project Explorer и убедитесь, что выбран именно требуемый проект.) Выбрав <All libraries> (Все библиотеки) в списке Project/Library, можно просмотреть все объекты, доступные в проекте (это объекты из всех библиотек, на которые есть ссылки в проекте). Чтобы использовать отдельную библиотеку, просто выберите ее в списке. Для просмотра объектов из других библиотек добавьте на них ссылку, используя диалоговое окно Tools⇨References (Сервис⇨Ссылки).

Разделы окна просмотра объектов

Список классов Classes находится в левом разделе средней части окна Object Browser. В нем содержатся все объекты, семейства, модули, формы и константы, доступные в выбранной библиотеке или проекте. При выборе любого элемента из этого списка в правом разделе отображаются члены данного класса в списке Members Of. Членами могут быть процедуры, методы, свойства, события или отдельные константы.

Хотите узнать, что включено в программу VBA? Просто посмотрите на элементы списков (левого или правого), которые выделены жирным шрифтом.

В нижнем разделе окна Object Browser, разделе Details (Подробности), находится информация о выделенном объекте. Здесь указывается тип элемента и объект, которому принадлежит элемент (если эта информация невидима, потяните верхний разделитель вверх или используйте полосу прокрутки, находящуюся справа). По щелчку на имени объекта, которому принадлежит элемент, в окне отобразится сам объект.

Object Browser достаточно удобно размещается на экране. Размеры любого раздела (а также столбцов раздела поиска Search Results, описанного ниже) можно изменить, перетаскивая разделители на нужное место.

Перемещение по объектам

В основном, Object Browser работает подобно обозревателю Web. Как и обозреватель Web, он позволяет осуществлять переходы к связанным элементам по гиперссылкам. Щелкая на объекте, выделенном подчеркиванием в разделе Details, вы увидите его в окне Object Browser.

Так же, как и Web-обозреватель, Object Browser позволяет вернуться обратно на нужное количество шагов. Чтобы вернуться к объектам, которые вы просматривали ранее, щелкните на кнопке Go Back (Назад) вверху окна просмотра объектов. Конечно, есть также кнопка Go Forward (Вперед), которая позволяет снова изменить направление и вернуться к объекту, изучаемому в момент первого нажатия кнопки Go Back.

Доступ к коду программы

Если выбранный в Object Browser элемент — модуль или процедура вашей программы, то при нажатии <Enter> откроется окно программы. (Любители мыши для этой же цели могут щелкнуть на кнопке Show Definition (Показать определение).)

Справка в Object Browser

Предполагается, что при нажатии клавиши <F1>, щелчке на кнопке Help (Справка) или выборе команды Help из контекстного меню откроется окно справки по объекту, методу, свойству или событию, выделенному в Object Browser. Обычно так и бывает. Произвольное перемещение Object Browser и вызов справки по интересующим элементам — отличный и (относительно) безболезненный способ изучения VBA для любителей творческого подхода.

Поиск компонентов

Случается, что вы не знаете, какой именно модуль содержит определенную процедуру или какой объект имеет определенный метод или событие. Чтобы не прочесывать все модули в Project Explorer или искать в справочной системе, используйте Object Browser для нахождения этого элемента. Вот как это делается.

1. В списке Project/Library выберите <All libraries> для поиска во всем проекте или задайте имя нужной библиотеки.

Выбрав не ту библиотеку, можно найти кое-что, но, скорее всего, не то, что нужно.

2. Введите искомый текст в поле Search Text (Искать).

   Вы можете повторить любой из последних четырех поисков, выбрав введенный ранее текст из раскрывающегося списка поля Search Text.

3. Нажмите <Enter> или щелкните на кнопке Search (Найти) с пиктограммой бинокля.

   После обязательного поиска по всему жесткому диску подходящие элементы отобразятся в новом разделе Search Results (Результаты поиска). Можно изменить размер любого столбца, если он мал для отображаемого элемента.

4. Для закрытия раздела Search Results щелкните на той кнопке с двумя стрелками, направленными вверх, которая находится рядом с кнопкой Search.

Щелкните на той же кнопке (которая теперь имеет вид двух стрелок, направленных вниз), чтобы снова отобразить раздел Search Results.

Использование в программе элементов, отображенных в окне Object Browser

Object Browser может сделать нечто большее, чем удовлетворить любопытство относительно объектов проекта. Он может оказать скромную практическую помощь при написании кода программы. После того, как вы нашли элемент, который хотите использовать в программе, нажмите <Ctrl+C> или щелкните на кнопке Copy to Clipboard (Скопировать в буфер обмена) с целью поместить синтаксическую конструкцию в буфер обмена. Затем, переключившись в соответствующее окно программы, вставьте содержимое буфера в нужное место. Такая техника обеспечит правильное написание, что обезопасит вас от неожиданных ошибок при запуске программы.

Секреты программирования

Окна Code — сердце редактора Visual Basic. Здесь создаются конструкции VBA, которые и выполняют работу. В данной главе суть этих конструкций отступает на задний план, а внимание обращено на то, как извлечь максимум пользы из окна Code при написании программы.

Открытие окна Code

Редактор Visual Basic предоставляет множество способов открытия окна Code для существующего модуля, модуля класса или формы. Вначале найдите и выделите элемент в Project Explorer. Далее одно из следующих действий приведет к открытию соответствующего окна программы:

- нажатие клавиши <F7>;
- щелчок на кнопке View Code, находящейся вверху окна Project Explorer;
- щелчок правой кнопкой мыши на объекте и выбор команды View Code в контекстном меню;
- выбор команды View⇨Code;
- для модулей: двойной щелчок на элементе или простое нажатие клавиши <Enter> (для форм такой способ не работает).

Если вы находитесь в окне формы, нажмите <F7> или выберите команду View⇨Code, чтобы отобразить окно программы данной формы.

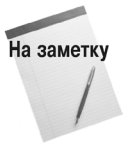
Создание нового окна программы

Вставка нового модуля в проект сопровождается для него автоматическим открытием нового окна Code. При создании новой формы для нее также автоматически создается окно Code (но вы его не увидите, пока не используете один из вышеперечисленных способов отображения).

Процесс создания нового модуля описан в главе 53.

Работа с текстом программы

По своей сути, окно Code — простой текстовый редактор, но он включает в себя набор специальных возможностей, разработанных специально для написания программ. Вы пишете инструкции VBA точно так же, как в текстовом редакторе, используя те же клавиши редактирования и управления курсором, которые являются стандартными для Windows (нажатие клавиши <Home> перемещает курсор в начало строки, <Ctrl+Home> — вверх окна и т.д.). Текст можно выделить мышью либо перемещая курсор и удерживая нажатой клавишу <Shift>.



Как любой уважающий себя текстовый редактор конца XX века, окно Code поддерживает технику перетаскивания.

После того, как текст, с которым вы планируете работать, выделен, можно:

- переместить его, перетащив и отпустив в нужном месте;
- скопировать, удерживая нажатой клавишу <Ctrl> при перетаскивании и отпускании.

Текст можно перетащить на новое место в том же окне программы, в другое окно Code, в окно проверки или окно контрольных значений (о последнем рассказывается в главе 55). При перетаскивании в другое окно программы оба окна нужно расположить так, чтобы и начальное положение текста, и место вставки были видимыми.

В окне программы также можно отменить предыдущие изменения, внесенные в программу. Каждый раз при нажатии комбинации клавиш <Ctrl+Z> (или при использовании команды Undo (Отменить) из меню Edit (Правка)) отменяется одна операция. Меню Edit содержит команду Redo (Повторить) для отмены изменений, внесенных командой Undo, но ей не назначена комбинация клавиш.

Идеальный наставник

Подобно безукоризненному слуге, о котором вы всегда мечтали, редактор Visual Basic постоянно, но ненавязчиво проверяет и корректирует работу одним из следующих способов.

- Если в одной строке сделан отступ, такой же отступ автоматически будет сделан на следующей (эту функцию можно отключить через диалоговое окно Tools⇨Options, во вкладке Editor (Редактор), сняв флажок Auto Indent (Автоматический отступ)).
- Если редактор распознает ключевое слово VBA, он автоматически заменяет его первую букву на заглавную, в соответствии с соглашением, принятым в VBA (например, при создании конструкции `if...then...else`, редактор заменит ее на `If...Then...Else`). К тому же, ключевые слова автоматически выделяются синим цветом, так что они отличаются от других (схему выделения цветом можно изменить во вкладке Editor Format (Формат) диалогового окна Tools⇨Options).
- В окне Code при создании новой процедуры вы печатаете слово `Sub` или `Function`, за которым следуют скобки и список аргументов, если необходимо. После ввода данной строки редактор Visual Basic автоматически создает необходимую инструкцию `End Sub` или `End Function`, соответственно.
- Самым важным является то, что при написании очевидно неполной (или не соответствующей синтаксису) инструкции VBA появляется предупреждающее сообщение от редактора (рис. 52.7), что дает возможность исправить ошибку, пока вы еще помните, что должна выполнять написанная инструкция. Даже если вы пропустите предупреждение на данный момент, редактор будет отображать оператор красным цветом, напоминая об ошибке.

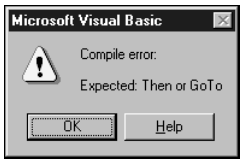


Рис. 52.7. Когда строка программы VBA не соответствует правильному синтаксису, на экран выводится сообщение с предупреждением, подобным этому

Навигация в окне программы

При создании программы любой сложности окно Code будет содержать страницы и страницы кода. В таком элегантном окружении беспорядочное прокручивание вверх и вниз является слишком примитивным способом навигации по программе.

Воспользуйтесь преимуществом расположенных вверху окна программы полей с раскрывающимся списком. Они переместят вас прямо к процедуре, которую необходимо просмотреть или отредактировать, как описано ниже.

- **Поле Object** (Объект). Поле со списком, расположенное слева. В окне модуля оно содержит лишь "(General (Общие))", и вам не о чем беспокоиться. Но при работе с окном Code для формы поле позволяет выбрать отдельный элемент управления формы (или саму форму). Окно Code отобразит стандартную процедуру для данного объекта.
- **Поле Procedures/Events** (Процедуры/События). При выборе раздела Declarations (Объявления) (для всего окна) или определенной процедуры отображается код выбранного элемента. При работе с окном программы формы, список этого поля содержит лишь события, доступные для объекта, находящегося в поле Object. При выборе события окно Code отображает соответствующую данному событию процедуру.

Использование закладок

Три часа пополудни. Ваши веки тяжелые, а пальцы еле шевелятся. Но сроки поджимают, и вы продолжаете печатать строку за строкой. Вдруг на вас нисходит вдохновение, и вы уже знаете, как решить ту важную задачу, над которой бились вчерашним утром.

Перед тем, как перейти в другой модуль, чтобы реализовать гениальную идею, установите *закладку* в месте нынешней работы. А когда придет время вернуться к этому модулю, вы сможете это сделать в мгновение ока. Чтобы установить на определенной строке программы (на самом деле рядом со строкой — рис. 52.8) закладку, щелкните на кнопке Toggle Bookmark (Включить закладку).

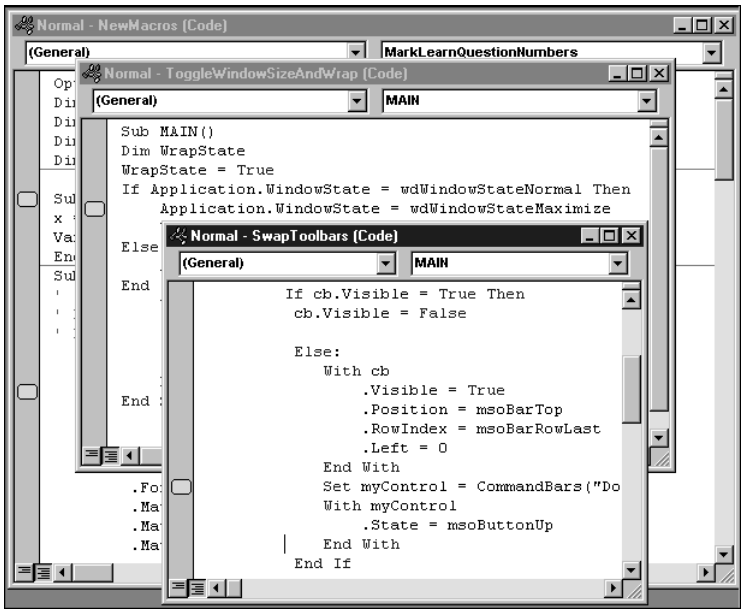


Рис. 52.8. Светлые овалы вдоль левого края окна программы представляют собой закладки. На цветном мониторе они голубого цвета

Все кнопки, связанные с закладкой, появляются на панели инструментов редактирования. Чтобы получить к ним доступ, нужно отобразить эту панель инструментов. С другой стороны, можно использовать команду меню Edit⇨Bookmarks⇨Toggle Bookmark (Правка⇨Закладки⇨Включить закладку), что гораздо дольше. (В Microsoft поступили жестоко, не привязав комбинации клавиш к этим командам.) Вместо того, чтобы отображать панель инструментов редактирования для доступа к кнопке Toggle Bookmark, можно щелкнуть правой кнопкой мыши в пределах окна программы и выбрать в контекстном меню команду Toggle⇨Bookmark (Включить⇨Закладку). Можно установить нужное вам число закладок.

Конечно, сама по себе закладка принесет мало пользы. Используйте кнопки Next Bookmark (Следующая закладка) и Previous Bookmark (предыдущая закладка) для последовательного перехода от одной закладки к следующей, пока не найдете нужную.

Чтобы удалить одну закладку, поместите курсор на содержащую ее строку и снова щелкните на кнопке Toogle Bookmark. Если накопилось столько закладок, что кнопки Next Bookmark и Previous Bookmark работают так же медленно, как полоса прокрутки, можно снять все закладки, щелкнув на кнопке Clear All Bookmarks (Снять все закладки).

Разделение окна

Любое окно Code можно разделить на две отдельные части (рис. 52.9), что позволяет видеть разные части модуля одновременно и таким образом облегчает операции удаления и вставки из буфера обмена в разных частях программы.

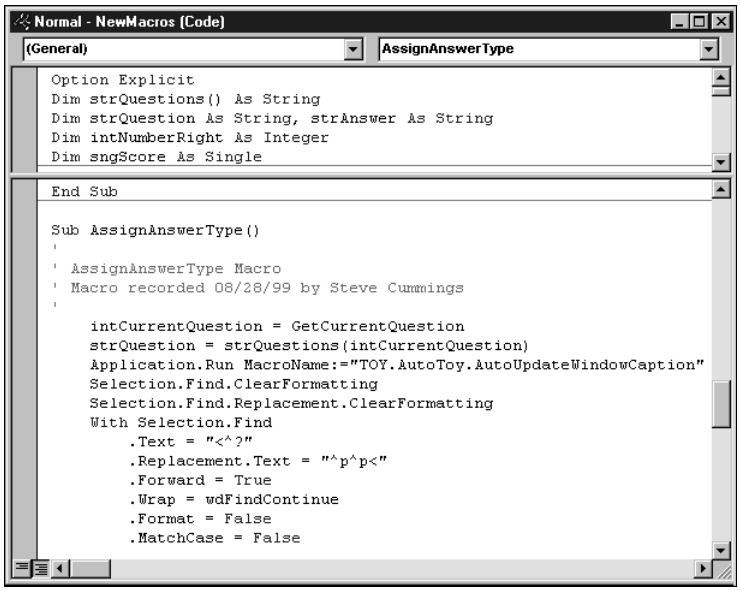


Рис. 52.9. Это окно программы разделено на две части

Чтобы разделить окно программы, используйте полосу разделения — небольшой серый прямоугольник над верхней правой стрелкой прокрутки. Потяните разделитель вниз так, чтобы установить приемлемые размеры частей окна. Для объединения этих частей удалите разделитель двойным щелчком мыши или перетащите его обратно вверх.

Дополнительные возможности окна Code

Зачем писать программу, если кто-то другой может сделать это за вас? Редактор Visual Basic может побаловать вас, вводя эти “чокнутые” выражения VBA. Такой подход позволяет не только сэкономить время при печатании, но и застрахует от ошибок при написании.

Вот несколько связанных возможностей, любезно предоставленных редактором. Все они находятся в меню Edit:

- List Properties and Methods — список свойств и методов;
- List Constants — список констант;
- Complete Word — завершение слова.

Средство List Properties and Methods

Из всех возможностей List Properties and Methods (список свойств и методов) — наиболее полезная. Она действует следующим образом. Чтобы заставить объект VBA работать, нужно изменить одно из его свойств или активизировать один из его методов. Для задания свойства объекта или активизирования его метода нужно написать имя объекта, поставить точку, затем ввести имя свойства или метода, наподобие этого:

```
ActiveWindow.Seletion.Group
```

Работа с объектами, их свойствами и методами полностью описана в главе 54.

Имея список свойств и методов, достаточно написать имя объекта и точку, после чего появится окно с небольшим списком возможных свойств и методов данного объекта. На рис. 52.10 показан

такой список. Чтобы найти нужное свойство или метод, можно использовать полосу прокрутки или ввести одну-две первых буквы элемента. Когда нужный элемент выделен, нажмите клавишу <Tab>, чтобы вставить его в документ.

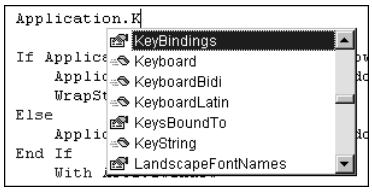


Рис. 52.10. Средство List Properties and Methods предлагает неоценимую помощь при вводе свойств и методов в программе

List Properties and Methods предоставляет ту же услугу при объявлении переменной (задании ее типа). Когда вы начинаете вводить инструкцию, подобную следующей

```
Dim Muesli As CerealObject
```

список всех возможных типов данных и объектов появляется сразу после ввода слова `As`. Как и ранее, просто выберите тип из списка, прокручивая его или введя первые одну-две буквы, и нажмите <Tab>.

Подробнее об объявлении переменных см. главу 53.

Средство List Properties and Methods работает не только со встроенными объектами VBA, но также с вашими собственными объектными переменными, *однако лишь с теми, которые были явно объявлены таковыми*. Это, в общем, веская причина для явного объявления переменных.

Еще один момент. Список свойств и методов всплывает автоматически, только если установлен соответствующий параметр в диалоговом окне Options (Параметры). Если список не появляется, выберите команду Tools⇨Options и установите флажок Auto List Members. Список можно вывести вручную одноименной командой в меню Edit или с помощью комбинации клавиш <Ctrl+J>.

Автоматический вывод списка констант

Средство List Constants работает аналогично List Properties and Methods, за исключением того, что оно содержит имена определенных ранее констант для свойства, с которым вы работаете. Этот список открывается, как только в операторе введен знак “=”:

```
Muesli.Crispness = Mushy
```

Комбинация клавиш такая же, как и раньше: <Ctrl+J>.

Автоматическое дополнение слова

Средство Complete Word редактора Visual Basic может ввести практически любое слово. Для его активизации нажмите <Ctrl+"Пробел">. Данная комбинация клавиш выводит на экран список, который выглядит и работает так же, как List Properties and Methods. Отличие заключается в том, что список содержит практически все элементы, которые могут понадобиться: объекты, функции, процедуры, константы, методы, свойства и ваши собственные переменные. Исключение составляют лишь ключевые слова для встроенных типов данных, таких как `Integer` и `Variant`.

Ввод одной-двух букв перед нажатием <Ctrl+"Пробел"> ограничит список элементами, начинающимися с этих букв. Лучше всего, правда, ввести достаточное количество букв для однозначного определения элемента. В таком случае редактор вставит его сразу после нажатия комбинации <Ctrl+"Пробел"> — список даже не появится на экране.

Работа с аргументами

Многие процедуры VBA при выполнении требуют одного или более *аргументов*. Такие процедуры основывают свои вычисления (или другие действия) на информации, предоставляемой аргументами. Многие методы объектов требуют аргументов, подобно процедурам.

Редактор Visual Basic не введет аргументы вместо вас — он не знает, какие конкретно значения должны иметь аргументы. Однако он выведет небольшое окошко, содержащее подобие шпаргалки, которая подскажет, какие аргументы требует функция, их тип, а также определит, какие из них являются необязательными (рис. 52.11).

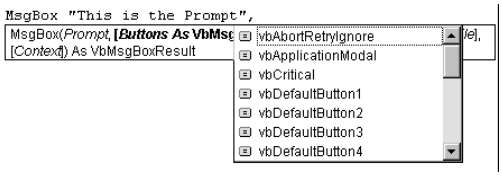


Рис. 52.11. Окно подсказки Quick Info

Окно Parameter Info как дополнение окна Quick Info

Когда вы занимаетесь серьезным программированием, используя сложные конструкции, окно Parameter Info дополняет всплывающую подсказку Quick Info. Часто бывает, что одна функция определяет значения аргумента другой функции, как в следующем примере:

```
MsgBox(Str(AnIntegerVariable))
```

Здесь функция `Str` преобразовывает целое значение переменной `AnIntegerVariable` в текстовую строку, используемую впоследствии как аргумент функции `MsgBox` (она становится справкой, которую вы видите, когда VBA отображает сообщение на экране).

Поскольку допускается создавать вложенные функции, окна Quick Info может оказаться недостаточно. Оно всегда содержит список аргументов для функции, в которой находится точка вставки. Дополнительный помощник, окно Parameter Info, содержит список аргументов для самой внешней функции. Комбинация клавиш для инструмента Parameter Info — <Ctrl+Shift+I>.

Это инструмент Quick Info. Если установлен флажок Auto Quick Info (Всплывающая подсказка) в диалогом окне Tools⇨Options, всплывающая подсказка будет появляться автоматически при вводе имени функции, метода или процедуры, требующих аргументов. Если вы отказались от автоматического отображения этого окна, по-прежнему можно воспользоваться комбинацией клавиш <Ctrl+I>.

Некоторые функции и процедуры требуют множества аргументов. Чтобы помочь отследить место положения точки вставки в списке аргументов, окно Quick Info отобразит вводимый аргумент полужирным шрифтом. После ввода запятой, которая определяет конец аргумента, полужирным шрифтом в окне Quick Info выделяется следующий аргумент.

Использование окна Properties

Окно Properties (Свойства) позволяет просматривать и редактировать свойства любого существующего объекта (проекта, модуля, формы или элемента управления), который является активным на данный момент в редакторе Visual Basic. На рис. 52.12, например, область заголовка содержит имя активного объекта, который также является выделенным в Project Explorer. (Если выбранный объект является отдельным элементом управления формы, список Project Explorer содержит только имя формы.)

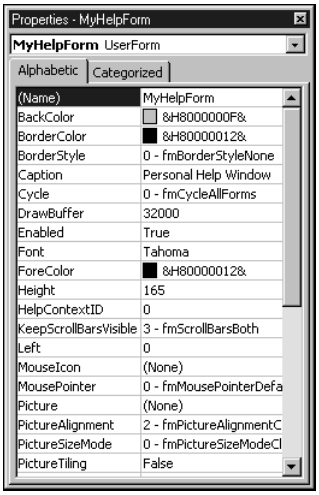


Рис. 52.12. Окно свойств для формы

Когда выделен проект или модуль, единственным свойством, содержащимся в окне свойств, является имя проекта. Для таких объектов, как форма или элемент управления, однако, окно свойств предоставляет доступ ко множеству свойств, управляющих видом и поведением объекта.

Подробнее о работе окна Properties см. главу 56.

Чаще всего окно Properties используется при разработке форм. Поэтому далее вкратце описано, как его отобразить и применить для переименования проектов и модулей.

Открытие окна свойств

Для отображения окна Properties выполните одно из следующих действий:

- нажмите клавишу <F4>;
- на стандартной панели инструментов щелкните на кнопке Properties;
- выберите команду View⇨Properties Window (Вид⇨Окно свойств).

Когда окно Properties открыто, свойства элемента можно отобразить, переключившись в окно программы или формы либо выбрав элемент в Project Explorer.

Переименование проекта или модуля

Каждый проект или модуль имеет лишь одно свойство — имя. Но для изменения этого единственного свойства все равно потребуется использовать окно Properties. Чтобы переименовать проект или модуль, выполните следующие действия.

1. Выберите проект или модуль в Project Explorer.
2. Отобразите окно свойств, если оно не отображено.
3. Активизируйте окно Properties, если оно не активно. (Щелкните на нем или нажмите <F4>.)
4. Введите новое имя.

   Поскольку данное свойство единственное, нет необходимости щелкать в строке свойства `Name` перед началом ввода.

Окна отладки

Для отладки программ разработано четыре окна: Immediate (Проверка), Locals (Локальные значения), Watch (Контрольные значения) и Call Stack (Запущенные процедуры).

Чтобы разобраться, какое из окон использовать для поиска и исправления ошибок в программе, обратитесь к главе 55.