

# самоучитель

Владимир Волков


## ПРОГРАММИРОВАНИЕ ДЛЯ КАРМАННЫХ КОМПЬЮТЕРОВ



Инструменты •  
разработки  
для Pocket PC

Создание •  
собственных  
программ

Работа •  
с базами данных  
и графикой

 ПИТЕР®

# Владимир Борисович Волков

# Программирование для

# карманных компьютеров

*Текст предоставлен издательством*

*[http://www.litres.ru/pages/biblio\\_book/?art=176381](http://www.litres.ru/pages/biblio_book/?art=176381)*

*Программирование для карманных компьютеров: Питер; Санкт-Петербург; 2006*

*ISBN 978-5-469-01300-6*

## **Аннотация**

Эта книга предназначена для того, чтобы читатель смог быстро научиться создавать мощные и эффективные приложения для наладонных компьютеров Pocket PC. Рынок программ для карманных компьютеров сейчас бурно развивается, и именно в этой отрасли программист-одиночка может успешно соперничать с большими корпорациями, которые создают программное обеспечение. Прочитав эту книгу, читатель сможет уверенно и быстро разрабатывать достаточно сложные приложения для Pocket PC. Огромное количество разнообразных примеров позволит быстро совершенствоваться и развиваться, обогащая свой опыт работы.

Книга предназначена для начинающих программистов.

# Содержание

Введение	7
Необходимое ПО	12
От издательства	15
Глава 1	16
Внутреннее устройство Pocket PC	16
Pocket PC с точки зрения программиста	19
Windows CE и Pocket PC	19
Инструменты программирования для Pocket PC	28
eMbedded Visual Tools 3.0 и Pocket PC 2002 SDK	28
eMbedded Visual C++ 4.0 и Pocket PC 2003 SDK	31
Visual Studio.NET 2003	32
Visual Studio 2005	33
Порядок установки	35
Глава 2	38
Эмулятор	39
Описание эмулятора	39
Ключи командной строки эмулятора	42
Требования к компьютеру	45
Установка Microsoft Loopback Adapter	46
Ограничения эмулятора	47

«Горячие» клавиши эмулятора	48
Эмулятор и Platform Manager	50
ActiveSync	57
Эмулятор и ActiveSync	59
CabWiz и создание установочного комплекта	61
Ключи командной строки	62
Mobile Developer Power Toys	65
ActiveSync Remote Display	65
CECopy	66
Convert_PPC_DAT_to_SP_XML	67
Hopper	68
JShell	68
PPC_Command_Shell	71
RAPI_Debug	73
RAPI_Start	74
TypeIt	74
Общие вопросы создания интерфейса пользователя	76
Глава 3	84
Среда разработки eMbedded Visual Basic 3.0	84
Настройка панелей инструментов	88
Упражнение 3.1. Создание и настройка панели инструментов	91
Панель инструментов Standard	97
Общие свойства окон дизайнеров	101
Использование и настройка палитры	101

компонентов (Toolbox)	
Упражнение 3.2. Настройка палитры компонентов	103
Помещение компонентов на форму	106
Изменение размеров и выравнивание компонентов	107
Упражнение 3.3. Создание простого приложения	110
Настройки среды	116
Управление проектами. Файлы проекта.	121
Компиляция и запуск программы	
Управление файлами проекта	121
Состав и назначение файлов проекта	122
Компиляция и запуск программы	126
Настройки проекта	127
Использование встроенных компонентов eVB	132
Окно настройки свойств (Properties Window)	132
Типы свойств	134
Компонент Form	136
Упражнение 3.4	141
Упражнение 3.5	151
Другие встроенные компоненты	161
Использование компонентов ActiveX	170
Расположение компонентов ActiveX	170
Общие вопросы использования	171

компонентов ActiveX	
Компонент MenuBar	177
Упражнение 3.6	177
Конец ознакомительного фрагмента.	188

# Владимир Борисович Волков

## Программирование для карманных компьютеров

### Введение

Я еще помню Советский Союз. Я помню очереди за всем, кроме макарон, я помню советскую ЭВМ «большой» серии, помню первые советские микро-ЭВМ, самодельные «ZX-Spectrum», компьютер БК-0010 и первую советскую «персоналку», собранную на Киевском «Электронмаше», которая была аналогом IBM 8086. Иногда я себе кажусь просто динозавром, который до сих пор еще не вымер. Я помню даже ощущение гордости от того, что мой компьютерный класс в университете оснащен самыми современными компьютерами. Это были компьютеры IBM 486, 133 МГц, 32 Мбайт ОЗУ и жесткий диск с емкостью 256 Мбайт. Тогда это были просто роскошные машины.

Когда я покупал свой Pocket PC, я не вспоминал об этом. Мне нужно было что-то удобное. То ли электронная записная книжка, то ли блокнот, то ли ежедневник. Чтобы можно

было и мысли записать в дороге, и почитать, и поиграть, и не забыть сделать нужные дела. Я покупал карманный компьютер не так, как это делают все нормальные люди. Я сначала купил это устройство, а потом стал выяснять, что именно попало мне в руки.

Продавец-консультант при покупке не смог мне толком объяснить, чем отличается электронная записная книжка от Palm, а Palm, в свою очередь, от Pocket PC. «Это, в общем-то, одно и то же, – сказал он. Можно книжки читать, можно записывать что-то. Pocket PC красивее, он цветной. И на нем стоит такая маленькая Windows.» И это все, что я знал о своем компьютере на момент покупки. И оказалось, что эта машинка стоит столько же, сколько и обычный современный компьютер средней категории без монитора. Каково же было мое удивление, когда я обнаружил у себя в руках не дорогой аналог электронной игры «Tetris», а полноценный компьютер. В придачу ко всему он был оснащен не «маленькой», а практически полноценной операционной системой Windows. Систему Linux обнаружить там было бы не так удивительно, поскольку я давно знаю, какой маленькой, без потери функциональности, при необходимости может становиться эта операционная система.

Познакомившись с машинкой лучше, я выяснил, что процессор у нее мощнее, чем у тех IBM 486, которыми я когда-то гордился. У них было больше памяти, и на моем новом карманном компьютере можно было даже играть в

Doom. Конечно, в Doom я не играю уже давно, но сама эта возможность радовала и вызывала теплые воспоминания о коллективных сражениях в локальной сети.

Прошло еще немного времени. Вдоволь наигравшись со своим новым компьютером и установив на него множество разных программ, я пережил незабываемые мгновения «жесткой» перезагрузки, когда система вернулась к своему первоначальному состоянию. Куда же делось все, что я три месяца устанавливал? После этого я, наконец, задался вопросом, как же создаются программы для Pocket PC? Судя по количеству бесплатных и условно-бесплатных программ, для этого вряд ли надо было покупать специализированное программное обеспечение.

После того, как я задал себе этот вопрос, моя жизнь не стала легче. Есть такой сорт людей, о которых можно сказать, что их всегда мучает любопытство и они никогда не идут легкими путями. Я как раз из таких людей. Знаете ли, есть определенная прелесть в том, чтобы сначала выполнить жесткую перезагрузку своего Pocket PC, а потом уже обнаружить в нем утилиту QBackup, которая позволяет безболезненно восстановить состояние системы. Именно поэтому в поиске инструментов программирования для своего «наладонника» я потратил много времени на исследование всяких путей. Я пытался заставить работать на Pocket PC программы, написанные на Java, и программировать для Pocket PC на нем самом, а не на настольном компьютере.

Наигравшись с нестандартными средствами, я решил попробовать пойти законным путем, который предлагает Microsoft. Я принял решение воспользоваться инструментами программирования, которые были созданы разработчиком операционной системы для Pocket PC. Загрузив с сайта компании Microsoft инструменты разработки, я был приятно удивлен. Я получил в свое распоряжение полноценные среды разработки на языках Visual Basic и Visual C++, эмуляторы и отладку приложений. Можно было даже использовать низкоуровневое программирование на языке Assembler. Все было предусмотрено для того, чтобы можно было сосредоточиться на разработке логики программы, а не стандартных элементов интерфейса. Но и освоение этих довольно простых инструментов программирования отняло немало сил.

Эта книга написана для того, чтобы помочь людям, которые начинают программировать для Pocket PC. И не важно, будете ли вы заниматься этим просто ради интереса, или собираетесь сделать программирование для Pocket PC частью своей профессиональной деятельности. Прочтя эту книгу и выполнив приведенные в ней упражнения, вы сможете уверенно и быстро создавать достаточно сложные приложения для вашего наладонника. Конечно, как и любое мастерство, программирование не подразумевает остановки в развитии и достижения всезнания. Прочтя эту книгу, вы не будете знать о программировании для Pocket PC все, но вы очень быстро получите достаточно большой и разнообразный опыт, кото-

рый поможет вам совершенствоваться и развиваться, обогащая свои знания все новыми находками и овладевая приемами, умениями и навыками.

Даже если ваш опыт программирования весьма невелик, вы все равно получите несомненную пользу от чтения этой книги и достигните практических результатов. Каждая глава, каждый раздел, каждая новая тема сопровождаются выполнением полезных упражнений или созданием отдельного приложения. Я рассчитывал на то, что книгу будут читать в первую очередь не профессиональные программисты, а люди, интересующиеся тем, как можно достаточно легко и быстро написать приложение для Pocket PC. Весь исходный код и алгоритм действий описан на страницах книги.

# Необходимое ПО

Чтобы выполнять упражнения, приведенные в большей части книги, необходимо установить определенное программное обеспечение

В этой книге речь будет идти в основном о Pocket PC 2003. Но, поскольку компания Microsoft остается трогательно верной своей традиции обратной совместимости, все программы, которые работали на Pocket PC 2002, могут работать и с версией Pocket PC 2003. Компания Microsoft исключила язык Visual Basic из средств разработки программ для Pocket PC 2003, но при этом оставила возможность запускать программы, написанные на этом языке. Именно поэтому изучение начнется с разработки программ на eMbedded Visual Basic 3. После чего будет рассмотрен процесс разработки на eMbedded Visual C++ 3. Для обоих этих языков необходимо установить пакет eMbedded Visual Tools 3.0 2002 Edition.

Чтобы загрузить и установить этот инструмент, необходимо выполнить следующие действия.

1. Перейти на веб-страницу сайта компании Microsoft, располагающуюся по адресу <http://msdn.microsoft.com/mobility/downloads/updates/default.aspx>.

2. На открывшейся странице Products & Updates найти ссылку на средство разработки eMbedded Visual Tools 3.0 – 2002 Edition и щелкнуть мышью на ней.

3. Перейдя по этой ссылке, нужно нажать кнопку Download, располагающуюся в правой части страницы. Через некоторое время на ваш компьютер будет загружен файл с наименованием evt2002web\_min.exe. Это самораспаковывающийся архив. При запуске этого файла будет отображено диалоговое окно, в котором нужно будет указать путь для распаковки архива. Каталог, в который вы хотите распаковать файлы, необходимо указать в поле Unzip To Folder, после чего следует нажать кнопку Unzip.

4. После распаковки необходимо найти в соответствующем каталоге файл setup.exe и запустить его. После этого останется лишь следовать инструкциям программы установки, выбирая пункт Установить все.

### **ПРИМЕЧАНИЕ.**

*Поскольку файл evt2002web\_min.exe достаточно объемный, а доступ в Интернет не всегда позволяет за один раз быстро выкачать большой файл, кажется логичным скачать этот файл при помощи какого-нибудь менеджера закачек. Но на веб-странице нет прямой ссылки на этот файл. Однако получить ее достаточно легко. В браузере нужно открыть исходный код страницы, на которой расположена кнопка Download. После этого нужно просто отыскать в исходном коде прямую ссылку на файл evt2002web\_min.exe.*

После установки комплекта разработчика для Pocket PC 2002 необходимо загрузить и установить такой же комплект для более поздней версии Pocket PC 2003. Для этого нужно на странице Products & Updates найти ссылку на пакет eMbedded Visual C++ 4.0. После этого нужно произвести те же действия, что и для предыдущей версии пакета.

Эти два комплекта позволяют разрабатывать полноценные приложения для Pocket PC. Но у компании Microsoft есть еще одна новая технология, которая позволяет разрабатывать приложения для Pocket PC 2003. Эта технология носит наименование .NET. Чтобы создавать приложения, работающие на этой платформе .NET, необходимо использовать среду Visual Studio 2003. Увы, эту среду невозможно загрузить бесплатно с сайта Microsoft, однако ее можно купить у дилеров этой компании.

Если вы не можете приобрести Visual Studio 2003, то это не значит, что программирование для .NET на Pocket PC закрыто для вас. Есть и другие средства разработки для этой платформы, о которых будет рассказано в следующих главах книги.

## От издательства

Ваши замечания, предложения и вопросы отправляйте по адресу электронной почты [comp@piter.com](mailto:comp@piter.com) (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

Подробную информацию о наших книгах вы найдете на веб-сайте издательства: <http://www.piter.com>.

# Глава 1

## Рocket PC с разных точек зрения

### Внутреннее устройство Rocket PC

Если не брать в расчет сам факт миниатюрности и отсутствие некоторых привычных на настольном ПК устройств, таких как клавиатура и мышь, то можно считать, что Rocket PC является полноценным компьютером. Мало того, по своим параметрам он является куда более производительной машиной, чем многие, совсем недавно сошедшие со сцены компьютеры.

Как и в каждом настольном компьютере, сердцем Rocket PC является микропроцессор, установленный на материнскую плату и соединенный с различными периферийными устройствами при помощи шины. Поскольку основное достоинство Rocket PC это компактность и мобильность, то устройств, подключаемых к нему, существует не очень много. Сначала их вообще не было, но со временем все же появились мобильные клавиатуры и некоторые другие внешние устройства. Поэтому Rocket PC вполне следует известному изречению «все свое ношу с собой».

Большинство современных Rocket PC снабжено следую-

щими устройствами и органами управления:

- ◆ TFT-экран. Для Pocket PC стандартным является размер 320x240 пикселей. Экран играет роль и клавиатуры, и мыши, поэтому для работы с ним в состав Pocket PC обычно входит стилус, при помощи которого осуществляется щелчок на определенной области экрана.

- ◆ Кнопка питания.

- ◆ Кнопка включения диктофона.

- ◆ Программируемые кнопки (кнопки приложений).

- ◆ Навигационная кнопка.

- ◆ Светодиоды индикации состояния батарей и соединений.

- ◆ Разъем подключения кабеля соединения с настольным компьютером.

- ◆ Слот для подключения расширений.

- ◆ Слот для подключения дополнительной памяти.

- ◆ IRDA-устройство для беспроводной связи на короткие расстояния в инфракрасном диапазоне.

- ◆ Антенна (если Pocket PC оснащен встроенным адаптером Wi-Fi для беспроводной связи).

- ◆ Микрофон.

- ◆ Микродинамик.

- ◆ Разъем для наушников.

- ◆ Кнопки мягкой и жесткой перезагрузки.

- ◆ Аппаратный регулятор громкости (присутствует только в некоторых моделях).

Внутри корпуса Pocket PC, как уже упоминалось, находится системная плата с установленным на ней микропроцессором и микросхемами дополнительных устройств. Также на материнской плате размещены платы памяти ROM и RAM.

Pocket PC производится множеством фирм на основе разных типов микропроцессоров. Несмотря на то что 80 % этих устройств работают с микропроцессорами ARM, среди наладонников присутствуют также устройства на чипах MIPS и SH. Средства разработки от Microsoft поддерживают создание программ для всех видов процессоров, включая x86, который применяется для запуска на эмуляторе, в то время как сторонние производители средств разработки ограничиваются каким-либо одним типом.

Более подробно о всех типах микропроцессоров, которые сертифицированы для работы с Windows CE, вы можете узнать на русском сайте Microsoft по адресу: <http://www.membedded.ru/processors.aspx>.

# **Pocket PC с точки зрения программиста**

## **Windows CE и Pocket PC**

Для программиста общение с устройством – это прежде всего общение с его операционной системой. Даже те программисты, которые пишут драйверы на языке ассемблера, машинных кодах или языке С, все равно пишут их для определенной операционной системы.

Но ознакомившись с историей развития Pocket PC мы можем обнаружить, что с этими компьютерами связывают очень уж много названий операционных систем. В списке находятся Windows CE 2.11, Windows CE 3.0, Windows CE 4.0, Windows CE 4.1, Windows CE 4.2, Windows CE 5.0, Windows CE.NET, Pocket PC 2000, Pocket PC 2002, Windows Mobile 2003, Windows Mobile 2003 SE и Windows Mobile 5.0.

Дело в том, что Windows CE – это модульная, настраиваемая в очень широком диапазоне операционная система, которая в своем чистом виде нигде не используется. Когда производитель создает устройство, то после окончательной компоновки всех комплектующих, определения установленных и подключаемых устройств, органов управления и всего спектра выполняемых функций, наступает этап сборки опе-

рационной системы. Из выбранной версии Windows CE создается операционная система для конкретного устройства. Эта операционная система практически собирается из модулей Windows CE как конструктор. В качестве отдельных деталей к ней добавляются модули, которые разрабатывает сам производитель устройства и, возможно, дополнительные приложения. Такая операционная система для Pocket PC в ее современном исполнении называется Windows Mobile for Pocket PC, а первые ее варианты назывались Microsoft Pocket PC. Версии Windows Mobile обычно меняются вместе с версиями Windows CE, из которых они собираются.

В таблице 1.1 показано, как соответствуют друг другу поколения устройств и версии соответствующих операционных систем.

**Таблица 1.1. Соответствие поколений устройств Pocket PC, версий Windows Mobile и Windows CE**

<b>Устройство</b>	<b>Версия OS</b>	<b>Windows CE</b>
Palm-size PC	Windows CE 2.11	2.11
Pocket PC 2000	Microsoft Pocket PC	3.0
Pocket PC 2002 Smartphone 2002	Microsoft Pocket PC 2002 или Windows Mobile 2002	4.0
Pocket PC 2003 Smartphone 2003	Microsoft Pocket PC 2003 или Windows Mobile 2003	4.20
Pocket PC 2003 SE Smartphone 2003 SE	Microsoft Pocket PC 2003 SE или Windows Mobile 2002 SE	4.21
Новые устройства	Windows Mobile 5.0	5.0

Нельзя не признать, что Microsoft остается верной своей давней традиции обратной совместимости. Начиная с версии Windows CE 3.0, подавляющее большинство программ, созданных для более ранних версий, запускается и на более свежих версиях OS. Практически все программы, предназначенные для Pocket PC 2002, без особых проблем запускаются на Pocket PC 2003 без перекомпиляции, если при их создании были использованы обращения только к документированным возможностям программных интерфейсов.

Название Windows CE.NET является общим для всех версий Windows CE 4.x. Дело в том, что начиная с версии Windows CE 4.0, Microsoft объявила о включении в состав Windows CE своей новой технологии. NET в виде специального исполнения для встраиваемых устройств Compact Framework (CF). В версию Windows CE 4.0 CF была включена со статусом Beta, в версию 4.1 включалась окончательная реализация, а в версию 4.2 вошла уже. NET Framework service pack 2.

Несмотря на то, что Pocket PC – маленький компьютер и выглядит почти как детская игрушка, программирование для него является вполне серьезным делом. Операционная система Windows Mobile (Windows CE) предоставляет программисту подмножество функций Win32 API и других программных интерфейсов, с которыми программисту придется взаимодействовать на настольном компьютере.

Достаточно простая схема, отображающая архитектуру

операционной системы, приведена на рис. 1.1.



Рис. 1.1. Схема архитектуры Windows Mobile для Pocket PC.

Расшифровка терминов, использованных в этой схеме, приведена в следующем списке.

OAL (OEM Adaptation Layer) – минимальное программное обеспечение, необходимое для того, чтобы стандартное ядро операционной системы начало взаимодействовать с устройством. Обычно OAL включает в себя код загрузки ядра и набор нестандартных драйверов.

GWES (Graphics, Windowing, and Events Subsystem) включает в себя графический интерфейс устройства (GDI, Graphics Device Interface) и компоненты пользовательского интерфейса.

Windows CE предоставляет программисту богатейший набор разнообразных API, позволяющий взаимодействовать со всеми устройствами и подсистемами OS.

Core OS Interface отвечает за базовые операции операционной системы по управлению объектами ядра, памятью системы, сообщениями, системным временем, вызовами отладки, динамическими загружаемыми библиотеками.

Блок DLL содержит процедуры управления загрузкой и выгрузкой динамических библиотек, а также вызовом функций из них. Интерфейс Fiber предназначен для создания потоков, управляемых «вручную». В блок Memory Management входят процедуры и функции для управления распределением памяти. Интерфейс Power Management предназначен для управления потреблением электропитания. Интерфейс Message Queue Point-to-Point позволяет с минимальным использованием ресурсов осуществлять обмен сообщениями между окнами, устройствами и ядром. Блок Process and Thread отвечает за создание процессов и потоков, реализацию и управление многопоточной моделью.

Интерфейс Time обслуживает все системные потребности в определении времени, от получения и установки текущего системного времени до миллисекундного измерения вре-

менных интервалов. API ToolHelp используется для отладки, а API Pointer обеспечивает управление графическим курсором. Блок Stylus занимается организацией взаимодействия программы с командами, подаваемыми пользователем при помощи стилуса. Блок Authentication отвечает за управление безопасностью системы и занимается защитой от неавторизованного доступа. Он поддерживает механизмы авторизации и аутентификации, отвечающие протоколам NTLM и Kerberos. Блок Cryptography содержит механизм шифрования информации. Механизм опознавания систем, пользователей, приложений и сервисов, с которыми устанавливается контакт через сеть, управляется блоком Certificates. Блок RAS API и Dial-Up Networking занимается установкой подключения к удаленному компьютеру, а через него к локальной или глобальной сети при помощи протокола PPP. В блоке Fonts содержится интерфейс управления системными шрифтами, а блок MLang отвечает за определение и преобразование текстовых ресурсов одной кодировки в другую. Интерфейс к функциям программно-организованной клавиатуры Pocket PC организован при помощи API Software-Based Input Panel (SIP API).

Блок JScript 5.5 отвечает за поддержку одноименного скриптового языка общего назначения. Организация доступа к Active Directory Service реализована при помощи технологии Lightweight Directory Access Protocol (LDAP). Блок Multilingual User Interface (MUI) отвечает за многоязычный

интерфейс пользователя. Он позволяет создавать приложения, которые могут переключать свой интерфейс с одного языка на другой. Очередь сообщений Message Queuing (MSMQ) позволяет приложениям обмениваться сообщениями в распределенных сетях. При помощи блока Network User Interface (NetUI) производится определение конфигурации сетевых и модемных подключений. В API Object Exchange Protocol (OBEX) входит набор функций, позволяющий устройствам обмениваться данными в упрощенном порядке через IrDA или Bluetooth.

Набор интерфейсов для работы с хранилищем данных и реестром содержит в себе дополнительные функциональные блоки. В блоке Database реализована работа со встроенной базой данных Windows CE, представляющей собой легкую файловую базу данных. Она позволяет использовать элементарную функциональность по созданию, хранению, сортировке и доступу к полям записей. Блок File I/O содержит набор функций и процедур доступа к файлам и каталогам, а также методы для их создания, удаления и переименования. API File Mapping позволяет связывать виртуальное адресное пространство процесса с определенной частью файла или целым файлом, давая возможность процессу получить доступ к содержимому файла через указатели, а также позволяя разделять доступ к файлу между несколькими процессами.

При помощи File System Driver (FSD) файловая структура хранилища данных (например, карты памяти) встра-

ивается в общую структуру файловой системы устройства. В API Registry содержится набор функций для работы с реестром. Информационная модель Pocket Outlook Object Model (POOM) позволяет использовать интерфейсы Pocket Outlook для организации пользовательской информации, которая имеет сходную структуру. Разработчик может использовать эту модель для хранения информации об адресах, контактах и расписании пользователя. API Windows Networking API/Redirector обеспечивает доступ к файловой системе на удаленном компьютере. Блок Virtual Private Networking позволяет объединять Pocket PC и настольный компьютер в единую сеть, а блок Waveform Audio отвечает за воспроизведение звуков. Поддержка протоколов HTTP и FTP и высокоуровневые функции работы с WinSock реализуются при помощи API Windows Internet Services (WinInet), в то время, как API Windows Sockets предоставляет доступ к разнообразным сетевым транспортным протоколам.

Легко заметить, что Windows CE с точки зрения богатства интерфейсов программирования мало в чем отличается от настольного компьютера. Можно сказать, что, программируя для Pocket PC, вы будете работать практически со всеми интерфейсами, с которыми можно работать, создавая приложения для полноценной операционной системы Windows. Разница состоит лишь в том, что количество функций, макросов, COM-интерфейсов и структур в Windows CE меньше, чем в полновесной операционной системе. Это и понят-

но. Какие-то функции оказались лишними из-за разницы в наборе управляющих компонентов компьютера, а другими пришлось пожертвовать для того, чтобы обеспечить работу операционной системы и приложений в оперативной памяти размером 32 Мбайта.

На самом деле разработчики Microsoft проделали большую работу, создавая Windows CE и заставляя ее работать в условиях ограниченных ресурсов. Невольно закрадывается мысль о том, что если бы такая же работа была проделана с десктопными и серверными моделями, то нам долго не пришлось бы беспокоиться об апгрейде наших компьютеров.

# Инструменты программирования для Pocket PC

Компания Microsoft предлагает разработчикам приложений для Pocket PC множество разных инструментов, полностью покрывающих все потребности в создании приложений самого разного назначения, размера и функциональности. В этом разделе будут кратко описаны средства разработки и их предназначение. Естественно, основным источником сведений в этом случае будет сайт Microsoft, в частности опубликованная там в мае 2005 года статья *Introduction to Development Tools for Windows Mobile-based Devices*.

Мы обсудим следующие инструменты программирования:

- ◆ eMbedded Visual Tools 3.0;
- ◆ eMbedded Visual C++ 4.0;
- ◆ Visual Studio.NET 2003;
- ◆ Visual Studio.NET 2005.

## **eMbedded Visual Tools 3.0 и Pocket PC 2002 SDK**

Этот набор инструментов включает в себя четыре составные части:

- ◆ eMbedded Visual Basic;
- ◆ eMbedded Visual C++ 3.0;
- ◆ Pocket PC 2002 SDK;
- ◆ Smartfone 2002 SDK.

Все четыре части устанавливаются из одного установочного файла.

## **eMbedded Visual Basic**

В этой среде вы можете создавать приложения, которые будут работать на Pocket PC 2002, Pocket PC 2003 и Pocket PC 2003 SE, то есть приложения для наиболее распространенных сегодня платформ.

Несмотря на то, что Microsoft настоятельно рекомендует отказаться от этого инструмента, я считаю, что он отлично подходит для тех, кто хочет освоить программирование для Pocket PC. Преимущества среды eVB заключаются в скорости и легкости освоения как языка, так и среды разработки, в огромном количестве примеров кода и приложений в Интернете, в простоте и понятности кода. К недостаткам можно отнести лишь ограниченное число встроенных элементов управления, медлительность приложений, вызванная интерпретирующей природой языка, необходимость иметь на устройстве run-time библиотеку, и некоторую ограниченность применения. Например, в этой среде нельзя создавать компоненты ActiveX.

Среда разработки eVB дает возможность быстро создавать приложения с несколькими экранными формами, которые позволяют работать чаще всего с текстовой информацией. Это могут быть всякого рода тестировщики знаний, маленькие обучающие программы, записные книжки или игры, к которым можно отнести шашки и логические головоломки.

## **eMbedded Visual C++ 3.0**

Приложения, созданные в eVC 3.0, получают полный доступ ко всей функциональности Pocket PC 2002 и будут работать на Pocket PC 2003 и Pocket PC 2003 SE, если написаны с использованием только документированных функций API.

Приложения, написанные на eVC 3.0, выполняются гораздо быстрее, чем приложения, написанные на eVB. При помощи eVC разработчик практически не ограничен в своих возможностях. Можно создавать приложения, которые компилируются в «родной» код устройства, динамически загружаемые библиотеки и компоненты ActiveX. Среда eVC позволяет использовать легкую интеграцию с COM и WinAPI, а библиотека MFC предоставляет в распоряжение разработчика шаблоны для создания сложных приложений с функциональностью, которая в eVB была просто недоступна. Но за все надо платить, и сложные многофункциональные при-

ложения потребуют много сил на изучение среды, языка, детального знакомства с вызовами API и внимательного отслеживания правильности использования памяти вашими программами. Это важно и для «большого» компьютера, но на Pocket PC с его ограниченными ресурсами утечки памяти могут иметь весьма плачевные последствия.

## **eMbedded Visual C++ 4.0 и Pocket PC 2003 SDK**

Пакеты eVC 4.0 и Pocket PC 2003 SDK поставляются как два отдельных комплекта установки. При этом, если вы хотите разрабатывать полноценные приложения для Pocket PC 2003, то вам надо будет еще установить и Service Pack 2. Если же вы собираетесь использовать eVC 4.0 для разработки приложений для Pocket PC 2003 SE с поддержкой VGA-экрана разрешением в 192 dpi и возможностью поворота ориентации экрана с книжной на альбомную, то необходимо установить Service Pack 4, дополнительные образы для эмулятора и загрузить с сайта Microsoft пакеты Mobile Application Development Toolkit и Developer Resources for Windows Mobile 2003 Second Edition. Ссылки на загрузку всех дополнительных ресурсов можно найти в нижней части той же страницы на сайте Microsoft, с которой будет производиться загрузка eVC 4.0.

Все, что было сказано о eVC 3.0, остается верным и для

eVC 4.0, с той лишь разницей, что разработчик получает возможность создавать полнофункциональные приложения для Pocket PC 2003 и Pocket PC 2003 SE. При этом eMbedded Visual Basic был исключен из числа инструментов разработки для Pocket PC 2003, как утверждает Microsoft, «по многочисленным просьбам трудящихся». Однако у меня есть подозрение, что просто надо было освободить место на устройстве для среды исполнения. NET CF, поэтому и была удалена библиотека run-time для Visual Basic.

## **Visual Studio.NET 2003**

Как уже было сказано ранее, пакет разработки Visual Studio.NET 2003 нельзя загрузить с сайта компании Microsoft. Однако именно эту среду компания Microsoft в свое время позиционировала, как один из основных инструментов разработки для Pocket PC 2003. В этой среде разработчик может использовать языки Visual Basic.NET и C#. На обоих языках можно создавать приложения только для исполнения в среде. NET Compact Framework.

Работать с обоими этими языками – одно удовольствие. За счет введения общих стандартов языки стали похожи друг на друга, а также на Pascal и Java одновременно. Visual Basic стал полностью объектно-ориентированным, что добавляет ему функциональности, а программам придает стройность и структурированность. Большинство приложений общего на-

значения в Visual Studio.NET 2003 удобно создавать и отлаживать. Собственно говоря, для этого Visual Studio.NET 2003 и создавалась. Если же вам надо создавать приложения, которые выполняют очень интенсивные видеооперации или ведут интенсивные расчеты, тогда вам придется выбрать eVC.

## Visual Studio 2005

Этот инструмент разработчика пока находится в стадии Beta, и его можно бесплатно загрузить с сайта компании Microsoft, как и соответствующий ему SDK (Windows Mobile SDK 5.0).

Разрабатывая Visual Studio 2005, Microsoft пошла по пути комплексного решения и совместила разработку как управляемого, так и чистого кода для компактных устройств. Список возможностей Visual Studio 2005 приведен ниже.

- ◆ Разработка родного кода для Windows Mobile 2003, Windows Mobile 2003 Second Edition или Windows Mobile 5.0 на языке C++.

- ◆ Разработка управляемого кода для выполнения под управлением. NET Compact Framework 1.0 на платформах Windows Mobile 2003, Windows Mobile 2003 Second Edition или Windows Mobile 5.0 на языках C# или Visual Basic.NET.

- ◆ Разработка управляемого кода для выполнения под управлением. NET Compact Framework 2.0 на платформе

## Таблица 1.2. Назначение и возможности инструментов разработки

	eMbedded Visual Tools 3.0 (2002 Edition)		eMbedded Visual C++ 4.0 (с SP4)	Visual Studio .NET 2003	Visual Studio .NET 2005
	eMbedded Visual C++ 3.0	eMbeddedVisual Basic 3.0			
<b>Тип кода</b>					
Родной	X		X		X
Интерпретируемый		X			
Управляемый				X	X
Выполняемый на сервере				X	X
<b>SDK для платформ</b>					
Pocket PC 2000 и Pocket PC 2002	X	X		X	
Smartphone 2002	X				
Windows Mobile 2003			X	X	X
Windows Mobile 2003 Second Edition			X	X	X
Windows Mobile 5.0					X

## Таблица 1.3. Инструменты. NET Compact Framework и их поддержка мобильными ОС

	Версия 1.0	Версия 2.0
<b>Среда разработки</b>		
Visual Studio .NET 2003	X	
Visual Studio 2005	X	X
<b>Версия Windows Mobile</b>		
Windows Mobile 5.0	Встроенная (1.0 SP3)	Устанавливается
Windows Mobile 2003 Second Edition	Встроенная (1.0 SP2)	Устанавливается (только для Pocket PC)
Windows Mobile 2003	Встроенная (1.0 SP1)	Устанавливается (только для Pocket PC)
Smartphone 2002		
Pocket PC 2002	Устанавливается	
Pocket PC 2000	Устанавливается	

◆ Разработка управляемого кода для выполнения под управлением. NET Compact Framework 2.0 на платформе Pocket PC с операционной системой Windows Mobile 2003 на языках C# или Visual Basic.NET.

Вся эта информация в более наглядном виде показана в табл. 1.2 и 1.3.

## Порядок установки

Все инструменты разработки от Microsoft вполне мирно уживаются на одном компьютере, но только при условии их правильной установки. Чтобы установить средства разработ-

ки в правильном порядке, сначала надо удалить все установленные на данный момент среды от Microsoft и перезагрузить компьютер. После этого установить инструменты разработчика в необходимом порядке.

1. Установить Microsoft ActiveSync 4.0. Все инструменты разработки для мобильных устройств требуют наличия ActiveSync 4.0 для разработки и поставки приложений.

2. Установить eMbedded Visual Tools – 2002 Edition, Pocket PC 2002 SDK и Smart-phone 2002 SDK.

3. Установить eMbedded Visual C++ 4.0 и Service Pack 4.

4. Установить Visual Studio.NET 2003.

### ***ПРИМЕЧАНИЕ.***

*Устройство Pocket PC 2003 и эмулятор Pocket PC 2003 уже содержат в себе предустановленную .NET Compact Framework. На более ранние устройства надо предусматривать включение среды выполнения. NET Compact Framework в комплект поставки. При отладке устройств в среде разработки. NET Compact Framework, при необходимости, устанавливается на устройство автоматически.*

5. Установить Pocket PC 2003 SDK.

6. Если вы собираетесь тестировать и отлаживать приложения с высоким разрешением и альбомной ориентацией экрана, то нужно установить эмулятор для Pocket PC 2003

Second Edition.

7. УСТАНОВИТЬ Developer Resources for Windows Mobile  
2003 Second Edition.

8. УСТАНОВИТЬ Visual Studio 2005.

9. УСТАНОВИТЬ Windows Mobile 5.0 SDK.

# Глава 2

## Общие вопросы программирования для Pocket PC

Поскольку большинство сред разработки, которые мы будем обсуждать в данной книге, произведены в Microsoft, не удивительно, что они используют для создания, отладки и поставки приложений одни и те же инструменты. Таковыми инструментами являются эмулятор устройства Pocket PC, программа для соединения мобильного устройства с настольным компьютером ActiveSync, программа для создания поставочного комплекта приложения CabWiz, программа для соединения среды разработки с эмулятором через ActiveSync, Emulator ActiveSync Connection Tool, а также набор небольших программ, помогающих при отладке приложения, объединенных в набор Windows Mobile Developer Power Toys.

Также существует некоторый набор рекомендаций по разработке и оформлению пользовательского интерфейса ваших программ. Эти правила не зависят от среды разработки. Именно они и будут обсуждаться в этой главе.

# Эмулятор

Эмулятор является очень важным звеном при создании приложений для Pocket PC. Конечно, приложение можно отлаживать и на самом устройстве, но это крайне неудобно, особенно если устройство установлено в специальный держатель для подключения Pocket PC к стационарному компьютеру. Отладка на эмуляторе позволяет выявить большую часть логических ошибок, которые были допущены при разработке приложения.

В этом разделе речь пойдет об эмуляторе, который входит в состав Microsoft SDK for Pocket PC 2003. Следует учитывать, что другие эмуляторы могут отличаться меньшей или большей поддержкой того или иного оборудования, сервисов и библиотек операционной системы. Впрочем, то же самое можно сказать и о самих устройствах.

## Описание эмулятора

После установки Microsoft Windows Pocket PC 2003 SDK эмулятор можно будет запускать как из среды eMbedded Visual C++ 4.0 (Service Pack 2), так и из среды Visual Studio.NET 2003. Однако его можно запустить и как независимое устройство и пользоваться им для самостоятельного изучения.

При установке SDK в главном меню компьютера создается соответствующая группа программ (рис. 2.1).



Рис. 2.1. Группа ярлыков Pocket PC 2003 SDK.

В этой группе можно отыскать пиктограмму Pocket PC 2003 Emulator, которая позволяет запускать эмулятор как самостоятельное устройство (рис. 2.2):



## Ключи командной строки эмулятора

Щелкнув правой клавишей мыши на пиктограмме эмулятора, и выполнив в контекстном меню команду Свойства, можно заметить, что эмулятор запускается командным файлом emul.cmd. Полный текст файла приведен ниже.

```
start "" "C: \Program Files\Common
Files\Microsoft Shared\Windows CE Tools\
Platman\
bin\PBEmulator.exe" /MemorySize 64 /
Fastbackground /Skin "C: \Program Files
\Windows CE
Tools\wce420\POCKET PC 2003\Emulation
\pocket_pc_emulator_skin.xml" /CEImage
«C: \Program Files\Windows CE
Tools\wce420\POCKET PC 2003\Emulation\
PPC_2003_WWE.bin»
/VMName «POCKET PC 2003 - POCKET PC 2003
Emulator» /HostKey 165
```

Внимательный просмотр этого файла показывает, что в нем просто производится вызов одного файла с несколькими ключами. Эти ключи рассмотрены в списке, приведенном

ниже.

◆ /Video – необязательный параметр. Он устанавливает размеры экрана в пикселах и глубину цвета в битах. Эти параметры не будут приняты во внимание, если для эмулятора будет применена обложка (skin), которая предусматривает свои собственные настройки размера и глубины цвета. Аргументы должны быть записаны в виде ширинавысотаглубина. Ширина страницы должна быть в пределах от 80 до 1024 пикселей, высота – от 64 до 768 пикселей, глубина цвета может принимать значения 8, 16 или 32.

◆ /Ethernet – необязательный параметр. Он позволяет включать или отключать работу контроллера Ethernet в эмуляторе. С этим ключом можно использовать значения None, Shared, Virtualswitch или адрес media access control (MAC) реальной платы адаптера на стационарном компьютере. Этот способ используется, если на компьютере установлено больше одного сетевого адаптера. Эмулятор по умолчанию будет использовать первый найденный адаптер, и если нужно использовать другой адаптер, то нужно указать его MAC-адрес. Предназначение остальных значений приведено ниже.

- None – в эмуляторе недоступна сетевая карта.
- Shared – эмулятор использует ту же сетевую карту (тот же IP-адрес), что и стационарный компьютер. При этом эмулятор может посылать запросы по сети и получать информацию в ответ на эти запросы, но он не воспринимает сетевые сообщения, причиной которых является не он сам.

- Virtualswitch – эмулятор получает свой собственный IP-адрес при помощи DHCP и начинает работать как полноценный хост в сети, доступный как для исходящих, так и для входящих сетевых сообщений.

- ◆ /Skin – необязательный параметр. Он указывает, какую нужно использовать обложку. В качестве параметра должен быть указан файл XML, в котором описана обложка эмулятора в соответствующем стандарте.

- ◆ /CEImage – обязательный параметр. Он задает полное имя файла ядра эмулятора. Собственно говоря, файл ядра и есть сам эмулятор.

- ◆ /MemorySize – необязательный параметр. Он задает количество памяти, используемой ядром для работы. Значение указывается в мегабайтах. Оно должно быть кратно четырем и находиться в промежутке от 32 до 256 Мбайт. Для Pocket PC 2003 это значение не должно быть менее 64 Мбайт.

- ◆ /Fastbackground – необязательный параметр. Он задает высокий приоритет для процесса эмулятора, когда его окно не обладает фокусом ввода.

- ◆ /HostKey – необязательный параметр. В качестве аргумента задается десятичный код клавиши, в сочетании с которой будут срабатывать «горячие клавиши» эмулятора, нажимаемые на клавиатуре стационарного компьютера. Более подробно горячие клавиши эмулятора будут рассмотрены ниже.

- ◆ /VMName – необязательный параметр, который позво-

ляет задавать текст, отображаемый на заголовке окна. Параметр представляет собой строку, которая должна быть помещена в кавычки. Если обложка эмулятора задает свой собственный текст заголовка, то этот параметр будет проигнорирован.

◆ /VMID – необязательный параметр, который позволяет задавать глобальный уникальный идентификатор (GUID) эмулятора. Используя этот параметр, разработчик может запускать одновременно несколько разных конфигураций эмулятора.

## **Требования к компьютеру**

Для того, чтобы эмулятор мог быть запущен на машине разработчика и для его эффективной работы, стационарный компьютер и его программное обеспечение должны удовлетворять некоторым минимальным требованиям, которые перечислены в следующем списке.

◆ Операционная система: Microsoft Windows 2000 Professional или Windows 2000 Server с установленным Service Pack 2, Microsoft Windows XP Home Edition или Windows XP Professional.

◆ Процессор Intel Pentium II или старше с частотой не менее 400 МГц.

◆ Не менее 196 Мбайт оперативной памяти.

◆ Сетевая карта или установленный Microsoft Loopback

Adapter.

### ***ВНИМАНИЕ!***

*Сетевая карта должна быть подключена к сети. Неподключенная сетевая карта не распознается как корректное устройство. При этом, если у вас установлена неподключенная сетевая карта, но работает модемное подключение к Интернету, то это вполне устраивает эмулятор.*

## **Установка Microsoft Loopback Adapter**

Компонент Microsoft Loopback Adapter предназначен для имитации наличия на компьютере сетевой платы, в то время, когда реальный физический сетевой адаптер на машине отсутствует. Процедура установки Microsoft Loopback Adapter приведена в следующем списке.

1. Открыть папку Пуск ► Настройка ► Панель управления.
2. Запустить инструмент Установка оборудования и нажать кнопку Далее.
3. Утвердительно ответить на вопрос, подключено ли устройство к компьютеру, и нажать кнопку Далее.
4. В новом диалоговом окне выбрать последний пункт из списка Добавление нового устройства.
5. Выбрать вариант Установка оборудования, выбранного

из списка вручную.

6. Выбрать из списка пункт Сетевые платы и нажать кнопку Далее.

7. Выбрать значение Адаптер Microsoft замыкания на себя и нажать кнопку Далее.

После того, как устройство будет установлено, из контекстного меню пиктограммы Сетевое окружение нужно выполнить команду Свойства. После этого в открывшемся диалоговом окне следует отыскать установленный адаптер. Скорее всего, он будет обозначен как Подключение по локальной сети. В его контекстном меню тоже нужно выполнить команду Свойства. В диалогом окне свойств нового подключения нужно установить флажок Virtual PC Emulated Ethernet Switch. Этот протокол еще пригодится в дальнейшем.

## Ограничения эмулятора

Конечно, эмулятор не настолько функционален, как настоящий Pocket PC. Основные ограничения эмулятора приведены в соответствующем списке.

1. Единственным протоколом, по которому эмулятор может связываться с внешними устройствами, является TCP/IP.

2. Эмулятор поддерживает только один процессор x86. Соответственно, все приложения, которые выполняются на эмуляторе, должны быть собраны для этой платформы. Это

накладывает ограничения на всякого рода тонкие настройки и оптимизацию быстродействия и памяти, поскольку на реальном устройстве с процессором ARM на уровне машинных команд быстродействие будет совершенно другим. Разработчик не может использовать эмулятор для запуска никаких программ для Pocket PC, если у него нет их исходного кода.

3. Эмулятор не поддерживает контактный экран Pocket PC. Вместо этого в эмуляторе применяется мышь стационарного компьютера.

4. Эмулятор не поддерживает запись звука, независимо от того, снабжен ли микрофоном и программой для звукозаписи стационарный компьютер.

5. Клавиатура эмулятора поддерживает только английскую раскладку, независимо от установленных на стационарном компьютере и эмуляторе региональных настроек.

## **«Горячие» клавиши эмулятора**

Несколько сочетаний клавиш позволяют получить быстрый доступ к функциям эмулятора.

◆ Сочетание **Host + F** выводит на экран диалоговое окно, позволяющее сделать существующую на стационарном компьютере папку каталогом общего пользования, который на эмуляторе будет опознаваться как карта дополнительной памяти. Наличие разделяемого каталога позволяет легко пере-

мещать файлы между эмулятором и стационарным компьютером.

◆ Сочетание Host + P позволяет приостановить и возобновить работу эмулятора.

◆ Сочетание Host + R инициирует жесткую перезагрузку эмулятора с потерей информации и настроек.

◆ Сочетание Host + T позволяет выполнить мягкую перезагрузку эмулятора.

◆ Сочетание Host + F4 завершает работу эмулятора.

### ***ВНИМАНИЕ!***

*При завершении работы с эмулятором вы можете выбрать из выпадающего списка в окне диалога завершения работы пункт **Save Emulator State**. В этом случае текущее состояние эмулятора, включая установленные соединения и запущенные программы, будет сохранено в специальном файле на диск. При последующем запуске эмулятора его состояние будет восстановлено. Это может значительно ускорить работу с эмулятором. Файлы состояния записываются в каталог **C: \Documents and Settings\Имяпользователя\Application Data** с расширением. **vsv** (например, **{31F6C796-7FD5-43AC-A8C2-A7CDB8C00A3B}.vsv**). Удаление этих файлов ведет к потере сохраненного состояния.*

В приведенном выше сочетании горячих клавиш клавиша

Host – это та клавиша, код которой задан ключом /HostKey командной строки эмулятора.

Список возможных кодов клавиш приведен в табл. 2.1.

**Таблица 2.1. Клавиши и соответствующие им десятичные коды**

<b>Клавиша HostKey</b>	<b>Десятичный код</b>
Левый ALT	164
Левый CTRL	162
Левый SHIFT	160
Левая клавиша Windows logo	91
Правый ALT	165
Правый CTRL	163
Правый SHIFT	161
Правая клавиша Windows logo	92

## **Эмулятор и Platform Manager**

Между эмулятором и средой разработки существует соответствующий программный слой. Эта технология обеспечивает взаимодействие среды разработки с реальным устройством или эмулятором. Этот программный слой называется Platform Manager и позволяет среде разработки установить сетевое соединение с устройством (эмулятором), загрузить на него файлы и произвести основные операции отладки на уровне приложения. Сам пакет Platform Manager и вспомогательные инструменты отладки распола-

гаются в каталоге C: \Program Files\ Common Files\Microsoft Shared\Windows CE Tools\Platman\bin\wce420. Каждый из инструментов Platform Manager и каждая среда разработки Microsoft, предназначенная для работы с Pocket PC или Windows CE, имеет в меню команду Configure Windows CE Platform Manager..., которая позволяет настроить запуск нужного эмулятора и в интерактивном режиме отредактировать ключи командной строки.

Теперь следует перейти в папку C: \Program Files\Common Files\Microsoft Shared\ Windows CE Tools\Platman\bin\wce420 и запустить программу cefilevw.exe. Эта программа позволяет просматривать папки эмулятора и копировать файлы на эмулятор. Но сейчас следует обратить особое внимание не на функциональность этой программы, а на пункт меню Connection ► Configure Windows CE Platform Manager. После выполнения этой команды на экран будет выведено диалоговое окно (рис. 2.3).

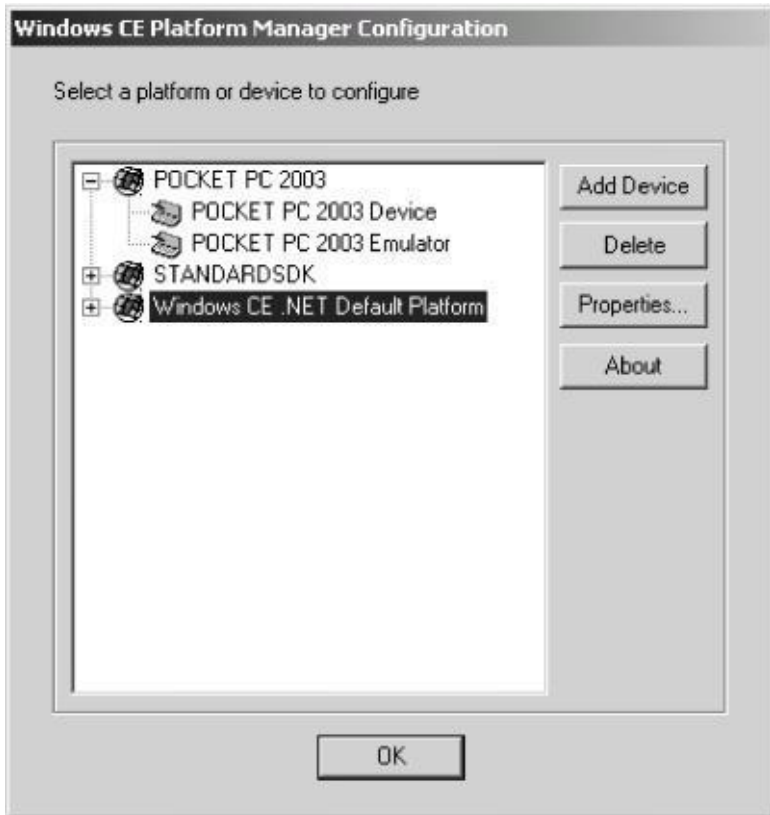


Рис. 2.3. Выбор устройства для настройки.

В списке нужно выбрать пункт Pocket PC 2003 Emulator, после чего следует нажать кнопку Properties. На экране появится очередное окно настройки (рис. 2.4).



Рис. 2.4. Окно настройки эмулятора Pocket PC 2003.

В выпадающем списке Transport нужно выбрать значение TCP/IP Transport For Windows CE, а в списке Startup Server выбрать значение Emulator Startup Server. После нажатия на кнопку Configure, которая относится к списку Startup Server, еще одно окно настройки будет выведено на экран (рис. 2.5).

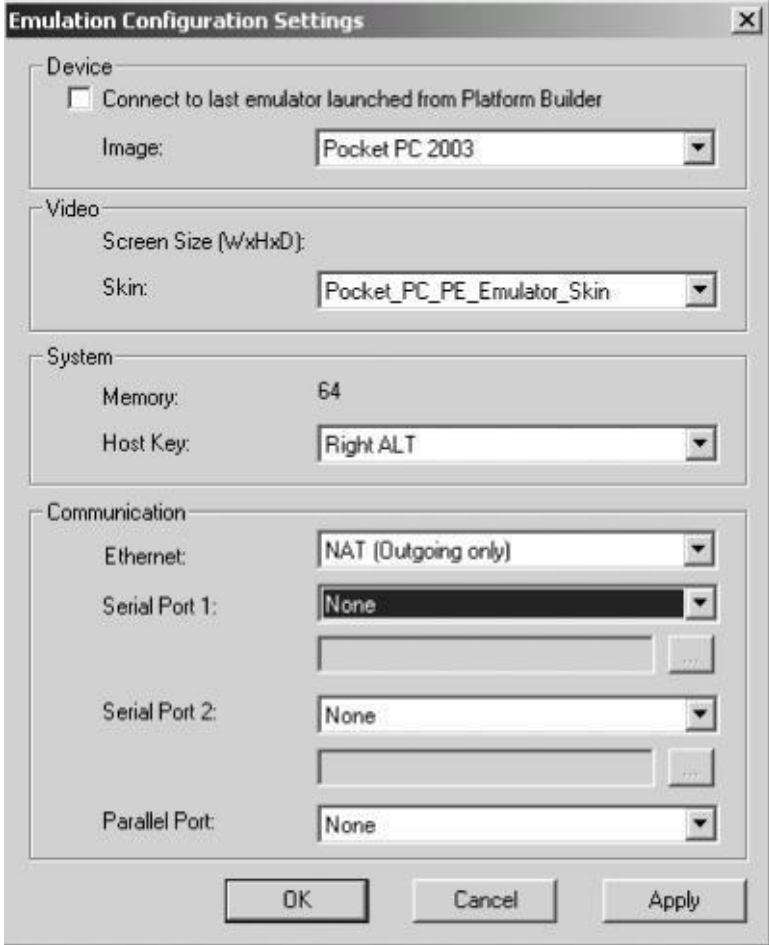


Рис. 2.5. Окно конфигурации стартовых параметров эмулятора.

В этом окне можно выбрать параметры, с которыми будет запускаться эмулятор.

◆ Параметр Device ► Image позволяет указать, какое ядро эмулятора будет загружено. Разработчик может использовать одно из трех стандартных ядер – Pocket PC 2003 или два ядра Pocket PC 2003 Phone Edition с разными сетевыми устройствами.

◆ Параметр Video ► Skin позволяет указать, какая обложка будет выбрана для эмулятора. На обложке для Phone Edition есть дополнительные кнопки дозвона и прекращения связи.

◆ Параметр System ► HostKey позволяет установить клавишу HostKey.

◆ Параметр Communication позволяет задать параметры связи эмулятора с внешним миром.

- Значение Ethernet позволяет установить NAT. Проблемы настройки Virtual Switch будут рассмотрены позже.

- Значение Serial Port 1 позволяет соединить первый COM-порт эмулятора с реальным портом рабочей станции или маппировать его на файл.

- Значение Serial Port 2 позволяет соединить второй COM-порт эмулятора с реальным портом рабочей станции или маппировать его на файл.

- Значение Parallel Port позволяет соединить параллельный порт эмулятора с реальным портом рабочей станции.

После настройки всех необходимых параметров нужно закрыть окно нажатием кнопки ОК. В окне настройки устройства нужно нажать кнопку Test. Это приведет к запуску эмулятора с заданными параметрами, и через две-три минуты будет установлено соединение. Данный факт будет сопровождаться громким звуком, а кнопка Cancel в окне Testing Device Connection сменится кнопкой ОК.

Теперь эмулятор настроен и готов к работе.

# ActiveSync

ActiveSync – это инструмент, предназначенный для связи мобильного устройства (Pocket PC) с настольным компьютером (рис. 2.6).

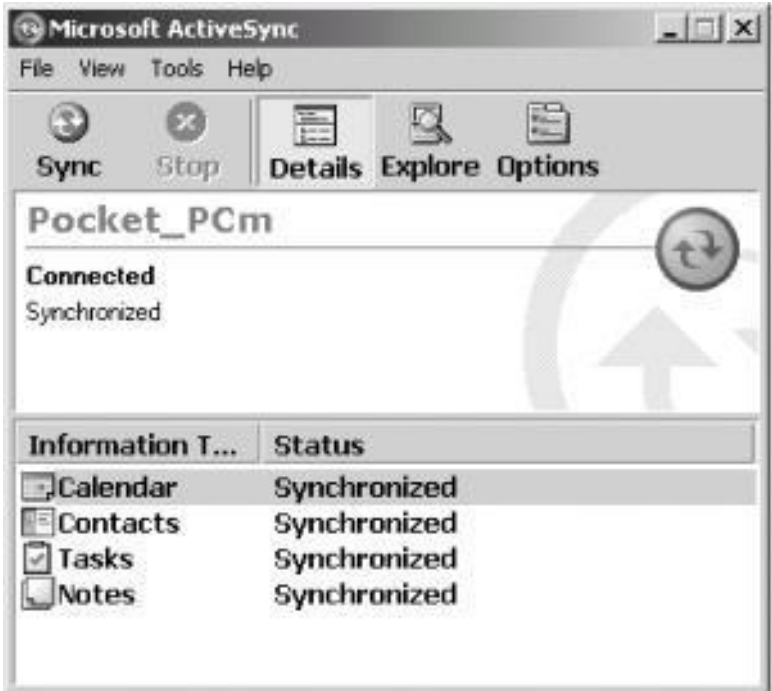


Рис. 2.6. Окно ActiveSync.

На рисунке показана настольная часть ActiveSync. На мобильном устройстве, естественно, работает второй компонент этого приложения.

ActiveSync автоматически отслеживает наличие присоединенного к компьютеру мобильного устройства и, при наличии физического соединения, устанавливает связь и выполняет синхронизацию настольного компьютера и Pocket PC. Таким образом, основное назначение ActiveSync – это автоматизированная синхронизация различных баз данных Pocket Outlook и папок на мобильном компьютере с соответствующими базами данных и папками на настольном компьютере. Кроме этого, при помощи ActiveSync выполняются операции, перечисленные в следующем списке.

1. Резервное архивирование данных и настроек мобильного устройства, с сохранением файла архива на настольном компьютере. Восстановление данных с настольного компьютера на мобильный.

2. Установка и удаление программ на мобильном компьютере, с ведением базы данных установленных программ на настольном компьютере.

3. Импорт и экспорт таблиц баз данных.

Таким образом, ActiveSync является очень полезной программой, и с ней могут взаимодействовать приложения, созданные для работы на мобильном компьютере. Для этого в состав SDK включены соответствующие библиотеки, заголовочные файлы и примеры. Их можно найти в пап-

ке C: \Program Files\Windows CE Tools\wce420\POCKET PC 2003\ActiveSync.

Но есть одна серьезная проблема. По умолчанию ActiveSync упорно не желает работать с эмулятором, как с настоящим устройством. Для того чтобы эмулятор мог установить связь и взаимодействие с ActiveSync настольного компьютера, необходимо предпринять заметные усилия по настройке.

## Эмулятор и ActiveSync

Для настройки соединения эмулятора и ActiveSync на стационарном компьютере необходимо сделать последовательность действий, приведенную в следующем списке.

1. Необходимо установить и настроить Microsoft Loopback Adapter.

2. Установить Microsoft.NET Framework 1.1 SDK на стационарный компьютер. Установочный комплект можно взять с сайта Microsoft бесплатно. Если в системе установлена среда Visual Studio.NET, то и SDK тоже установлен.

3. Настроить соответствующим образом Platform Manager.

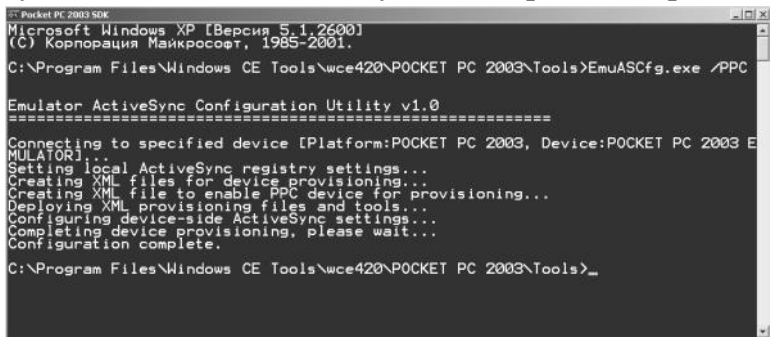
- Надо убедиться, что при настройке транспорта TCP/IP был взведен флажок Use Fixed Address.

- При настройке Startup Server нужно выбрать Emulator Starup Server и для параметра Communication ► Ethernet вы-

брать значение Virtual Switch.

- Протестировать Platform Manager.

4. Необходимо протестировать соединение эмулятора с ActiveSync. Для этого нужно выполнить команду Пуск ► Программы ► Microsoft Pocket PC 2003 SDK ► Pocket PC 2003 SDK. На экран будет выведено окно командного интерпретатора. В командной строке нужно набрать команду `EmuASCfg.exe /PPC` и нажать клавишу Enter. В результате эмулятор будет запущен, а в окне командного интерпретатора будет выведена соответствующая информация (рис. 2.7).



```
C:\Pocket PC 2003 SDK
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.
C:\Program Files\Windows CE Tools\wce420\POCKET PC 2003\Tools>EmuASCfg.exe /PPC

Emulator ActiveSync Configuration Utility v1.0
=====
Connecting to specified device [Platform:POCKET PC 2003, Device:POCKET PC 2003 E
MULATOR]...
Setting local ActiveSync registry settings...
Creating XML files for device provisioning...
Creating XML file to enable PPC device for provisioning...
Deploying XML provisioning files and tools...
Configuring device-side ActiveSync settings...
Completing device provisioning, please wait...
Configuration complete.

C:\Program Files\Windows CE Tools\wce420\POCKET PC 2003\Tools>_
```

Рис. 2.7. Окно командного интерпретатора после выполнения команды.

### ***ВНИМАНИЕ!***

*Команду `EmuASCfg.exe/PPC` необходимо выполнять каждый раз, когда необходимо установить соединение эмулятора с ActiveSync.*

# **CabWiz и создание установочного комплекта**

Установочный комплект для любого приложения должен выполнять некоторые вполне определенные действия, позволяющие развернуть приложение на устройстве. Перечень необходимых действий приведен в списке.

1. Распаковать установочные файлы, если они запакованы.
2. Создать папки, в которые будут копироваться установочные файлы, если такие папки отсутствуют на устройстве.
3. Скопировать файлы в определенные конфигурацией установки папки.
4. Если нужно, то создать какие-либо файлы.
5. При необходимости удалить какие-либо файлы.
6. Внести необходимые изменения в конфигурационные файлы устройства.
7. Внести необходимые изменения в файл реестра на устройстве.
8. Обеспечить достаточную интеллектуальность процесса установки. Установочный комплект должен понимать, сохранять уже имеющиеся на устройстве файлы и настройки или заменить их.
9. Обеспечить необходимую интерактивность установки. Пользователь должен иметь возможность выбрать, куда

именно будет установлена программа, а в некоторых случаях – выбрать, в каком составе будет установлена программа.

Большинство этих задач решается при помощи механизма AppInstall (установка приложений), частью которого является программа CabWiz, создающая установочные пакеты (файлы с расширением. cab), которые затем могут быть установлены на Pocket PC.

Более предметно применение инструмента CabWiz будет рассмотрено в следующих главах, а в этом разделе будет приведена некоторая информация, которая может понадобиться для создания установочных пакетов.

## Ключи командной строки

Программа CabWiz.exe находится в каталоге C: \Program Files\Windows CE Tools\ wce420\POCKET PC 2003\Tools. Файл CabWiz.exe необходимо запускать с соответствующими ключами в командной строке.

```
cabwiz.exe          «inf_file»          [/dest
dest_directory]    [/err  error_file] [/cpu
cpu_type [cpu_type]]
[/platform          platform_label
[platform_label]]
```

Расшифровка ключей приведена ниже.

В параметре `inf_file` указывается полное имя конфигурационного файла (включая путь, взятый в кавычки) с расширением. `inf`, в котором содержится вся информация, на основе которого программа CabWiz создает установочный комплект. Папка, в которую будут помещены созданные файлы с расширением. `cab`, указывается в параметре `dest_directory`. Если этот параметр не указан, то файлы. `cab` будут помещены в папку, в которой находится конфигурационный файл. Имя файла, в который будет записываться отчет об ошибках при создании установочного комплекта, указывается при помощи ключа `errlog_file`. Если этот параметр не задан, то программа будет выводить ошибки и предупреждения в окно сообщений на экране. При помощи ключа `cpu_type` разработчик может указать перечень типов процессоров, для каждого из которых будут создаваться установочные файлы. Возможные значения этого параметра соответствуют названиям папок в каталоге `C:\Program Files\Windows CE Tools\wce420\POCKET PC 2003\Target`. Например, для создания установочных файлов, которые будут запускаться на эмуляторе, следует выбрать значение `/cpu ARMV4 X86`. Параметр `platform_label` определяет типы платформ, для каждой из которых будет создан файл. `cab`. Возможные значения этого параметра приведены в табл. 2.2.

**Таблица 2.2. Возможные значения  
параметра platform\_label**

<b>Платформа</b>	<b>platform_label</b>
Pocket PC 2003 device	2577
Pocket PC 2003 emulator	686
Pocket PC 2002 device	2577
Pocket PC 2002 emulator	686

# Mobile Developer Power Toys

Этот пакет дополнительных утилит может быть очень полезен при отладке, особенно если применять инструменты разработки, которые выпускались не компанией Microsoft, а сторонними производителями. В этом разделе будут подробно рассмотрены все основные утилиты этого пакета.

## ActiveSync Remote Display

Эта утилита позволяет вывести экран Pocket PC в окно настольного компьютера. В это окно передаются все движения мыши, при помощи чего имитируется работа стилуса, и клавиатурный ввод. Несмотря на то что при работе Remote Display происходит замедление работы, это очень полезная утилита, без которой иногда просто не обойтись. Утилита имеет режимы масштабирования изображения, поэтому экран Pocket PC отображается корректно даже на больших мониторах. Внешний вид окна программы показан на рис. 2.8.

# ActiveSync Remote Display

Эта утилита  
В это время  
клавиатура  
Несмотря на  
(примерно  
без рук  
поэтому  
Внешний



сти  
ДВИ  
мул  
те  
нь I  
мы  
браз  
ока:

Рис. 2.8. Окно ActiveSync Remote Display.

## СЕСору

Эта утилита командной строки предназначена для копи-

рования файлов и каталогов на устройство, подключенное в данный момент к ActiveSync. Учитывая то, что сам пакет ActiveSync имеет средство для визуального копирования файлов, эта утилита вряд ли может быть названа самой полезной из состава Mobile Developer Power Toys.

Синтаксис запуска утилиты приведен ниже.

```
SeCopy [ключи командной строки] <источник> <приемник>
```

Ключ командной строки /is указывает утилите, что необходимо перезаписывать имеющиеся файлы, а ключ /s заставляет в процессе копирования учитывать подкаталоги, копируя информацию с воспроизведением иерархической структуры вложенности. Так, команда `secopy /s *.dll dev:\windows` копирует все файлы с расширением. dll из текущего каталога в каталог windows на Pocket PC.

## **Convert\_PPC\_DAT\_to\_SP\_XML**

Утилита `Convert_PPC_DAT_to_SP_XML` позволяет из текущего файла. cab для Pocket PC генерировать файл для создания установочного файла, который можно использовать на смартфонах.

## **Hopper**

Этот инструмент позволяет воспроизвести на устройстве ситуацию экстремального пользовательского ввода в разных режимах и довести при помощи этого краш-теста устройство до зависания или перезагрузки. При этом программа ведет подробный файл отчета и в момент краха устройства снимает копию экрана и сохраняет ее. Также в лог-файле сохраняются текущие параметры распределения памяти.

Программа позволяет измерить устойчивость приложения при большом потоке сигналов пользовательского ввода и время реакции приложения на пользовательский ввод данных.

Hopper имитирует нажатия стилусом на управляющих элементах окон и ввод данных с программной клавиатуры устройства.

## **JShell**

Это весьма полезная программа для отладки приложений на устройстве. Эта утилита выводит на экран список процессов, загруженных модулей и распределения памяти устройства. Внешний вид запущенной программы показан на рис. 2.9.

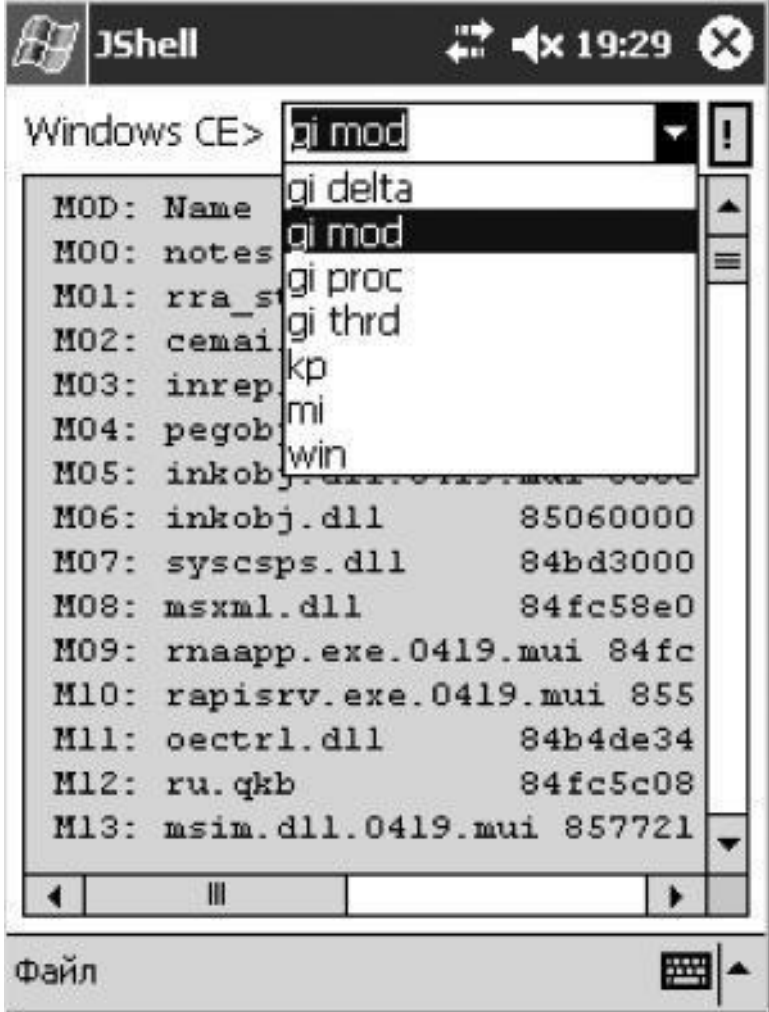


Рис. 2.9. Окно программы JShell.

Для того чтобы запустить программу, ее необходимо скопировать на устройство и затем запустить с соответствующими ключами командной строки или без них. Ключи командной строки приведены в следующем списке.

◆ Ключ -r <cmd> позволяет при запуске один раз выполнить команду <cmd>.

◆ Ключ -f заставляет приложение выводить информацию каждые 60 с, запуская программу с приоритетом реального времени.

◆ Ключ -m активирует утилиту mi и выводит результаты в файл отчета каждые 4 часа. Файл отчета записывается с именем miXXX.txt в каталог \My Documents, где XXX это номер файла-отчета.

◆ Ключ -w позволяет запустить команду win и записывать результаты в файл отчета каждые 4 часа. Файл отчета записывается с именем winXXX.txt в каталог \My Documents, где XXX это номер файла-отчета.

Когда вы запускаете программу без ключей командной строки, то она запускается с графическим интерфейсом пользователя (см. рис. 2.9), и можно вручную запустить одну из команд. Для того чтобы команда была выполнена, необходимо нажать кнопку «!». Результат выполнения команды выводится в окно JShell.

Предназначение всех команд приведено в следующем списке.

- ◆ Команда `win` выводит список всех окон.
- ◆ Команда `gi proc` отображает список всех активных в системе процессов.
- ◆ Команда `кр #` позволяет завершить выполнение процесса с номером, указанным в качестве параметра `#`.
- ◆ Команда `gi thrd` позволяет отобразить список потоков.
- ◆ Команда `gi mod` позволяет вывести список модулей (DLL).
- ◆ Команда `gi mod «xx»` позволяет вывести список модулей (DLL), имена которых начинаются с подстроки `xx`.
- ◆ Команда `mi` отображает информацию о распределении памяти.
- ◆ Команда `mi full` отображает подробную информацию о распределении памяти.
- ◆ Команда `s <app> <args>` создает процесс, полное имя к файлу которого передается в параметре `<app>`.

## PPC\_Command\_Shell

Это простой командный интерпретатор, выводящий на экран консольное окно (рис. 2.10).

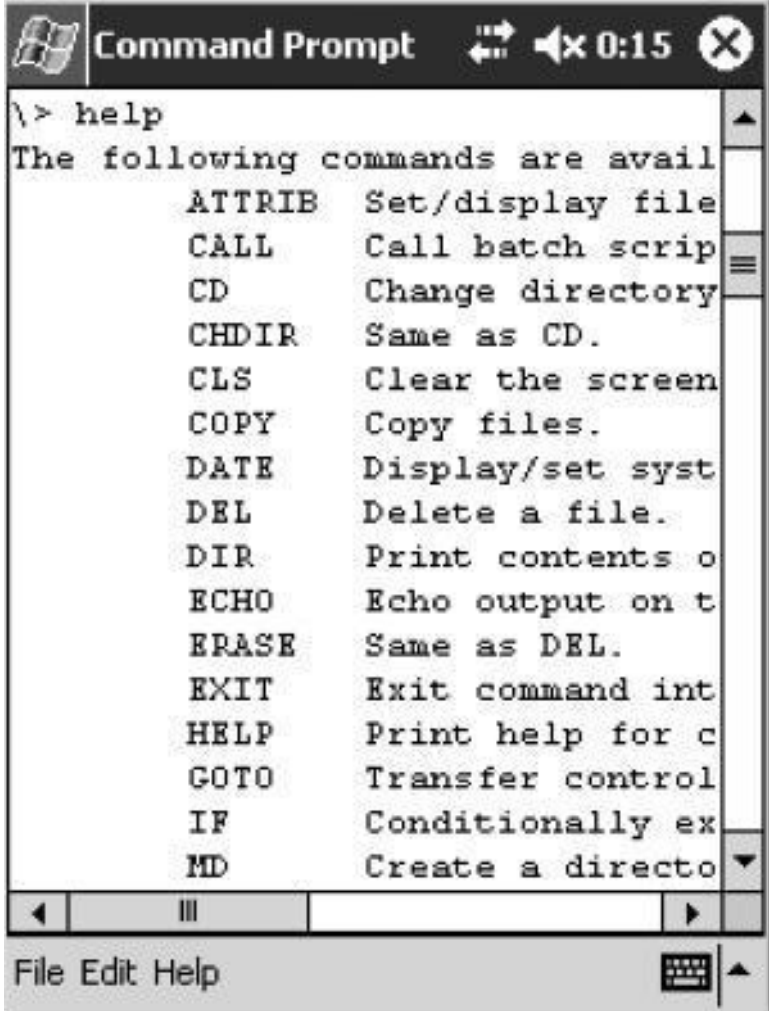


Рис. 2.10. Окно командного интерпретатора.

Содержимое каталога arm необходимо скопировать в каталог Windows на самом Pocket PC и затем нужно запустить файл cmd.exe. Список возможных команд выводится на экран, если набрать команду help и нажать клавишу Enter.

## **RAPI\_Debug**

Эта утилита состоит из двух частей. Программу bugtrap.exe необходимо скопировать на устройство в каталог Windows, а программу RapiDbg.exe – запустить на рабочей станции. Естественно, при отладке надо установить соединение при помощи ActiveSync. После запуска на рабочей станции программа RapiDbg.exe свяжется с устройством и отобразит на экране отладочную информацию (распределение памяти, запущенные процессы и потоки), и останется в режиме ожидания, отслеживая и отображая весь пользовательский ввод, который будет происходить на Pocket PC. Окно программы на рабочей станции показано на рис. 2.11.

```
C:\WINDOWS\System32\cmd.exe - RAPI00G
MODULE LOAD commctl.dll PID 0x2554E662 BASE 0x02DE0000
MODULE LOAD oleaut32.dll PID 0x2554E662 BASE 0x027D0000
MODULE LOAD ossvcs.dll PID 0x2554E662 BASE 0x02760000
MODULE LOAD avgshell.dll.0419.mui PID 0x2554E662 BASE 0x7FF30000
MODULE LOAD avgshell.dll PID 0x2554E662 BASE 0x02D70000
MODULE LOAD ws2_0.dll PID 0x2554E662 BASE 0x03680000
MODULE LOAD ole32.dll PID 0x2554E662 BASE 0x02800000
MODULE LOAD coredll.dll.0419.mui PID 0x2554E662 BASE 0x7FFE0000
MODULE LOAD coredll.dll PID 0x2554E662 BASE 0x01F60000
MODULE LOAD ceshell.dll PID 0x2554E662 BASE 0x02DA0000
MODULE LOAD uiocom.dll PID 0x2554E662 BASE 0x038C0000
MODULE UNLOAD PID 0x2554E662 BASE 0x7FED0000
MODULE UNLOAD PID 0x2554E662 BASE 0x020A0000
MODULE UNLOAD PID 0x2554E662 BASE 0x03B00000
MODULE UNLOAD PID 0x2554E662 BASE 0x02C30000
MODULE UNLOAD PID 0x2554E662 BASE 0x02A70000
MODULE UNLOAD PID 0x2554E662 BASE 0x01AD0000
MODULE UNLOAD PID 0x2554E662 BASE 0x02B00000
MODULE UNLOAD PID 0x2554E662 BASE 0x02A90000
MODULE UNLOAD PID 0x2554E662 BASE 0x02BB0000
MODULE UNLOAD PID 0x2554E662 BASE 0x01B60000
MODULE UNLOAD PID 0x2554E662 BASE 0x02F50000
MODULE UNLOAD PID 0x2554E662 BASE 0x03680000
PROCESS EXIT PID 0x2554E662 PPR0C 0x820D86A8 PTHREAD 0x857E95F8
```

Рис. 2.11. Окно программы с выводом отладочной информации.

## RAPI\_Start

Эта консольная программа позволяет с рабочей станции запустить программу на Pocket PC, если установлено соединение ActiveSync.

К примеру, на рабочей станции в консоли можно выполнить команду RAPI\_Start iexplore.exe <http://yandex.ru/>, а на Pocket PC будет запущен Pocket Explorer и загружена первая страница поисковой системы.

## TypeIt

Эта программа предназначена исключительно для смартфонов. После запуска она передает на смартфон кла-

виатурный ввод с рабочей станции.

# Общие вопросы создания интерфейса пользователя

Вся остальная книга будет посвящена именно программированию, поэтому в данном разделе будут кратко освещены общие вопросы, связанные с созданием интерфейса пользователя для Pocket PC.

Главным фактором, с которым приходится считаться при создании пользовательского интерфейса для программ, запускаемых на мобильных устройствах, является размер. Экран Pocket PC по площади на порядок меньше, чем экран современного настольного компьютера. Тем не менее функционально приложения для Pocket PC часто сравнимы по функциональности с приложениями для рабочей станции. Значит, органов настройки и управления должно быть столько же, а разместить их надо на куда меньшей площади.

Часто при решении этого конфликта между стремлением к реализации серьезной функциональности и малым размером экрана в жертву приносится размер элементов управления и разборчивость их обозначений. Мне запомнилось, как я установил на свой Pocket PC триальную версию одного из коммерческих органайзеров и удалил ее после первого же запуска. Причина столь резкого отказа от «мощной функциональности планирования и организации множества задач в одной программе» была проста. Органайзер нужен пользо-

вателю для облегчения жизни, а не для усложнения ее. В ситуации, когда пользователю приходится разбираться, на какую из вкладок, идущих в два ряда с двух сторон экрана надо нажать, чтобы получить нужный результат, любой из нас предпочел бы стандартный дизайн и возможности Microsoft Outlook. Именно поэтому Грег Кейзер (Greg Keyser), руководитель группы разработки мобильных устройств корпорации Microsoft, посвятил свое выступление на конференции Microsoft Mobile DevCon 2004 вопросам создания пользовательского интерфейса. Основные его советы приведены ниже.

1. Выводите на экран только ту информацию, которая нужна пользователю для выполнения текущей операции. Обычно отображается гораздо больше информации, чем нужно пользователю в данный конкретный момент. Примером может быть записная книжка «Контакты». Когда вы смотрите на список контактов, попытка отобразить в нем для каждого контакта ВСЮ информацию закончилась бы плохо даже в настольном компьютере, а на Pocket PC в действие вступает закон минимализма, который требует показывать только то, что крайне необходимо (рис. 2.12).

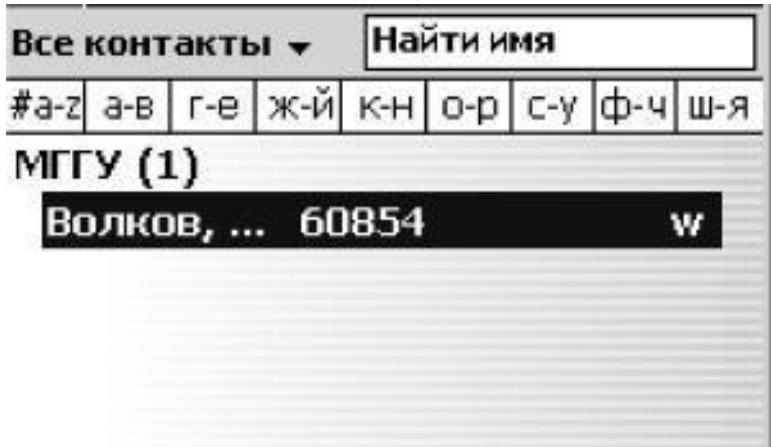


Рис. 2.12. Список «Контакты».

Поскольку этот список чаще всего вызывают, чтобы освежить в памяти телефон, то на первую страницу списка выводится имя и телефон (или имя и адрес электронной почты).

2. Ограничив выводимую информацию в объеме, оставьте пользователю возможность самому определить, что конкретно (например, телефон или адрес электронной почты) будет отображено на экране.

3. Пользователь должен затрачивать как можно меньше усилий для ввода данных.

- Следует группировать наиболее важные поля ввода так, чтобы они располагались первыми в списке.
- Поскольку для ввода в Pocket PC используется про-

граммная клавиатура, нужно располагать поля ввода таким образом, чтобы они не были скрыты выведенной на экран клавиатурой.

- Если возможно, следует предлагать пользователю выбрать значение из списка, а не вводить его в поле.
- Следует минимизировать необходимые для редактирования перемещения курсора и нажатия стилусом.

4. При создании интерфейса пользователя нужно привязывать все элементы управления к узлам сетки с шагом 4x4 пиксела.

5. Левое и правое поля страницы должны занимать не менее 8 пикселей.

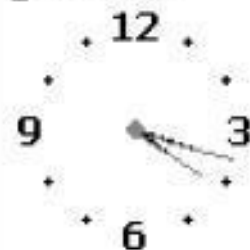
6. Элементы управления в высоту должны быть не менее 20 пикселей.

7. При создании окон настройки (рис. 2.13) следует придерживаться определенных правил.



Часы

Дома

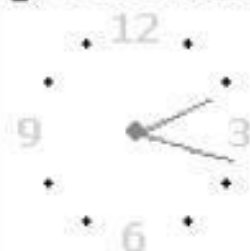


GMT+3 Москва

4 :18:49

03.10.2005

В поездке



GMT+1 Париж, Ма,

2 :18:49

03.10.2005

Время

Сигналы



## Рис. 2.13. Типичное окно настройки.

- Не стоит применять меню.
- Все органы управления должны уместиться в пределах окна. Следует избегать использования линеек прокрутки.
- Не следует применять более трех вкладок.
- Нужно оставить справа поле размером не менее 8 пикселей.

8. Помните, что при работе с контактным экраном владелец устройства может использовать не только стилус, но и пальцы.

- Области для нажатия стилусом или пальцами должны быть контрастны и иметь хорошо видимую границу.
- Область для нажатия стилусом должна занимать не менее 5 пикселей по горизонтали и вертикали.
- Область для нажатия пальцем должна занимать не менее 10 пикселей по горизонтали и вертикали. На рис. 2.14 показано окно калькулятора, работа с которым возможна и без стилуса.

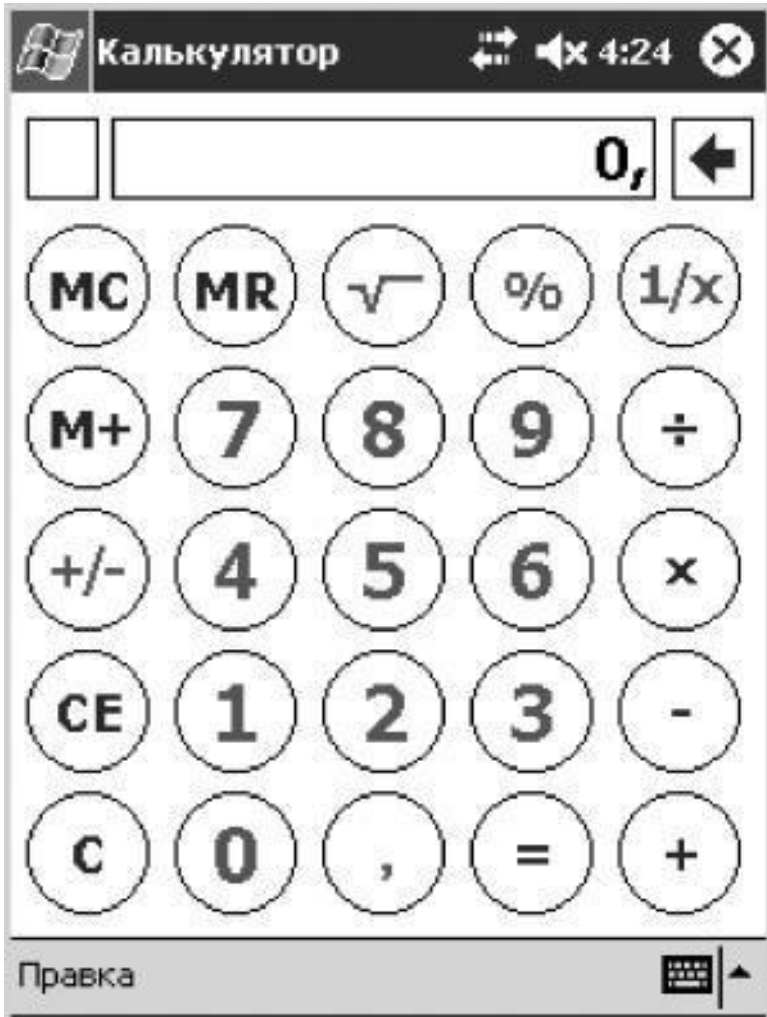


Рис. 2.14. Окно калькулятора.

9. При создании надписей не следует использовать наклонный шрифт, а жирный шрифт можно применять только для выделения заголовков или разделов.

10. Нужно помнить о возможности перевода приложения на другой язык. Следует учитывать, что длина терминов на другом языке может быть больше или меньше, чем на русском или английском языке.

На этом можно закончить вводную часть и перейти к созданию своих первых приложений для Pocket PC.

# **Глава 3**

## **Разработка программ для Pocket PC с помощью Microsoft eMbedded Visual Basic 3.0**

### **Среда разработки eMbedded Visual Basic 3.0**

Для краткости в этой главе для обозначения среды eMbedded Visual Basic 3.0 будет использовано сокращение eVB.

Запустить eVB можно либо при помощи команды меню Пуск (Пуск ► Программы ► Microsoft eMbedded Visual Tools ► eMbedded Visual Basic 3.0), либо найти в папке установки исполняемый файл программы (при стандартной установке это будет C:\Program Files\Microsoft eMbedded Tools\EVB\EVB3.EXE). При запуске на экран будет выведено окно создания проекта (рис. 3.1).

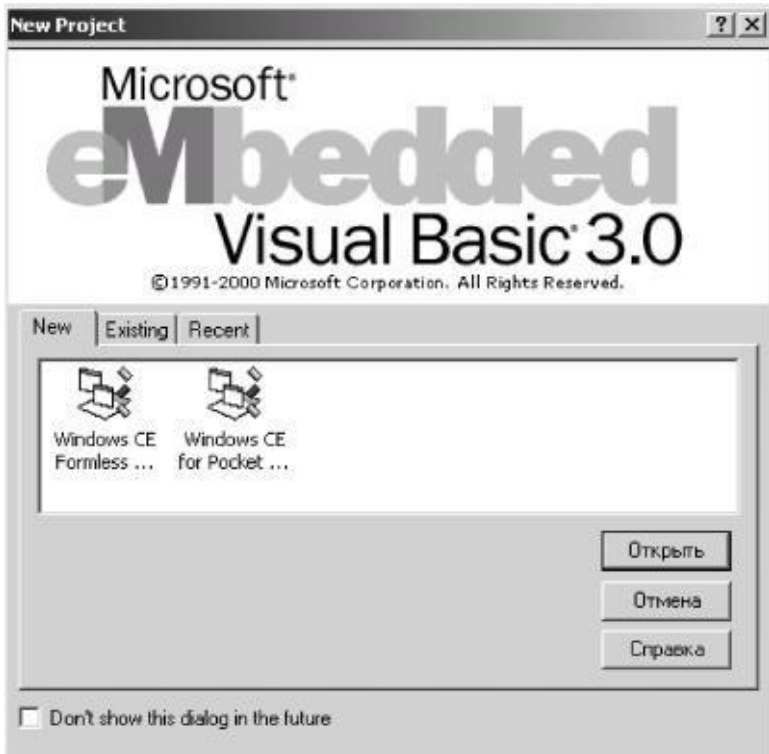


Рис. 3.1. Окно выбора типа создаваемого проекта.

В этом окне необходимо выбрать пиктограмму Windows CE for PocketPC и нажать кнопку Открыть. Это приведет среды разработки в состояние работы со «свежим» проектом, содержащим одну чистую форму. Следует подробно рассмотреть структуру экрана среды и составляющие ком-

поненты.

Среда eVB по умолчанию состоит из одного основного окна, в верхней части которого находится главное меню и панель инструментов (рис. 3.2).



Рис. 3.2. Главное меню и панель инструментов Стандартная.

Внутри основного окна может располагаться несколько окон дизайнера форм и панелей инструментов. При этом только позиция главного меню является неизменной. Все остальные панели инструментов и окна дизайнеров внутри основного окна можно свободно перемещать, либо переводя их к той или иной стороне окна, либо оставляя их в «плавающем» состоянии (рис. 3.3).

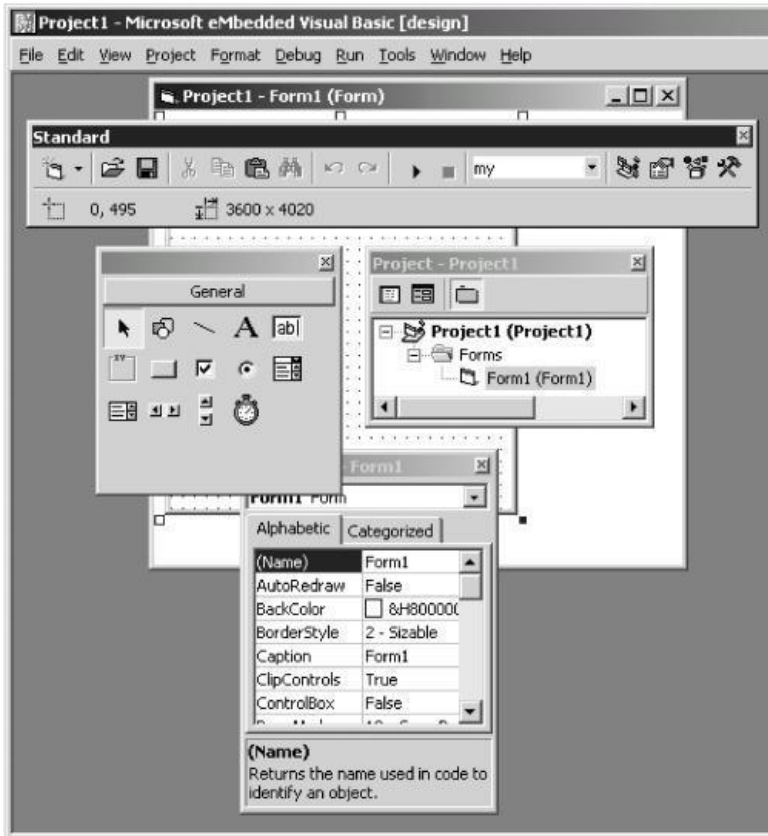


Рис. 3.3. Плавающие панели и окна.

Перемещение окон дизайнеров производится при помощи захвата мышью за заголовки окна, а перемещение панелей инструментов за специальный элемент в левой части па-

нели.

## ***ВНИМАНИЕ!***

*В случае, когда вам более привычна работа со множеством окон, не связанных между собой основным окном (SDI—интерфейс), вы можете переключиться в этот режим, установив соответствующий флажок в основном меню **Tools # Options... # General # SDI Development Environment**.*

## **Настройка панелей инструментов**

Доступ к меню настройки панелей инструментов осуществляется при помощи выполнения команды **Toolbars** из меню **View**. Это же меню можно вызвать щелчком левой кнопки мыши на любой из панелей инструментов (рис. 3.4).

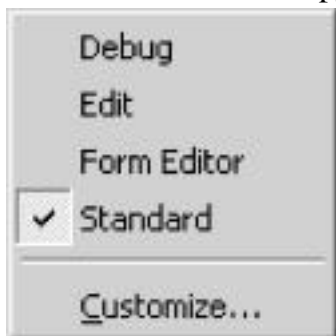


Рис. 3.4. Меню **Toolbars**.

Назначение команд этого меню приведено в соответствующем списке.

- ◆ Команда **Debug** отображает панель с инструментами отладки.

- ◆ Команда **Edit** отображает панель с инструментами редактирования.

- ◆ Команда **Form Editor** отображает панель с инструментами дизайнера форм.

- ◆ Команда **Standard** отображает панель со стандартным набором инструментов.

- ◆ Команда **Customize** выводит на экран окно настройки панелей инструментов (рис. 3.5).

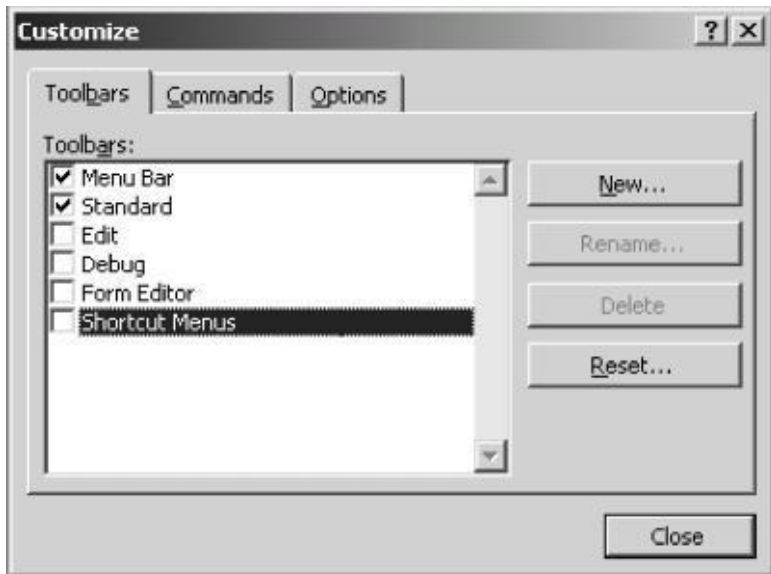


Рис. 3.5. Окно настройки панелей инструментов.

Вкладка Toolbars позволяет отобразить те же панели инструментов, что и меню настройки. Следует обратить внимание, что в списке присутствует еще одна панель Shortcut Menus, содержащая в себе все контекстные меню среды разработки.

При помощи кнопки New разработчик может создать собственную панель инструментов, на которой он соберет именно те инструменты, которыми пользуется чаще всего.

## Упражнение 3.1. Создание и настройка панели инструментов

1. Выполнить команду View ► Toolbars ► Customize. На экран будет выведено окно, изображенное на рис. 3.5.
2. Нажать кнопку New.
3. В строке ввода указать значение MyTV.
4. Нажать кнопку Close. Теперь нужно отыскать на экране созданную панель. Сначала это может вызвать определенное затруднение, так как панель довольно мала, поскольку инструментов на ней пока нет (рис. 3.6).



Рис. 3.6. Созданная панель инструментов.

5. Переместить созданную панель вправо или влево от центра, чтобы она не скрывалась выводимыми на экран диалоговыми окнами.
6. Снова вывести на экран окно Customize. В этом окне следует открыть вкладку Commands. Эта вкладка содержит в себе перечень всех пунктов меню и команд, соответствующих этим пунктам. Несмотря на то что изменить этот пе-

речень невозможно, вкладка позволяет настроить любую из панелей инструментов, поместив на нее инструменты из левой части окна. Именно это и нужно сделать, перетащив на панель MyTV несколько пиктограмм (рис. 3.7).

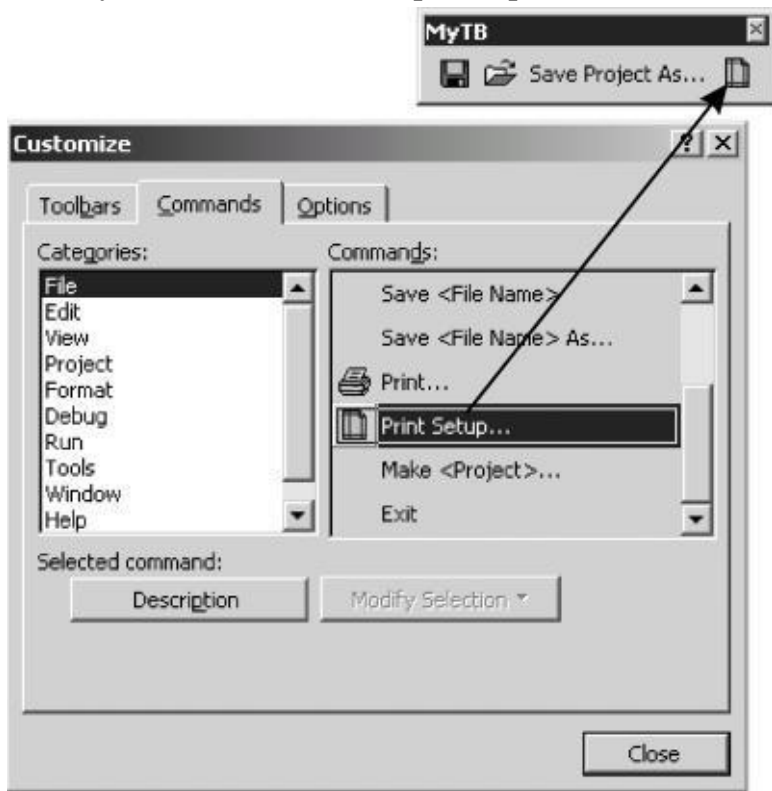


Рис. 3.7. Перетаскивание инструментов на панель.

После того как на панель будут перемещены пиктограммы необходимых команд, среда разработки позволит изменить параметры созданной панели и каждого инструмента на ней. Для этого достаточно выбрать любой из инструментов панели, после чего в окне Customize станет активной кнопка Modify Selection. Нажатие этой кнопки выводит на экран список команд (рис. 3.8).

Reset

Delete

---

Name: &Open Project...

---

Copy Button Image

Paste Button Image

Reset Button Image

Edit Button Image...

Change Button Image ▶

---

Default Style

Text Only (Always)

Text Only (in Menus)

Image and Text

---

Begin a Group

Modify Selection ▼

### Рис. 3.8. Команды настройки инструмента.

Назначение команд рассматривается в следующем списке.

- ◆ Команда **Reset** возвращает все настройки выбранного инструмента в исходное состояние.
- ◆ Команда **Delete** удаляет с панели выбранный инструмент.
- ◆ Команда **Name** позволяет изменить имя выбранного инструмента.
- ◆ Команда **Copy Button Image** копирует пиктограмму выбранного инструмента в буфер обмена.
- ◆ Команда **Paste Button Image** использует рисунок из буфера обмена в качестве пиктограммы выбранного инструмента.
- ◆ Команда **Reset Button Image** возвращает первоначальную пиктограмму выбранного инструмента.
- ◆ Команда **Edit Button Image** позволяет отредактировать пиктограмму выбранного инструмента. При этом используется простой графический редактор.
- ◆ Команда **Change Button Image** позволяет выбрать пиктограмму из дополнительного набора рисунков.
- ◆ Команда **Default Style** назначает инструменту сочетание рисунка и текста по умолчанию.
- ◆ Команда **Text Only (Always)** позволяет указать, что в качестве пиктограммы инструмента всегда будет использоваться только текст с именем инструмента.

◆ Команда Text Only (In menus) указывает, что в качестве пиктограммы в меню будет выводиться только текстовое наименование, а на панели инструментов используется установленное по умолчанию значение.

◆ Команда Image and Text указывает, что в качестве пиктограммы инструмента всегда будут выводиться как рисунок, так и текст.

◆ Команда Begin a Group указывает, что перед выбранным инструментом на панели будет размещена вертикальная черта, начинающая новую группу.

Таким образом, в этом упражнении была создана настраиваемая панель. Пользователь может расположить ее на экране в удобном для него месте или даже закрыть, если панель больше ему не нужна.

Следующая вкладка окна Customize называется Options и содержит команды, которые перечислены в следующем списке.

◆ Команда Large icons включает режим отображения больших пиктограмм.

◆ Команда Show ScreenTips on Toolbars включает отображение всплывающих подсказок для команд.

◆ Команда Show ShortCut Key on ScreenTips включает отображение горячих клавиш на ярлычках всплывающих подсказок.

◆ Команда Menu Animations позволяет выбрать режим анимации при раскрытии меню.

# Панель инструментов Standard

На этой панели располагаются самые простые инструменты, которые применяются именно для редактирования исходного кода и работы с файлами проекта.



◆ Кнопка позволяет произвести добавление к проекту новой формы, модуля или пользовательского файла.



◆ Кнопка отвечает за открытие файла.



◆ Кнопка позволяет сохранить открытый файл на диск.



◆ Кнопки осуществляют стандартные операции редактирования – вырезание, копирование и вставка из буфера.



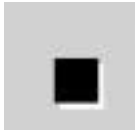
◆ Кнопка выводит на экран диалоговое окно поиска и замены, позволяющее гибко настроить параметры текстового поиска и замены как в текущем файле, так и во всем проекте.



◆ Кнопки выполняют стандартные функции отмены выполненного действия (Undo) и возврата отмененного ранее действия (Redo).



◆ Кнопка позволяет запустить проект в режиме отладки. В этом режиме возможна остановка приложения в заранее установленных точках остановки, пошаговое выполнение программы, отслеживание значений переменных и их изменение.



◆ Кнопка останавливает выполнение программы, запущенной предыдущей командой.



◆ Список позволяет выбрать платформу, на которую будет загружена программа.



◆ Кнопка выводит на экран окно Project Explorer со структурой файлов проекта.



◆ Кнопка отображает на экране окно настройки свойств выделенного объекта.



◆ Кнопка выводит на экран окно навигатора объектов.



◆ Кнопка отображает на экране окно палитры компонентов.

## **Общие свойства окон дизайнеров**

При щелчке правой клавишей мыши на любом из окон, располагающихся внутри основного окна eVB, на экран выводится контекстное меню. Поскольку меню контекстное, то команды в нем для каждого окна будут разными. Тем не менее существуют и совпадающие команды, которые имеют одно и то же назначение для всех окон.

Как уже было сказано выше, служебные окна можно располагать на экране в произвольном порядке. Команда Docable управляет поведением окна при перемещении его по экрану. Если флажок Docable установлен, то окно «прилипает» к другим окнам и сторонам экрана.

Команда Hide скрывает окно. Вывести окно на экран после его скрытия можно при помощи соответствующей команды пункта меню View.

## **Использование и настройка палитры компонентов (Toolbox)**

Палитра компонентов по умолчанию расположена в левой части экрана (рис. 3.9).



Рис. 3.9. Палитра компонентов eVB.

Палитра компонентов предназначена для выбора и размещения на форме элементов управления. Ненастроенная панель компонентов содержит на себе только «родные» элементы управления eVB, которые располагаются на единственной вкладке General. Набор этих компонентов невелик, и приложение со сложной функциональностью построить из них просто невозможно. В eVB нет возможности создавать элементы управления ActiveX, но зато можно исполь-

зовать сторонние компоненты ActiveX, написанные на других языках. Совместное использование сторонних компонентов ActiveX и встроенных элементов управления, а также возможность вызова из приложений eVB функций Windows API позволяет создавать полнофункциональные приложения любого назначения.

## Упражнение 3.2. Настройка палитры компонентов

1. Щелкнуть на палитре компонентов правой клавишей мыши. На экран будет выведено контекстное меню (рис. 3.10).

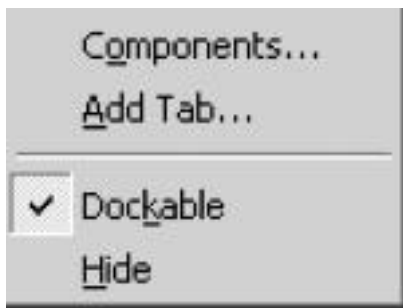


Рис. 3.10. Контекстное меню палитры компонентов.

2. В контекстном меню необходимо выполнить команду Add Tab. В выведенной на экран строке ввода нужно указать значение ActiveX, после чего следует нажать кнопку ОК.

Кнопка ActiveX будет добавлена к панели и ее можно будет найти в самой нижней части палитры компонентов.

3. После нажатия кнопки ActiveX будет активирована новая страница палитры. Эта страница, разумеется, пока пустая. Чтобы добавить на нее компоненты, нужно выполнить команду контекстного меню Components. По этой команде на экран будет выведено диалоговое окно добавления компонентов в палитру (рис. 3.11).

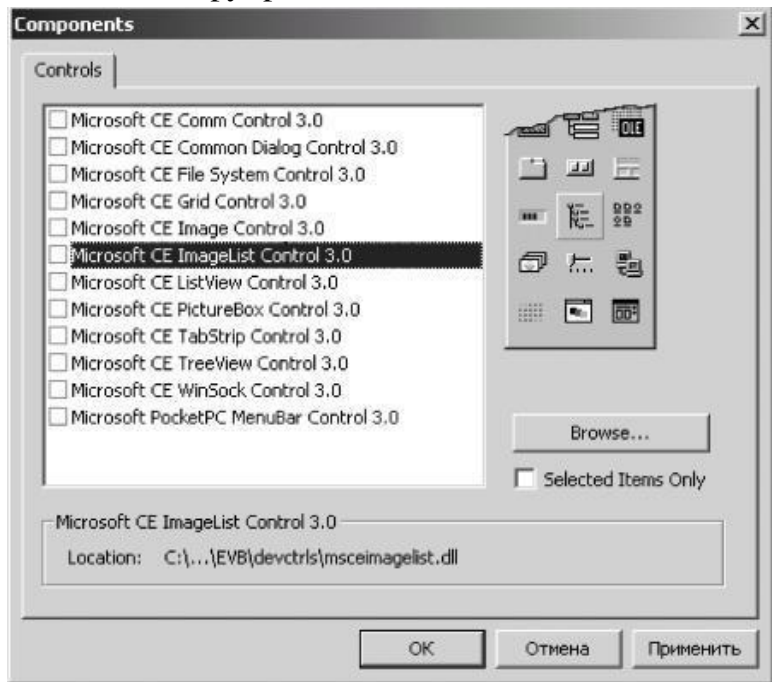


Рис. 3.11. Окно добавления компонентов в палитру.

В списке Controls отображаются все компоненты, установленные в данный момент. Чтобы расширить этот список и добавить дополнительные элементы управления, нужно нажать кнопку Browse и выбрать соответствующий файл, который содержит компоненты, предназначенные для применения в среде Windows CE.

4. Нужно выбрать несколько компонентов из списка, взведя флажок слева от имени компонента в списке Controls. После нажатия кнопки ОК выбранные компоненты будут добавлены в палитру.

5. Компоненты можно перемещать с одной страницы палитры на другую, просто перетаскивая их мышью.

6. Созданную страницу палитры компонентов можно переименовать или удалить, используя соответствующие команды Rename Tab и Delete Tab из контекстного меню, которое появляется при щелчке правой кнопкой мыши на названии палитры.

### ***ПРИМЕЧАНИЕ.***

*Страница палитры создается и присутствует во всех проектах, а установленные компоненты ActiveX только в том проекте, для которого производилась установка. В новый проект устанавливать эти компоненты придется заново.*

# Помещение компонентов на форму

Чтобы разместить на форме компонент с палитры, необходимо выполнить достаточно простую последовательность действий.

1. Щелкнуть мышью на выбранном компоненте в палитре. При этом изображение компонента будет «утоплено» в палитру.

2. Щелкнуть мышью на форме (курсор примет крестообразную форму) и, не отпуская левой клавиши мыши, переместить курсор по диагонали. По мере протягивания курсора будет отображаться рамка, обозначающая размеры устанавливаемого компонента.

3. После отпущения левой клавиши мыши компонент будет размещен на форме. При этом будут использованы те размеры, которые были заданы на предыдущем этапе.

Чтобы разместить на форме несколько одинаковых компонентов, при выделении компонента на палитре надо удерживать в нажатом состоянии клавишу **Ctrl**. В этом случае после размещения первого компонента на форме и отпущения левой клавиши мыши указатель мыши по-прежнему будет иметь крестообразную форму, и разработчик сможет разместить следующие компоненты, не выбирая их каждый раз на палитре компонентов. После того как размещение компонентов будет завершено, необходимо на палитре компонен-

тов щелкнуть на пиктограмме со стрелкой, чтобы выйти из режима множественного размещения компонентов.

Для удаления одного компонента необходимо выбрать его мышью и нажать клавишу Delete.

Выделенный щелчком мыши на форме компонент обрамляется якорями, которые выглядят как небольшие черные прямоугольники. Перетаскивая якоря, можно изменять размер компонента. Размер выделенного компонента также можно изменять при помощи клавиш перемещения курсора при нажатой клавише Shift. Перемещать компонент по форме можно как мышью, так и клавишами управления курсором при нажатой клавише Alt.

Операции редактирования компонентов, перемещения и изменения их размеров можно выполнять и над группой компонентов. Чтобы объединить компоненты на форме в группу, необходимо выделить их, щелкая поочередно на каждом из них мышью при нажатой клавише Shift.

## **Изменение размеров и выравнивание компонентов**

Для создания аккуратного дизайна помещенные на форму компоненты требуют определенной подгонки по размеру и точного выравнивания. Для этих целей в eVB есть ряд инструментов, объединенных в меню Format. Команды этого меню приведены в следующем списке.

◆ Команда **Align** управляет взаимным выравниванием компонентов на форме.

- **Lefts** – левые стороны выделенных компонентов будут выровнены по левой стороне первого компонента слева.
- **Centers** – вертикальные центры выделенных компонентов будут выровнены по центру первого компонента справа.
- **Rights** – правые стороны выделенных компонентов будут выровнены по правой стороне первого компонента справа.
- **Tops** – верхние стороны выделенных компонентов будут выровнены по верхней стороне первого компонента сверху.
- **Middles** – горизонтальные центры выделенных компонентов будут выровнены по центру первого компонента снизу.
- **Bottoms** – нижние стороны выделенных компонентов будут выровнены по нижней стороне первого компонента снизу.
- **To Grid** – левые верхние углы выделенных компонентов будут совмещены с ближайшим узлом координатной сетки. Эта команда имеет смысл только в том случае, если в настройках **Tools ► Options ► General** не взведен флажок **Align Controls to Grid**. Если этот флажок взведен, то привязка к узлам координатной сетки производится автоматически, в момент помещения компонента на форму или перемещения его по форме.

◆ Команда **Make same size** позволяет выравнивать размеры компонентов.

- **Width** – ширина выделенных компонентов приравняется к наибольшей ширине из группы.
- **Height** – высота выделенных компонентов приравняется к наибольшей из группы.
- **Both** – выравнивает как ширину, так и высоту выделенных компонентов.
- ◆ Команда **Size to Grid** совмещает все углы выделенного компонента с ближайшими узлами координатной сетки.
- ◆ Команда **Horizontal Spacing** позволяет выравнивать горизонтальные расстояния между компонентами.
- **Make Equal** – располагает выделенные компоненты на равном расстоянии друг от друга. При этом должно быть выделено не менее трех компонентов.
- **Increase** – увеличивает расстояние между выделенными компонентами по горизонтали.
- **Decrease** – уменьшает расстояние между выделенными компонентами по горизонтали.
- **Remove** – делает горизонтальное расстояние между компонентами равным нулю.
- ◆ Команда **Vertical Spacing** позволяет уравнивать вертикальные расстояния между компонентами. Содержит тот же набор подкоманд, что и **Horizontal Spacing**.
- ◆ Команда **Center in form** предназначена для центровки компонентов. Две подкоманды располагают по вертикальному и горизонтальному центру формы выделенный компонент или группу компонентов. Следует обратить внимание

на то, что при расположении по центру формы группы компонентов расстояния между компонентами остаются неизменными.

◆ Команда **Order** управляет расположением компонентов в слоях.

- **Bring to front** – выводит выбранный компонент или их группу в верхний слой.

- **Send to Back** – перемещает выбранный компонент или их группу в нижний слой.

◆ Команда **Lock Controls** делает компоненты недоступными для перемещения и изменения размеров при помощи мыши.

## **Упражнение 3.3. Создание простого приложения**

1. На форме нужно разместить три компонента **Shape**, один компонент **Label**, одну кнопку и компонент **Timer** так, как это показано на рис. 3.12.

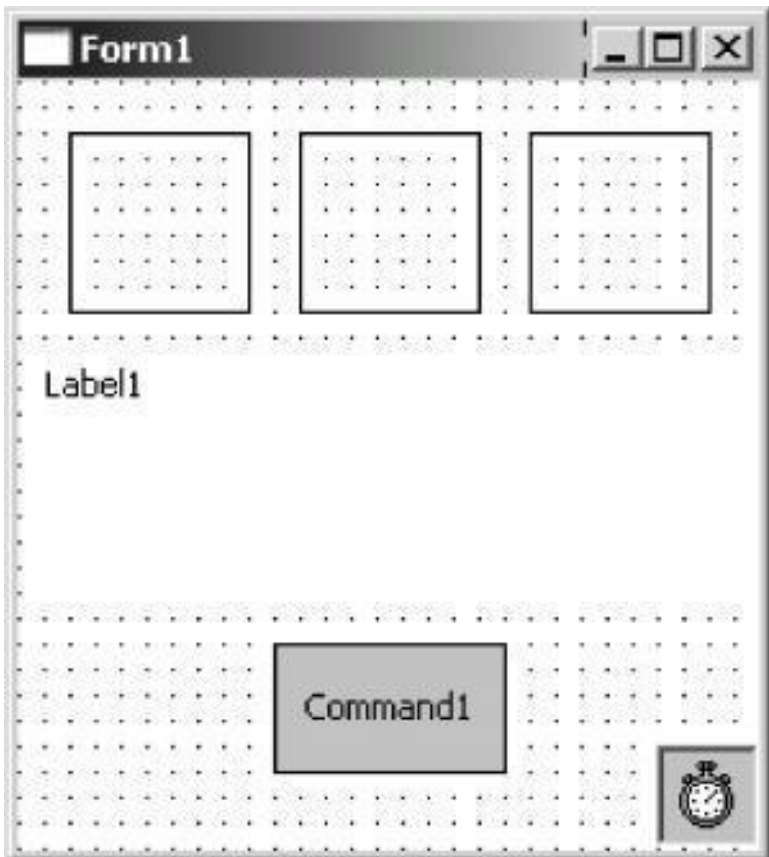


Рис. 3.12. Создание формы приложения.

2. Двойным щелчком на кнопке Command1 создайте обработчик события щелчка на кнопке. Окно с кодом будет открыто автоматически и шаблон обработчика данного собы-

тия будет создан. В этом шаблоне нужно ввести код, приведенный в листинге 3.1.

### **Листинг 3.1**

```
Private Sub Command1_Click()  
Label1.Font.Size = 20  
Label1.Font.Bold = True  
Label1.WordWrap = True  
Label1.Alignment = vbCenter  
Label1.Caption = «Здравствуй, Мир!»  
Timer1.Interval = 300  
Timer1.Enabled = True  
Shape1.FillColor = &HFF&  
Shape2.FillColor = &HFF&  
Shape3.FillColor = &HFF&  
End Sub
```

3. Двойным щелчком на компоненте Timer1 нужно создать обработчик события срабатывания таймера и ввести код из листинга 3.2.

### **Листинг 3.2**

```
Private Sub Timer1_Timer()
```

```
If (Shape1.FillStyle = 1) Then
Shape1.FillStyle = 0
Shape2.FillStyle = 0
Shape3.FillStyle = 0
Else
Shape1.FillStyle = 1
Shape2.FillStyle = 1
Shape3.FillStyle = 1
End If
End Sub
```

4. В списке устройств нужно выбрать эмулятор Pocket PC 2002 (рис. 3.13).

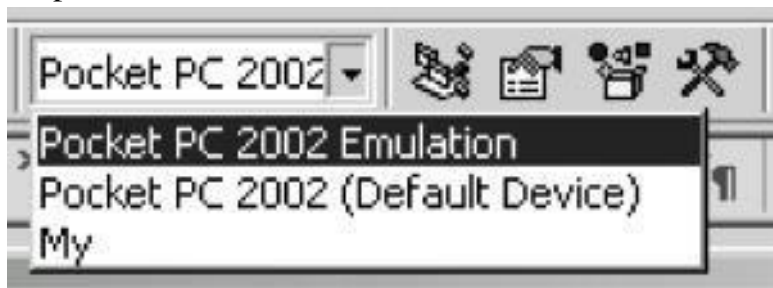


Рис. 3.13. Выбор устройства.

5. Выполнить команду меню Run ► Execute или использовать сочетание клавиш Ctrl+F5. Среда запустит эмулятор, загрузит в него программу и начнет ее выполнение.

6. В окне программы нужно нажать кнопку Command1. В результате будет выведена надпись «Здравствуй, Мир!», а три прямоугольника в верхней части окна будут мигать красным цветом (рис. 3.14).



Form1

3:38



**Здравствуй,  
Мир!**

Command1



Рис. 3.14. Результат выполнения программы.

## Настройки среды

Команда меню Tools ► Options выводит на экран окно с настройками среды eVB. Назначение элементов управления будет рассмотрено в этом разделе.

### Вкладка Editor

Эта вкладка позволяет настроить параметры редактора кода.

◆ Флажок Auto Syntax Check включает проверку синтаксиса после каждой введенной строки кода.

◆ При взводе флага Require Variable Declaration в каждый новый модуль будет добавлена строка Option Explicit, делающая имена переменных уникальными во всем пространстве имен проекта.

◆ Флажок Auto List Members включает автозавершение при наборе кода.

◆ Флажок Auto Quick Info включает отображение информации о функции и ее параметрах.

◆ Флажок Auto Data Tips включает вывод значений переменных во всплывающих ярлычках в пошаговом режиме отладки.

◆ Флажок Auto Indent включает автоматическую установ-

ку отступа первой строки кода.

- ◆ Поле Tab Width задает размер отступа в символах.

- ◆ Флажок Drag-and-Drop Text Editing включает возможность перетаскивания выделенного текста при помощи мыши.

- ◆ Флажок Default to Full Module View заставляет среду разработки для каждого нового модуля использовать режим отображения всего кода, а не только текущей процедуры.

- ◆ Флажок Procedure Separator включает отображение горизонтальной линии, отделяющей одну процедуру от другой.

## **Вкладка Editor Format**

На этой вкладке устанавливаются параметры отображения исходного кода программных модулей.

- ◆ Группа Code Colors позволяет установить цвета, применяемые в редакторе для того или иного элемента кода. Раздельно устанавливаются цвета шрифта (Foreground), закраски фона (Background) и индикатора (Indicator), выводимого на левое поле, если он предусмотрен для данного элемента кода.

- ◆ Блок Font отвечает за установку типа шрифта.

- ◆ Список Size позволяет задать размер шрифта.

- ◆ Флажок Margin indicator bar включает показ поля индикатора в окне редактирования.

## Вкладка General

На этой вкладке располагаются органы управления, которые позволяют задавать самые общие параметры среды разработки.

- ◆ Флажок **Show Grid** включает отображение координатной сетки на форме в режиме дизайна формы.

- ◆ Поле **Width** позволяет задавать горизонтальное расстояние между узлами координатной сетки. Допустимое значение располагается в интервале от 2 до 60 пикселей.

- ◆ Поле **Height** позволяет задавать вертикальное расстояние между узлами координатной сетки. Допустимое значение располагается в интервале от 2 до 60 пикселей.

- ◆ Флажок **Align Controls to Grid** включает автоматическое совмещение сторон помещенного на форму элемента с линиями координатной сетки.

- ◆ Флажок **Show ToolTips** включает отображение ярлычков подсказок при наведении курсора на элементы панелей инструментов, компоненты палитры компонентов и управляющие элементы окон дизайнеров.

- ◆ Флажок **Collapse Proj. Hides Windows** включает режим, который скрывает все окна проекта в среде разработки, когда в окне структуры проекта сам проект сворачивается до одной строки.

- ◆ Флажок **SDI Development Environment** позволяет активи-

вировать режим, при котором единое окно среды разработки отсутствует, а все дизайнеры и редактор кода работают в отдельных окнах.

## **Вкладка Docking**

Эта вкладка содержит список всех окон среды разработки. Установка или снятие флажка для определенного окна включает или, соответственно, выключает свойство Docable для данного окна.

## **Вкладка Environment**

На этой вкладке расположены органы управления, которые отвечают за общие параметры взаимодействия среды разработки с пользователем.

◆ Флажок Prompt for project включает вывод запроса на создание нового проекта при запуске среды eVB.

◆ Флажок Save Changes заставляет среду автоматически сохранять изменения в проекте при запуске программы на выполнение или отладку.

◆ Флажок Prompt To Save Changes заставляет среду запрашивать разрешения на сохранение изменений в проекте при запуске программы на выполнение или отладку.

◆ Флажок Do not Save Changes указывает, что не следу-

ет выполнять сохранение изменений в проекте при запуске программы на выполнение или отладку.

◆ Группа `Show Templates For` позволяет выбрать шаблоны, которые будут отображены в списке при использовании инструмента `Add`, который располагается на панели `Standard`.

◆ Поле `Templates Directory` позволяет указывать каталог, в котором располагаются файлы шаблонов.

# **Управление проектами. Файлы проекта. Компиляция и запуск программы**

В eVB все файлы, предназначенные для создания приложения, объединяются в проект. В отличие от таких сред, как Microsoft Visual Studio или Borland Delphi, среда программирования eVB не содержит средств управления группами проектов. Но работать с одним проектом она все же позволяет.

## **Управление файлами проекта**

Основные действия с файлами проекта осуществляются при помощи пункта меню File. Команды этого меню следует рассмотреть подробно.

◆ Команда New Project выводит на экран диалоговое окно создания нового проекта. В этом окне разработчик может указать тип создаваемого проекта. По умолчанию eVB предлагает не очень широкий выбор. Можно создавать приложение без формы или приложение, содержащее форму.

◆ Команда Open Project закрывает текущий проект, если он открыт в среде, и выводит стандартное диалоговое окно открытия проекта.

- ◆ Команда Save Project сохраняет на диск текущий проект и все его файлы.
- ◆ Команда Save Project As выводит на экран диалоговое окно, позволяющее сохранить текущий проект с новым именем и в новом месте.
- ◆ Команда Save сохраняет на диске текущий файл проекта.
- ◆ Команда Save As... выводит на экран диалоговое окно, позволяющее сохранить текущий файл проекта с новым именем и в новом месте.
- ◆ Команда Print выводит форму или исходный код на принтер с параметрами, выбранными при помощи команды Print Setup.
- ◆ Команда Print Setup выводит на экран стандартный диалог настройки принтера.
- ◆ Команда Make Project выводит на экран диалоговое окно, позволяющее выбрать место и имя для сохранения созданного исполняемого файла eVB, затем компилирует проект и создает файл с расширением. vb.
- ◆ Команда List of Recently-Used Projects отображает список проектов, открывавшихся последними.
- ◆ Команда Exit завершает работу со средой eVB.

## **Состав и назначение файлов проекта**

В этом разделе будут рассмотрены все типы файлов, ко-

торые входят в состав обычного проекта.

◆ Файл с расширением. ebr – это основной файл проекта, который содержит сведения о настройках проекта, входящих в проект формах и модулях, типе проекта. Это текстовый файл, который можно открыть и просмотреть в блокноте. Можно и отредактировать этот файл в блокноте, но в этом нет необходимости, поскольку все требуемые сведения корректно вносятся в этот файл средой разработки.

◆ Файл с расширением. ebf – это файл формы. Этот текстовый файл содержит в себе настройки формы, а также код процедур и объявлений, которые относятся к данной форме.

◆ Файл с расширением. frx предназначен для хранения дополнительных ресурсов формы. Этот бинарный файл содержит пиктограммы или другие бинарные данные, относящиеся к компонентам, размещенным на форме. Файл имеет то же имя, что и форма, для которой он создается.

◆ Файл с расширением. bas – это файл модуля. Текстовый файл хранит код процедур и объявлений, которые понадобилось вынести в отдельный модуль.

◆ Файл с расширением. osx – это бинарный файл, который содержит компонент ActiveX, не входящий в поставку eVB.

◆ Файл с расширением. vbw хранит состояние рабочего стола eVB. В этом текстовом файле указывается расположение и состояние форм и модулей проекта во время дизайна.

◆ Пользовательские файлы – звуковые, текстовые или

графические файлы, которые разработчик считает необходимым включить в состав проекта.

Для управления файлами, входящими в состав проекта, предназначен специальный инструмент Project Explorer, окно которого показано на рис. 3.15.

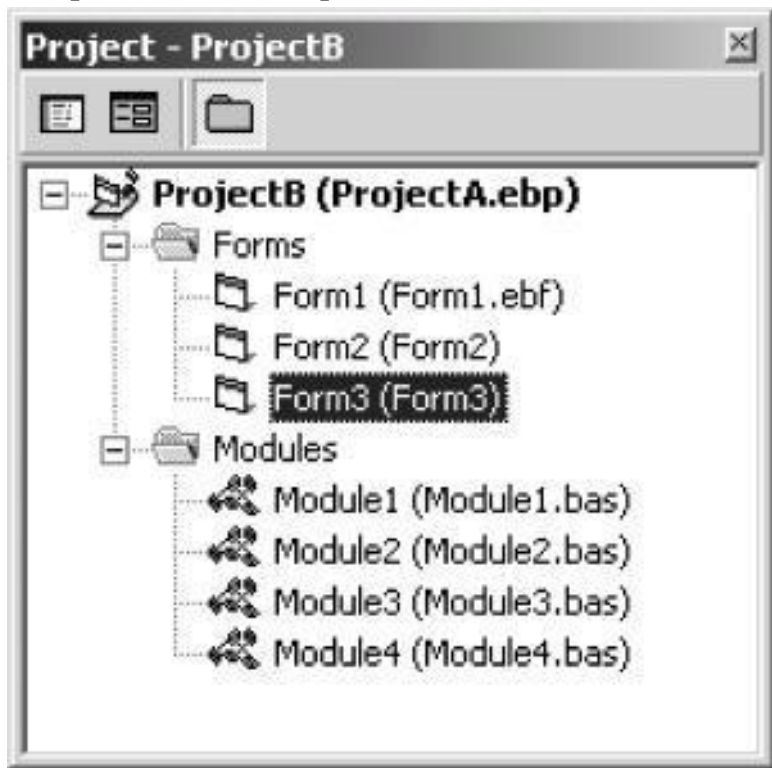


Рис. 3.15. Окно Project Explorer.

В основной части данного окна отображается список файлов, входящих в проект. В верхней части окна находятся три кнопки. Кнопка *Ш* позволяет переключиться на режим отображения кода выбранного в списке объекта.

Рядом с названием формы, проекта или модуля в списке указывается название файла, в котором хранятся данная форма, проект или модуль. Название файла и объекта могут не совпадать. На рисунке есть две формы (Form2 и Form3), которые были добавлены к проекту, но еще не сохранены на диск. Для них имена файлов отсутствуют.

В окне Project Explorer при щелчке на объекте выводится контекстное меню. Команды меню отличаются при выборе разных объектов. Следует обратить внимание на наиболее значимые моменты.

- ◆ Для форм можно использовать команды View Object и View Code, для модулей – только View Code.

- ◆ Для любого объекта присутствует команда Print, позволяющая вывести на печать как код всего проекта, так и отдельного модуля. При этом форма может быть выведена на печать в виде рисунка или в виде текстового файла.

- ◆ Для модулей и форм в контекстном меню присутствует команда Remove, позволяющая исключить объект из состава проекта.

# Компиляция и запуск программы

Язык Visual Basic не создает исполняемого кода. Результатом компиляции приложения является файл с расширением. vb. Этот файл является псевдокодом, то есть он не может быть запущен на устройстве, как обычный исполняемый файл. При запуске файла с расширением. vb на компиляторе или на Pocket PC сначала запускается программа-загрузчик, которая инициализирует среду исполнения VB и передает файл ей. Собственно говоря, файл с расширением. vb представляет собой последовательность вызовов процедур и функций из библиотек среды исполнения.

На эмуляторе и устройстве Pocket PC 2002 среда исполнения VB входит в состав операционной системы, а значит, не нуждается в дополнительной установке. На Pocket PC 2003 такую среду необходимо установить. Для этого необходимо пройти по ссылке <http://msdn.microsoft.com/mobility/windowsmobile/downloads/evbeula.aspx> или выполнить на сайте Microsoft поиск по ключевому слову msvbprc. Полученный файл с расширением. cab необходимо скопировать на PocketPC, и открыть его в файловом менеджере. После этого среда исполнения VB будет установлена.

Естественно, разработку и отладку приложений для Pocket PC 2003 надо производить на эмуляторе Pocket PC 2002, а затем отлаженное и работоспособное приложение по-

ставлять на Pocket PC 2003.

Чтобы скомпилировать и запустить приложение, достаточно нажать сочетание клавиш QH+F5 на клавиатуре или выполнить команду меню Run ► Execute. Приложение будет запущено на той платформе, которая выбрана из списка доступных целевых платформ (рис. 3.13).

## Настройки проекта

После того, как в среде создается новый проект, в меню становится доступной команда Project ► Properties. Эта команда управляет свойствами проекта и позволяет изменять некоторые настройки проекта. При выполнении команды Project ► Properties на экран выводится соответствующее диалоговое окно (рис. 3.16).

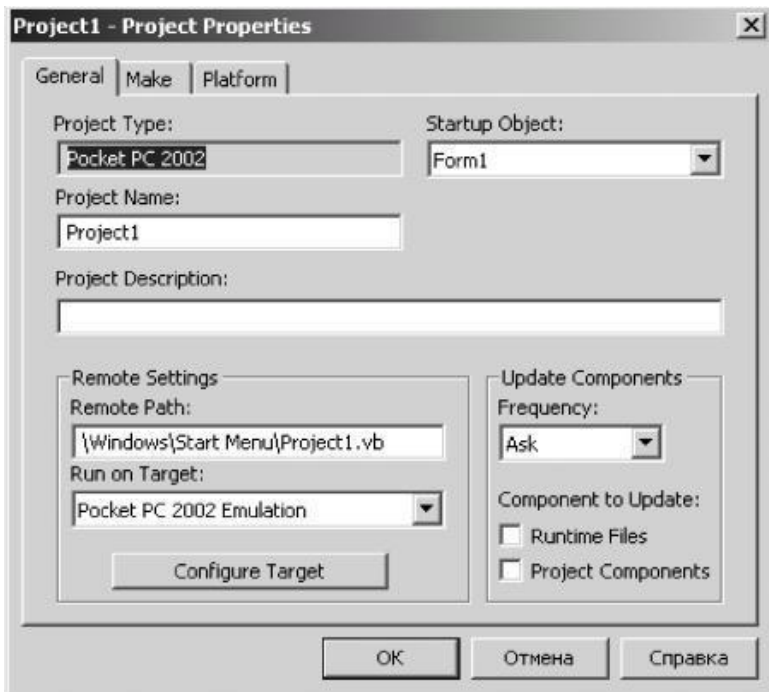


Рис. 3.16. Окно настройки свойств проекта.

В следующем списке рассматривается предназначение органов управления, располагающихся на вкладке General.

◆ Поле Project Type отображает тип проекта. Оно определяется в окне создания нового проекта и в дальнейшем не редактируется.

◆ Поле Project Name указывает имя проекта. Именно в этом поле видно имя, которое получит приложение при уста-

новке на целевое устройство или на эмулятор. По умолчанию имя проекта совпадает с именем файла проекта, но может и отличаться от имени файла проекта.

◆ В поле Project Description указывается краткое описание проекта. Это описание не используется нигде, кроме файла проекта.

◆ В списке Startup Object можно выбрать объект, который будет загружен, когда будет запущено приложение. Обычно в нем указывают одну из форм проекта.

◆ В поле Remote Path указывается каталог на устройстве, в который при установке будет помещен проект. По умолчанию используется каталог `\Windows\Start Menu\`, что позволяет сразу получить доступ к программе на устройстве (рис. 3.17).



Start

12:22

Recent Programs



Today



ActiveSync



Calendar



Contacts



Inbox



Internet Explorer



Notes



ProjectB



Tasks



Programs



Settings



Find



Help

New

Рис. 3.17. Приложение на устройстве помещено в каталог \Windows\Start Menu\.

◆ В списке Run on Target указывается платформа, на которой будет выполняться данное приложение. Разработчик может создать приложение для запуска на эмуляторе или на подключенном к компьютеру Pocket PC 2002.

◆ Нажатие кнопки Configure Target выводит на экран окно настройки Platform Manager, позволяющее настроить параметры соединения с выбранным устройством.

◆ Выпадающий список Frequency позволяет указать частоту обновления файлов на целевом устройстве.

◆ Флажок Always Ask указывает, что при каждом запуске приложения будет выведен запрос о том, нужно ли обновлять файлы.

◆ Орган управления Component to Update позволяет указать, какие компоненты, кроме файлов самого приложения, будут обновлены на устройстве.

- Runtime Files – среда выполнения VB.
- Project Components – компоненты ActiveX, если они входят в состав проекта.

Вкладка Make позволяет ввести информацию о версии файла, которая будет вписана в исполняемый файл проекта на стадии компиляции. Вкладка Platform позволяет выбрать устройство, которое будет использоваться по умолчанию для запуска данного приложения.

# **Использование встроенных компонентов eVB**

## **Окно настройки свойств (Properties Window)**

Окно настройки свойств по умолчанию располагается в правом нижнем углу экрана. Его внешний вид показан на рис. 3.18.



## Рис. 3.18. Окно настройки свойств.

При помощи окна настройки свойств пользователь может изменять значения параметров свойств выбранных на форме компонентов или самой формы.

В списке указано предназначение элементов этого окна в соответствии с цифрами на выносках.

1. Раскрывающийся список компонентов. В списке указывается имя компонента и его тип.

2. Список свойств для выбранного компонента.

3. Значения свойств.

4. Краткое описание текущего свойства. Оно далеко не всегда бывает информативным. Например, когда вы выбираете для формы свойство ShowOK, то описание сообщает вам, что это – свойство ShowOK. Больше никакой информации по поводу этого свойства не предоставляется.

5. Вкладки, переключающие представление списка свойств с алфавитного на категоризированный.

Чтобы изменить значение свойства, необходимо в списке 2 (см. рис. 3.18) найти соответствующее имя, щелкнуть мышью справа от него в зоне 3, а затем отредактировать значение свойства.

## **Типы свойств**

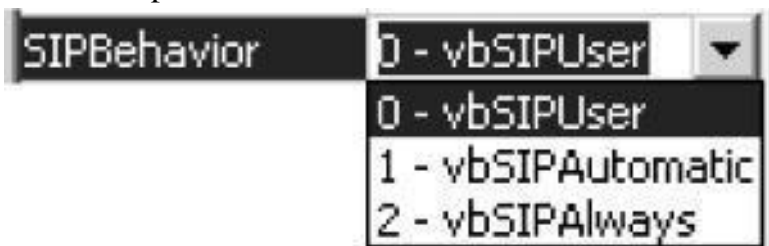
В eVB компоненты могут иметь свойства одного из трех

типов.

1. Обычное свойство редактируется прямым вводом в поле значения свойства. Оно представляет собой текстовую строку или число.



2. Перечислимое свойство имеет ряд предустановленных значений, из которых можно выбрать одно. В поле редактирования отображается в виде выпадающего списка.



3. Поведение сложного свойства зависит от комплекса параметров. Само свойство обычно является объектом. Сложное свойство редактируется при помощи редактора. Редактор свойства появляется при щелчке на кнопке с тремя точками, которая находится в поле ввода значения свойства и отображается при щелчке по этому полю.



# Компонент Form

## Методы компонента Form

Методы компонента Form приведены в табл. 3.1.

### Таблица 3.1. Методы компонента Form

Название метода	Описание метода
<b>Управление отображением и поведением формы</b>	
Show	Отображает на экране указанную форму. Для основной формы приложения вызывается автоматически в момент запуска приложения
Hide	Скрывает указанную форму
Move	Передвигает форму по экрану
ZOrder	Выводит форму на передний план или отправляет на задний
SetFocus	Устанавливает фокус ввода для формы
<b>Рисование на форме</b>	
Refresh	Перерисовывает поверхность формы
Cls	Очищает поверхность формы от нанесенных на нее во время выполнения рисунков
DrawCircle	Рисует на поверхности формы окружность или эллипс
DrawLine	Рисует на поверхности формы линии или прямоугольники
Point	Возвращает цвет точки, заданной координатами
PointSet	Рисует точку заданного цвета по заданным координатам

Название метода	Описание метода
<b>Масштабирование (преобразование единиц измерения)</b>	
SetScale	Устанавливает левый верхний и правый нижний углы для текущей системы координат
ScaleX	Возвращает горизонтальную координату точки, выраженную в новых единицах измерения
ScaleY	Возвращает вертикальную координату точки, выраженную в новых единицах измерения
TextHeight	Возвращает высоту текста, заданного в качестве аргумента, в единицах измерения текущей шкалы
TextWidth	Возвращает ширину текста, заданного в качестве аргумента, в единицах измерения текущей шкалы

## Свойства компонента Form

В табл. 3.2 приведено описание свойств компонента Form.

## Таблица 3.2. Свойства компонента Form

Название свойства	Описание свойства
ActiveControl	Возвращает компонент формы, который в данный момент находится в фокусе
AutoRedraw	Если это свойство имеет значение true, то рисованные объекты будут восстановлены после перерисовки формы, если используется значение false, то рисованные объекты будут уничтожены
BackColor	Цвет фона графических объектов
BorderStyle	Стиль рамки окна
Caption	Надпись на заголовке окна
ClipControls	Если свойство имеет значение false, то копии областей, занимаемых на форме визуальными компонентами, будут храниться в памяти программы, и при перерисовке формы будут просто скопированы в нее. Когда свойство имеет значение true, каждый компонент на форме будет вызывать свой собственный метод перерисовки
CurrentX	Устанавливает горизонтальную координату для последующего вызова графического метода (рисование линий или окружности)
CurrentY	Устанавливает вертикальную координату для последующего вызова графического метода (рисование линий или окружности)
DrawMode	Устанавливает режим вывода для графических методов (способ перекрытия или сочетания с цветом подложки)
DrawStyle	Позволяет рисовать графические объекты линиями разного стиля (сплошная, пунктирная, точечная и т. д.), но только при установленном значении DrawWidth=1. Если толщина линии больше единицы, то линия всегда будет сплошной
DrawWidth	Устанавливает толщину линии для рисования графических объектов
Enabled	Устанавливает, будет ли форма реагировать на действия пользователя
FillColor	Задает цвет заливки для графических объектов
FillStyle	Задает метод заливки для графических объектов
Font	Возвращает ссылку на объект Font
FontBold	Устанавливает полужирный стиль шрифта

<b>Название свойства</b>	<b>Описание свойства</b>
FontItalic	Устанавливает наклонный стиль шрифта
FontName	Устанавливает имя используемого шрифта
FontSize	Устанавливает размер используемого шрифта
FontStrikethru	Устанавливает перечеркнутый стиль шрифта
FontTransparent	Устанавливает цвет подложки шрифта, отличный от цвета подложки формы
FontUnderline	Устанавливает подчеркнутый шрифт
ForeColor	Цвет переднего плана (шрифта и рисования)
FormResize	Может находиться в трех состояниях. Значение 0 указывает, что форма отображается в полном размере, но автоматически уменьшается. Значение 1 указывает, что форма всегда отображается в полном размере, а экранная клавиатура перекрывает часть формы. Значение 2 указывает, что форма при отображении имеет размер, который был задан в режиме дизайнера
HDC	Идентификатор экранного контекста окна, устанавливается операционной системой динамически в режиме выполнения
Height	Высота окна
HWND	Идентификатор оконной процедуры, устанавливается операционной системой динамически в режиме выполнения
KeyPreview	Установка этого свойства в значение true приводит к тому, что любые манипуляции с клавиатурой (события нажатия и отпускания клавиш) всегда будут сначала приниматься формой, а затем уже объектом, находящимся на форме в фокусе ввода
Left	Горизонтальное расстояние левой кромки формы до левой кромки экрана
Moveable	Разрешает и запрещает перемещение формы по экрану
Name	Имя формы, используемое в коде
ScaleHeight	Устанавливает масштаб вертикальной шкалы внутренней системы координат формы. При установке ScaleHeight=50 вся текущая высота формы будет принята за 50 единиц, и этот масштаб будет использован при вызове графических методов
ScaleLeft	Устанавливает горизонтальную координату левой границы формы
ScaleMode	Устанавливает единицу измерения для выбранной системы координат
ScaleTop	Устанавливает вертикальную координату верхней границы формы
ScaleWidth	Устанавливает масштаб горизонтальной шкалы внутренней системы координат формы
ShowOK	При использовании значения true выводит в заголовок окна кнопку ОК
SIPBehavior	Задает поведение экранной клавиатуры. Значение 0 указывает, что клавиатура видима всегда. Значение 1 указывает, что клавиатура автоматически отображается при получении фокуса компонентом ввода текста. Значение 2 указывает, что клавиатура выводится по команде пользователя
SIPVisible	Отображает и скрывает экранную клавиатуру
Tag	Строковая переменная, связанная с объектом, которая может свободно использоваться программистом
Top	Вертикальное расстояние от верхней границы формы до верхней границы экрана
Visible	Отображает и скрывает форму
Width	Ширина формы

## События компонента Form

В этом разделе описываются события компонента Form. Все они приведены в табл. 3.3.

### Таблица 3.3. События компонента Form

Название события	Описание события
Activate	Происходит, когда форма становится активным окном
Deactivate	Происходит, когда форма становится неактивным окном, т. е. теряет фокус в пределах одного приложения
GotFocus	Событие наступает только тогда, когда форма получает фокус и на ней нет ни одного видимого компонента. Если на форме есть компонент, то при активизации формы фокус получает он
KeyDown	Происходит, когда пользователь нажимает любую клавишу на клавиатуре
KeyPress	Происходит, когда нажата и отпущена алфавитно-цифровая клавиша
KeyUp	Происходит, когда пользователь отпускает любую клавишу на клавиатуре
Load	Событие происходит, когда приложение загружает форму
LostFocus	Происходит, когда форма теряет фокус
MouseDown	Происходит в момент, когда пользователь нажимает клавишу мыши
MouseMove	Происходит, когда пользователь перемещает указатель мыши
MouseUp	Происходит, когда пользователь отпускает клавишу мыши
OKClick	Происходит при нажатии кнопки ОК
Paint	Происходит при перерисовке формы после передвижения, изменения размера или перевода на передний план
QueryUnload	Происходит непосредственно перед тем, как форма будет закрыта
Resize	Происходит во время первого показа и после изменения размера или статуса формы
SIPChange	Возникает при отображении и скрытии виртуальной клавиатуры
Terminate	Происходит при уничтожении всех ссылок на объект формы в памяти

## Упражнение 3.4

Это упражнение демонстрирует работу с методами объекта Form.

1. Создать новый проект с формой и сохранить его в папке FormMethods, задав имя файла FormMethods.ebp. Будет создан проект с формой Form1.

2. Нажать кнопку AddForm, что приведет к добавлению к проекту еще одной формы с именем Form2.

3. Разместить на форме Form1 две кнопки так, как это показано на рис. 3.19.

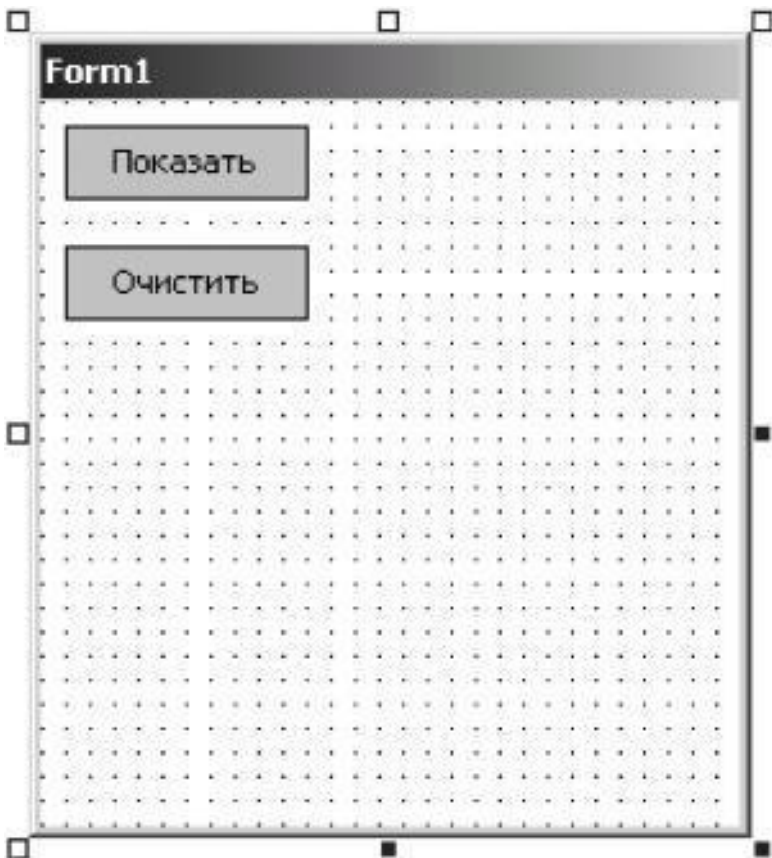


Рис. 3.19. Размещение кнопок на форме Form1.

4. Пользуясь инспектором свойств, установить для формы Form1 значение true для свойства AutoRedraw и значение 20 для свойства DrawWidth. Остальным свойствам можно оста-

вить их значения по умолчанию.

5. В окне Project Explorer выбрать форму Form2. Необходимые значения свойств для этой формы приведены в табл. 3.4. Остальные свойства нужно оставить без изменений.

**Таблица 3.4. Свойства формы Form2**

<b>Свойство</b>	<b>Значение</b>
AutoRedraw	true
BackColor	фиолетовый
BorderStyle	None
DrawWidth	10
FormResize	vbFormUserSize
Height	2000
Width	2000

6. Форма изменит цвет и размеры. На форме Form2 требуется разместить компонент Timer, как это показано на рис. 3.20.

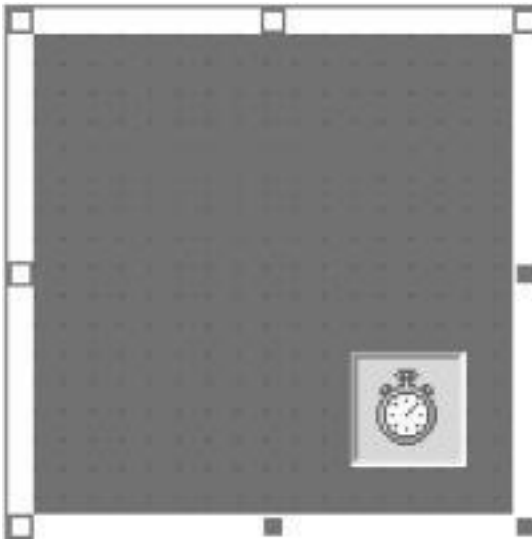


Рис. 3.20. Форма Form2 после редактирования свойств и размещения таймера.

7. Щелчком мыши выделить компонент Timer1 и для его свойства Interval установить значение 150.

8. Двойным щелчком мыши на компоненте Timer1 создать подпрограмму обработки события одного тика таймера и ввести в нее код, приведенный в листинге 3.3.

### **Листинг 3.3**

```
Private Sub Timer1_Timer()
```

```

i=i+a
j=j+b
If i > 1600 Then a = - a: BackColor =
&HFF&:
  DrawCircle 500, 500, 400, &HFF00FF:
  DrawLine 1100, 1100, 1800, 1800,
&HFF00FF, vbLineBoxFill
  If j > 2500 Then b = - b: BackColor =
&HFFFF&:
  DrawCircle 500, 500, 400, &HFF00FF:
  DrawLine 1100, 1100, 1800, 1800,
&HFF00FF, vbLineBoxFill
  If i < 0 Then a = - a: BackColor = &HFF00&:
  DrawCircle 500, 500, 400, &HFF00FF:
  DrawLine 1100, 1100, 1800, 1800,
&HFF00FF, vbLineBoxFill
  If j < 0 Then b = - b: BackColor =
&HFF0000:
  DrawCircle 500, 500, 400, &HFF00FF:
  DrawLine 1100, 1100, 1800, 1800,
&HFF00FF, vbLineBoxFill
Move i, j
End Sub

```

Этот код демонстрирует применение методов DrawCircle, DrawLine и Move для того, чтобы перемещать форму Form2

и рисовать на ее поверхности круг и прямоугольник.

9. Кроме кода обработчика события тика таймера необходимо после строки `Option Explicit` ввести еще один блок кода объявления и инициализации переменных, приведенный в листинге 3.4.

### Листинг 3.4

```
Dim a, b, i, j  
i=0  
j=0  
a = 100  
b = 50
```

10. В окне `Project Explorer` необходимо переключиться на объект `Form1`. Для кнопок `Command1` и `Command2` свойство `Caption` должно получить значения `Показать` и `Очистить` соответственно.

11. Двойным щелчком на кнопке `Command1` создать обработчик события щелчка на кнопке и ввести для него код, приведенный в листинге 3.5.

### Листинг 3.5

```
Private Sub Command1_Click()
```

```
If ShFrm2 Then
Form2.Show
ShFrm2 = False
Command1.Caption = «Скрыть»
Else
Form2.Hide
ShFrm2 = True
Command1.Caption = «Показать»
End If
End Sub
```

Этот метод демонстрирует применение методов Show и Hide для отображения и скрытия формы Form2 при нажатии кнопки Command1.

12. Двойным щелчком на кнопке Command2 создать обработчик события щелчка на кнопке и ввести для него код, приведенный в листинге 3.6.

### **Листинг 3.6**

```
Private Sub Command2_Click()
Cls
Form2.ZOrder vbBringToFront
End Sub
```

Команда Cls очищает поверхность формы Form1 от на-

рисованных на ней линий, а метод ZOrder выводит форму Form2 на передний план после выполнения этой команды. Без этого метода она становится невидимой, так как ее за-слоняет форма Form1.

13. Ввести еще один блок кода в этом же модуле, приведенный в листинге 3.7.

### **Листинг 3.7**

```
Private Sub Form_Load()  
DrawLine 1100, 1100, 9000, 18000, &HFF&  
DrawLine 1100, 1100, 18000, 10000, &HFF&  
DrawLine 1100, 1100, 18000, 18000, &HFF&  
End Sub
```

При помощи этого кода в момент загрузки формы Form1 на ней будут нарисованы три линии красного цвета.

14. В начале этого же блока следует объявить и проинициализировать переменную, которая будет хранить состояние формы. Делается это при помощи кода, приведенного в листинге 3.8.

### **Листинг 3.8**

```
Option Explicit
```

```
Dim ShFrm2 As Boolean  
ShFrm2 = True
```

15. Сохранить проект. В качестве целевого устройства нужно выбрать в списке значение Pocket PC 2002 Emulation, после чего следует выполнить команду Run ► Execute. На экран эмулятора будет выведена форма Form1 с нарисованными на ней линиями. После нажатия кнопки Показать на экран будет выведена форма Form2 с нарисованными на ней окружностью и квадратом, а надпись на кнопке изменится на Скрыть (рис. 3.21).



Form2

⏪ 1:45



Скрыть

ОЧИСТИТЬ



## Рис. 3.21. Программа FormMethods в работе/

Форма Form2 будет перемещаться по экрану и изменять цвет в момент столкновения с границами экрана. Нажатие кнопки Очистить приведет к тому, что будут стерты линии с поверхности формы Form1. При работе с приложением стоит несколько раз скрыть и снова отобразить форму Form2. Это покажет, что форма не уничтожается при скрытии и продолжает свое перемещение в невидимом состоянии.

### Упражнение 3.5

Это упражнение демонстрирует, как работают события объекта Form.

1. Создать новый проект с формой и сохранить его под именем FormEvents.
2. Разместить на форме компоненты TextBox, ListBox и три кнопки CommandButton так, как это показано на рис. 3.22.

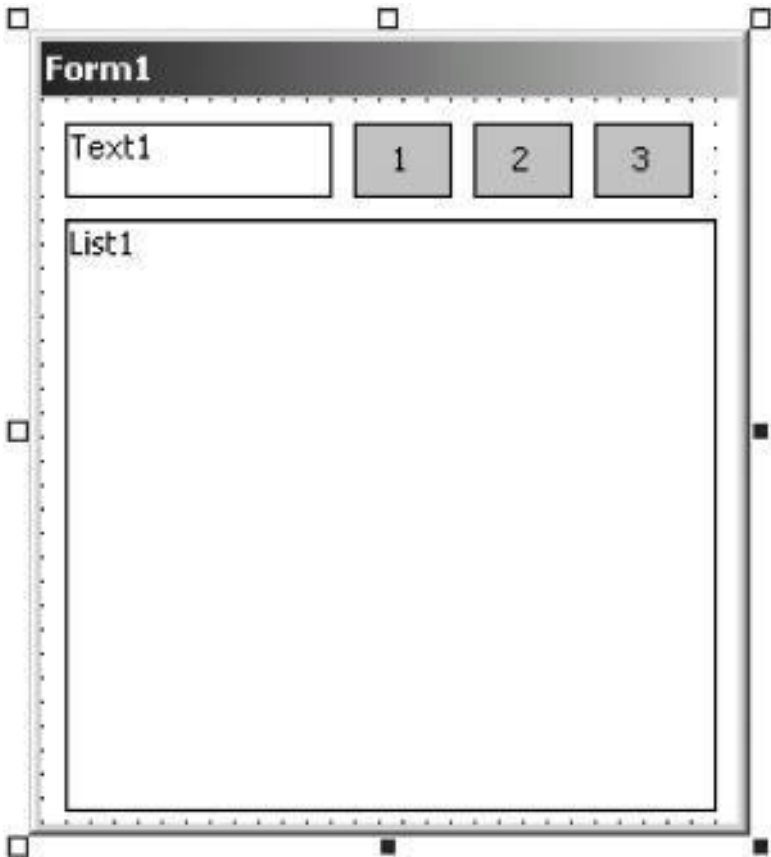


Рис. 3.22. Размещение компонентов на форме.

3. При помощи инспектора свойств нужно установить значения свойств для компонентов. Все необходимые значения приведены в табл. 3.5.

## Таблица 3.5. Значения свойств кнопок

Свойство	Значение
Command1.Caption	1
Command1.Caption	2
Command1.Caption	3

4. Добавить к проекту еще одну форму. Значения ее свойств можно найти в табл. 3.6.

## Таблица 3.6. Свойства новой формы

Свойство	Значение
BackColor	красный цвет
BorderStyle	Sizable
ControlBox	true
Height	3000
Left	1000
Top	1000
Width	3000

5. Перейти в окне Project Explorer к форме Form1. Правой кнопкой мыши нужно щелкнуть на форме, в контекстном меню выполнить команду View Code и ввести код модуля, приведенный в листинге 3.9.

## Листинг 3.9

Rem Первая кнопка выводит Форму Form2 на

экран

```
Private Sub Command1_Click()  
Form2.Show  
End Sub
```

```
Rem Вторая кнопка скрывает форму Form2  
Private Sub Command2_Click()  
Form2.Hide  
End Sub
```

```
Rem Третья кнопка изменяет размер формы  
Form2
```

```
Private Sub Command3_Click()  
If Form2.Height > 2500 Then  
Form2.Height = 1000: Form2.Width = 1000  
Else  
Form2.Height = 3000: Form2.Width = 3000  
End If  
End Sub
```

```
Private Sub Form_OKClick()  
App.End  
End Sub
```

```
Rem Двойной щелчок по компоненту списка  
очищает его содержимое
```

```
Private Sub List1_DblClick()  
List1.Clear  
End Sub
```

Rem Вывод сообщений о показе и скрывании виртуальной клавиатуры

```
Private Sub Form_SIPChange(ByVal IsV As Boolean)  
If IsV Then  
List1.AddItem «Клавиатура показана»  
Else  
List1.AddItem «Клавиатура скрыта»  
End If  
End Sub
```

Rem Далее в подпрограммах отслеживаем события от клавиатуры и мыши

```
Private Sub Form_KeyDown(ByVal KeyCode As Integer, ByVal shift As Integer)  
List1.AddItem «Text1_KeyDown»  
End Sub
```

```
Private Sub Form_KeyPress(ByVal KeyAscii As Integer)  
List1.AddItem «Text1_KeyPress»  
End Sub
```

```
Private Sub Form_KeyUp(ByVal KeyCode As Integer, ByVal shift As Integer)
    List1.AddItem «Text1_KeyUp»
End Sub
```

```
Private Sub Form_MouseDown(ByVal button As Integer, ByVal shift As Integer, x, y)
    List1.AddItem «Form_MouseDown»
End Sub
```

```
Private Sub Form_MouseUp(ByVal button As Integer, ByVal shift As Integer, _x, y)
    List1.AddItem «Form_MouseUp»
End Sub
```

Rem Для того чтобы не засорять список,  
событие передвижения указателя

Rem мыши над формой мы обрабатываем только  
при нажатой левой клавише

```
Private Sub Form_MouseMove(ByVal button As Integer, ByVal shift As Integer, x, y)
    If button = 1 Then List1.AddItem «Form_MouseMove»
End Sub
```

6. Перейти в окне Project Explorer к форме Form2. В ее модуле следует ввести код, приведенный в листинге 3.10.

### Листинг 3.10

```
Private Sub Form_QueryUnload(ByVal cancel As Integer, unloadmode)
Form1.List1.AddItem «Form2 QueryUnload»
End Sub

Private Sub Form_Activate()
Form1.List1.AddItem «Form2 Activate»
End Sub

Private Sub Form_Deactivate()
Form1.List1.AddItem «Form2 Deactivate»
End Sub

Private Sub Form_GotFocus()
Form1.List1.AddItem «Form2 Получила фокус»
End Sub

Private Sub Form_LostFocus()
Form1.List1.AddItem «Form2 Потеряла фокус»
End Sub

Private Sub Form_Paint()
Form1.List1.AddItem «Form2 перерисована»
```

```
End Sub
```

```
Private Sub Form_Load()  
Form1.List1.AddItem «Form2 загружена»  
End Sub
```

```
Private Sub Form_Resize()  
Form1.List1.AddItem « Form2 Resize»  
End Sub
```

7. Выполнить команду меню Run ► Execute и запустить приложение в эмуляторе. На экран будет выведена первая форма.

8. После нажатия кнопки 1 на экран будет выведена вторая форма, а в ListBox будут выведены сообщения о том, какими событиями сопровождался показ этой формы.

9. Переместить форму Form2 так, чтобы форма выходила за пределы экрана. При этом будет отображено сообщение о том, что форма перерисована.

10. Нажать кнопку 3. После этого нажатия фокус ввода будет перенесен на форму Form1, а форма Form2 будет скрыта. Поэтому появятся сообщения о том, что форма Form2 деактивирована, потеряла фокус, а затем о том, что она два раза поменяла размер. Это соответствует истине, так как сначала был изменен вертикальный, а затем горизонтальный размер формы.

11. Нажать кнопку 2. Будут отображены сообщения о том, какими событиями сопровождается сокрытие формы.

12. Остановить и снова запустить приложение. Нажать кнопку 3. При этом форма Form2 не появится на экране. Тем не менее, форма эта будет загружена, и ее размер будет изменен.

13. Сделать несколько щелчков мышью на форме Form1 вне визуальных компонентов. Будут отображены сообщения о событиях, связанных с нажатием и отпусканием кнопки мыши.

14. Перевести фокус в компонент TextBox и вывести на экран виртуальную клавиатуру. Появится сообщение о выводе клавиатуры. При работе с клавиатурой появится множество соответствующих сообщений. Если нажать на алфавитно-цифровую клавишу, и не отпустить ее, то в списке не будет отображено ни одно событие, хотя должно отобразиться событие Down. Однако сразу после отпускания отображаются сразу три события – Down, Press и Up. Это не особенность eVB и не особенность виртуальной клавиатуры, это явная ошибка среды разработки, о чем сказано в базе знаний на сайте Microsoft.

15. При скрывании экранной клавиатуры будет выведено соответствующее сообщение.

16. Следует обратить внимание, что наборы сообщений во время скрывания формы Form2 при помощи метода Hide и при помощи кнопки с крестиком на форме отличаются. Во

втором случае добавляется событие QueryUnload.

### **СОВЕТ.**

*Три кита, на которых стоит программирование интерфейса пользователя в eVB, – методы, свойства и события. Если описание свойств и методов довольно подробно дано в справочной системе, то события (особенно их последовательность) там описаны не так подробно. Я советую вам использовать тот же подход, который мы использовали в этом упражнении, для исследования событий других компонентов, порядка и особенности их генерации при работе компонентов в том или ином режиме, и то, какие параметры передаются в событие и какие из него возвращаются обратно приложению или операционной системе.*

17. Чтобы увидеть, как параметр возвращается из события в приложение, следует изменить код события QueryUnload, введя код из листинга 3.11.

### **Листинг 3.11**

```
Private Sub Form_QueryUnload(cancel, unloadmode)
Form1.List1.AddItem «Form2 QueryUnload»
cancel = 0
```

```
End Sub
Private Sub Form_Unload(cancel)
Form1.List1.AddItem «Form2 Unload»
End Sub
```

Параметр `cancel` передается не по значению, а по ссылке, значит, он будет возвращен приложению. Этот параметр, установленный в нулевое значение, позволяет форме быть деактивированной. Если же установить его в любое, отличное от нуля, значение, то форма останется активной и видимой.

### ***ВНИМАНИЕ!***

*Несмотря на то что событие `Unload` генерируется при установлении параметра `cancel = 0`, сама форма не выгружается из памяти.*

## **Другие встроенные компоненты**

На рис. 3.23 изображена палитра встроенных компонентов eVB. Каждый из компонентов на рисунке помечен выноской. Краткое описание каждого из компонентов в соответствии с номером на выноске приведено в соответствующем списке.

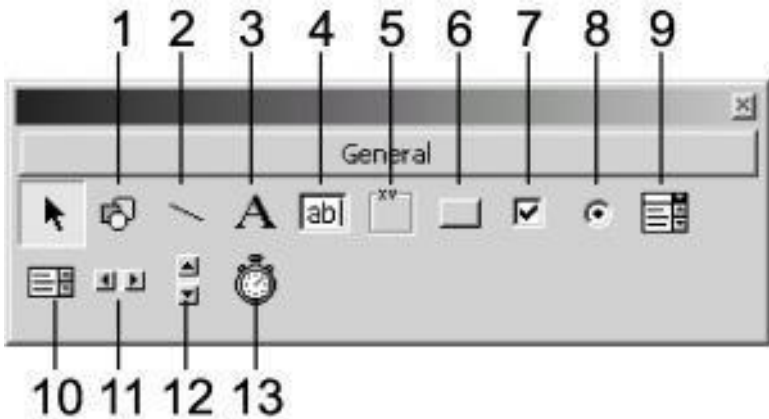


Рис. 3.23. Набор встроенных компонентов eVB.

1. Shape. Компонент предназначен для вывода на форму закрашенной плоской фигуры (круг, овал, прямоугольник, квадрат, квадрат или прямоугольник со сглаженными углами). Выбор типа фигуры осуществляется установкой свойства Shape. Благодаря наличию методов Move, Refresh и ZOrder возможно манипулирование объектом Shape во время выполнения программы. Объект Shape не может служить контейнером для других объектов. События у объекта Shape отсутствуют.

2. Line. Компонент выводит на форму отрезок прямой заданного цвета, толщины и длины. Отрезок задается координатами двух точек. Метод Move у объекта Line отсутствует, но перемещение отрезка во время выполнения возможно

при помощи изменения его координат.

3. **Label**. Компонент обеспечивает отображение на форме многострочного текста. Сам текст задается установкой свойства `Caption` и может иметь в длину до 1024 символов. На отображение текста влияют свойства `AutoSize` (компонент изменяет свой размер согласно объему введенного текста) и `WordWrap` (введенный текст, достигнув горизонтальной границы компонента, переносится на следующую строку по границе слова). Также возможна установка параметров шрифта при помощи изменения свойства `Font` и параметров выравнивания установкой значения свойства `Alignment`. Компонент `Label` реагирует на событие `Change`, которое возникает при программном изменении свойства `Caption`. Для управления расположением компонента на форме во время выполнения можно использовать методы `Move`, `Refresh` и `ZOrder`.

4. **TextBox**. Компонент обеспечивает вывод на форму однострочного или многострочного редактируемого текста. Текст присутствует в компоненте в качестве значения свойства `Text`. Свойство `MultiLine` позволяет переключаться между однострочным и многострочным режимом. В однострочном режиме длина строки ограничена 2048 символами, в многострочном режиме общий объем текста может достигать 32 Кбайт. Возможно задание параметров шрифта (свойство `Font`) и выравнивания текста (свойство `Alignment`). Если компонент находится в многострочном режиме, то наличие вертикальной и горизонтальной полос прокрутки

можно задать установкой свойства `ScrollBars`. В компоненте `Text` возможно выделение фрагментов при помощи сочетания клавиш виртуальной клавиатуры или стилуса, а также вырезание, копирование, удаление и вставка из буфера. При осуществлении этих операций программным путем хорошим подспорьем в работе будут свойства `SelLength` (количество выделенных символов), `SelStart` (индекс символа, с которого начато выделение) и `SelText` (строка, содержащая выделенный текст). Естественно, основным предназначением этого компонента является ввод текста с виртуальной клавиатуры. Поэтому компонент реагирует на такие события клавиатуры, как нажатие и отпускание клавиш (`KeyDown`, `KeyPress` и `KeyUp`). К сожалению, эти три события в `eVB` не используются по отдельности. Причины этого досадного недоразумения излагались в комментариях к упражнению 3.5.

5. `Frame`. Компонент-контейнер, который позволяет объединять другие видимые компоненты в функциональные группы. Событий не имеет. Свойства компонента в особых комментариях не нуждаются. Чаще всего в компонент `Frame` помещают несколько компонентов `OptionButton`, которые при этом автоматически объединяются в группу.

6. `CommandButton`. Командная кнопка, главное назначение которой – генерировать событие щелчка на кнопке. Это событие с именем `Click` является основным для компонента `CommandButton`.

7. **CheckBox**. Компонент, обеспечивающий индикацию или выбор состояния одного из нескольких независимых параметров. Основным свойством компонента является свойство `Value`. Значение 0 указывает, что компонент не выбран, значение 1 указывает, что пользователь взвел флажок, а значение 2 свидетельствует о том, что компонент не активен. Основным событием компонента **CheckBox** является событие `Click`, которое возникает при изменении значения свойства `Value`, вне зависимости от того, программным путем меняется это значение или в результате нажатия стилусом.

8. **OptionButton**. Компонент, обеспечивающий выбор только одного из нескольких зависимых параметров, объединенных в группу. Если не считать функционального назначения, то свойства компонента **OptionButton** практически не отличаются от свойств **CheckBox**. Еще одно важное отличие состоит в том, что любой контейнер, в который помещаются несколько компонентов **OptionButton**, объединяет их в зависимую группу.

9. **ComboBox**. Выпадающий список, обеспечивающий выбор одной из строк списка, или ввод нужного значения непосредственно в строку редактирования. Основным свойством компонента является набор значений. Этот набор можно ввести в режиме дизайна, нажав кнопку, которая появляется в поле значений при выборе свойства `Custom`. В окне редактора свойств для каждого элемента списка можно ввести строковое значение (`ListData`) или целочисленное значе-

ние (ItemData). Список выводится только во время выполнения программы, причем в списке отображается только набор строк ListData, а набор значений ItemData служит для связи списка с другими объектами. Свойство Style позволяет разрешить или запретить непосредственный ввод значений в строку редактирования. Редактирование текста в строке производится при помощи тех же свойств (SelLength, SelStart, SelText), что и в компоненте TextBox. Доступ к элементам списка в момент выполнения программы осуществляется при помощи дополнительных свойств. Свойство List возвращает строку, соответствующую указанному индексу. Свойство ListCount возвращает количество элементов в списке. Свойство ListIndex возвращает индекс выбранного элемента списка. Свойство NewIndex возвращает индекс последнего добавленного элемента списка, а свойство TopIndex возвращает индекс первого видимого элемента в прокручиваемом списке. При последовательном изменении значения этого свойства можно программным путем осуществлять прокрутку списка. Наконец, свойство Sorted при установке значения true вынуждает компонент отображать элементы списка отсортированными в алфавитном порядке. Для манипуляций элементами списка во время выполнения программы необходимо использовать методы AddItem и RemoveItem, которые добавляют и удаляют элементы списка с указанным индексом. Метод Clear позволяет очистить список.

События компонента `ComboBox` не радуют нас разнообразием. Событие `Change` наступает при программном или пользовательском редактировании строки ввода. Событие `Click` наступает при выборе элемента списка. При прокрутке списка наступает событие `Scroll`.

10. `ListBox`. Поскольку компонент `ComboBox` является комбинацией компонентов `TextBox` и `ListBox`, то все, относящееся к нему, будет верно и для компонента `ListBox`. Компонент `ListBox` отличается только реализацией. Он постоянно присутствует на форме в виде прямоугольной области, в которой размещен прокручиваемый список. Небольшое отличие состоит в том, что в компоненте `ListBox` возможно осуществлять выбор сразу нескольких элементов списка. Для этого необходимо установить свойству `MultiSelect` значение 1 для последовательного выделения элементов или значение 2 для произвольного выделения нескольких элементов списка одновременно.

11. `HScrollBar`. Компонент реализует горизонтальную полосу прокрутки. Используется для установки или для индикации значения, которое может плавно изменяться в широких пределах. Свойство `Max` позволяет задавать максимальное значение, которое может принимать свойство `Value`. Оно соответствует крайнему правому положению движка компонента. Свойство `Min` устанавливает минимальное значение, которое может принимать свойство `Value`. Свойство `Value` хранит значение, соответствующее текущему положению

нию движка полосы прокрутки. Свойство `LargeChange` содержит значение смещения, которое соответствует большому шагу движка, при щелчке стилусом на полосе прокрутки. Свойство `SmallChange` соответствует малому шагу движка при нажатии стилусом на кнопку со стрелкой на полосе прокрутки. Кроме этого, для передачи значений от компонента `Scrollbar` в настраиваемое свойство используют события `Scroll` (происходит при перемещении движка) и `Change` (происходит при перемещении движка или изменении свойства `Value` программным путем).

12. `VScrollbar`. Компонент реализует вертикальную полосу прокрутки. В остальном компонент соответствует `HScrollbar`.

13. `Timer`. Единственным назначением этого компонента является генерация события `Timer` с интервалом, который устанавливается свойством `Interval`. Свойство `Interval` может быть установлено в пределах от 1 до 65 535. Когда надо приостановить работу таймера, следует установить его свойство `Enabled` в `false`.

Несмотря на то что встроенные компоненты `eVB` позволяют реализовать довольно широкий набор функций, можно заметить, что для создания полноценного приложения не хватает достаточно многих компонентов. На палитре нет элементов меню, нет операций с файлами, нет компонентов для работы с данными, представленными в табличном виде или в виде иерархического дерева. Даже графические изоб-

ражения вывести нельзя.

Вся эта функциональность, безусловно, есть. Но реализована она в виде компонентов ActiveX. Компоненты ActiveX вынесены в отдельную группу, поскольку в самой среде eVB создать их невозможно. Visual Basic не умеет полноценно работать с COM-технологиями, но зато умеет эффективно использовать плоды их работы, то есть созданные ранее компоненты ActiveX.

# Использование компонентов ActiveX

## Расположение компонентов ActiveX

Если вы хотите использовать компоненты ActiveX в своем проекте, их надо соответствующим образом к этому проекту добавить. Для этого нужно выполнить команду меню Project ► Components или нажать сочетание клавиш Ctrl+T. На экран будет выведено окно добавления ActiveX-компонента к проекту (см. рис. 3.11). В этом окне нужно установить флажки напротив компонентов, которые нужно использовать в проекте, и нажать кнопку ОК. Добавленные к проекту компоненты будут отображены в палитре компонентов. На рис. 3.24 показано, как на палитре появился элемент MenuBar Control.

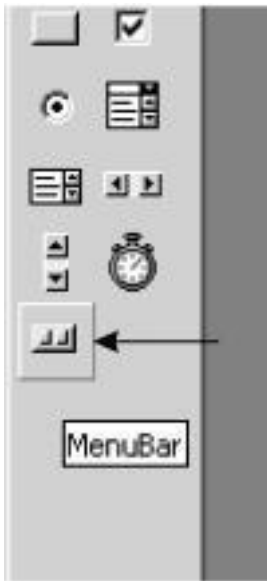


Рис. 3.24. Компонент MenuBar, помещенный на палитру.

## **Общие вопросы использования компонентов ActiveX**

В отличие от встроенных компонентов, которые для своей работы в проекте требуют только помещения на форму, компоненты ActiveX являются COM-объектами, размещены в отдельных библиотечных файлах и должны быть установлены и зарегистрированы на той машине, где будет запускаться приложение.

Для поставки и регистрации компонентов на пользовательском компьютере есть несколько способов. Самый простой способ заключается в том, что нужно поручить поставку и регистрацию самой среде eVB. Этот путь более всего подходит для отладки программ на эмуляторе. Чтобы среда скопировала файлы библиотек на пользовательскую машину и зарегистрировала их там, необходимо выполнить простую последовательность действий.

- ◆ Открыть окно настройки Project ► Project Properties.

- ◆ На вкладке General выбрать из списка Frequency значение Always, а в группе Component to Update установить флажок Project Components.

Эти настройки заставляют среду копировать все файлы проекта на компьютер и регистрировать компоненты ActiveX при каждом запуске. Если не добавлять постоянно файлы к проекту, то, запустив программу один раз, можно отключить эти настройки. Среда копирует библиотеки, входящие в состав проекта, и регистрирует их.

Вторым способом установить и зарегистрировать компоненты ActiveX является использование инструмента Control Manager. Для этого нужно выполнить другую последовательность действий.

- ◆ Выполнить команду меню Tools ► Remote Tools ► Control Manager. На экран будет выведено окно, внешний вид которого показан на рис. 3.25.

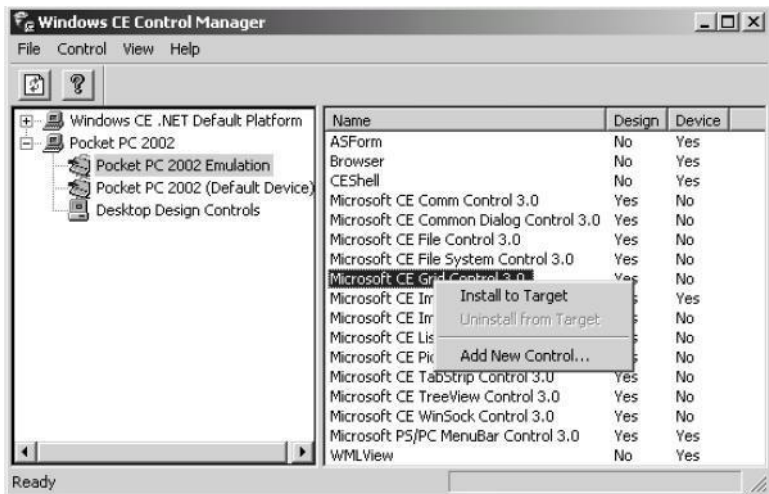


Рис. 3.25. Окно инструмента Control Manager.

◆ В левой части окна выбрать устройство, на которое будет осуществляться поставка приложения (на рисунке это Pocket PC 2002 Emulation). В правой части окна нужно найти библиотеку, которую необходимо установить и зарегистрировать (на рисунке выбрана библиотека Microsoft CE Grid Control). На имени библиотеки нужно щелкнуть правой кнопкой мыши и выполнить команду контекстного меню Install to Target. Библиотека будет установлена и зарегистрирована.

Наконец, установку и регистрацию библиотеки можно провести в «ручном» режиме. Для этого библиотеку необ-

ходимо скопировать на устройство (при помощи инструмента File Viewer или ActiveSync), а затем зарегистрировать при помощи программы RegSvrCE.exe. Эта программа обычно располагается в каталоге C:\Windows CE Files\wce300\Pocket PC 2002\target\arm или в каталоге C:\Windows CE Files\wce300\Pocket PC 2002\target\x86. В каталоге arm находится программа для работы на компьютере, а в каталоге x86 – для работы на эмуляторе. Эту программу также необходимо скопировать на целевое устройство. Библиотеки компонентов также находятся в двух подкаталогах arm и x86 каталога C:\Program File\Windows CE Files\wce300\Pocket PC 2002\controls.

◆ После того, как программа RegSvrCE.exe и библиотека с компонентом, который необходимо установить, будут скопированы на целевое устройство, необходимо запустить программу RegSvrCE.exe. На экран будет выведено диалоговое окно, внешний вид которого показан на рис. 3.26.



RegSvrCE

10:55



Full pathame of the  
DLL or control

OK

MSCETreeView.dll

Cancel

**Action**

Register  Unregister

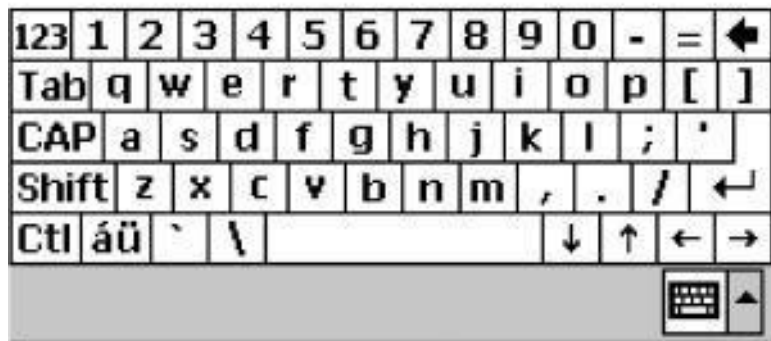


Рис. 3.26. Окно программы RegSvrCE.exe.

◆ В этом окне необходимо ввести полный путь к библиотеке, включая имя файла, взвести флажок Register и нажать кнопку ОК.

◆ Если регистрация прошла успешно, на экран будет выведено соответствующее сообщение (рис. 3.27).

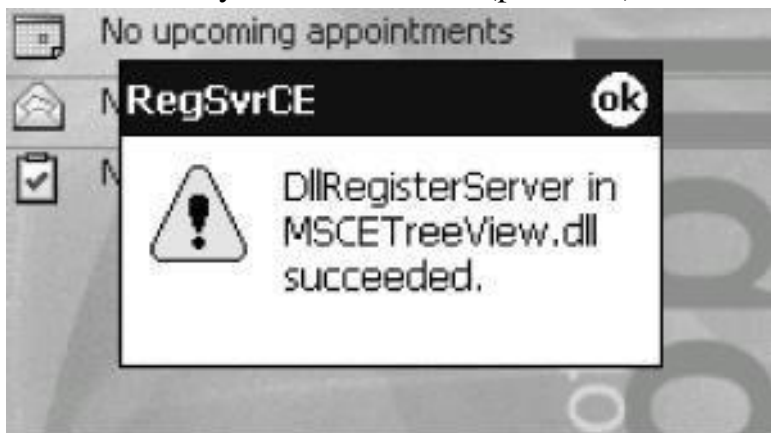


Рис. 3.27. Сообщение о регистрации компонента.

Данное сообщение не отображается поверх всех окон. Чтобы его увидеть, надо закрыть все приложения и вернуться в основное окно устройства.

Есть еще один способ регистрации компонентов на устройстве назначения. Компоненты могут регистрироваться при установке из поставочного пакета. Но этот процесс будет рассматриваться отдельно в следующих разделах.

# Компонент MenuBar

Этот компонент предназначен для создания основного меню в программах для Pocket PC.

При помещении на форму данный компонент может сбить с толку тех, кто привык к визуальному дизайну. Выглядит он бедненько, и никаких средств настройки в инспекторе объектов обнаружить нельзя. В режиме дизайна компонент похож на кнопку, а не на меню. Тем не менее этот компонент позволяет создавать полнофункциональное основное меню.

Свойство `Enabled` позволяет делать меню недоступным в ходе выполнения приложения. Для этого достаточно лишь установить для свойства значение `false`. Если для свойства `NewButton` задать значение `true`, то на полосе меню появится новая кнопка с именем `New`. Использование этих свойств обсуждается в следующем упражнении.

## Упражнение 3.6

1. Создать новый проект с формой и сохранить его в папке `EnPhrase` с именем `EnPhrase.ebp`.

2. Установить для свойства формы `Caption` значение `Учим английские фразы`, а значение свойства `Height` нужно сделать равным `7050`. Такая длинная форма нужна для того, чтобы посмотреть, как работает компонент `VScrollBar`.

3. Добавить к проекту компонент Frame. Значения его свойств приведены в таблице 3.7.

**Таблица 3.7. Свойства компонента Frame**

Свойство	Значение
Height	4050
LargeChange	1000
Left	3360
Max	3000
Min	0
SmallChange	100
Width	255

4. Добавить к проекту три компонента Label. Значения свойств этих компонентов приведены в табл. 3.8.

**Таблица 3.8. Значения свойств компонентов Label1, Label2 и Label3**

Свойство	Label1	Label2	Label3
BackStyle	1	1	1
BorderStyle	1	1	1
Caption	""	""	""
Height	1000	1000	1000
Left	120	120	120
Top	300	1350	1350
Width	3135	3135	3135
WordWrap	true	true	true

5. Добавить к проекту четыре компонента

CommandButton. Значения свойств этих компонентов приведены в табл. 3.9.

**Таблица 3.9. Значения свойств компонентов Command1, Command2, Command3 и Command4**

Свойство	Command1	Command2	Command3	Command4
Caption	Принять	Сброс	<—	Подсказка
Height	375	375	375	375
Left	120	960	1800	2250
Top	2520	2520	2520	2520
Width	735	735	350	1000

6. Добавить к проекту компонент ListBox. Значения свойств этого компонента приведены в табл. 3.10.

**Таблица 3.10. Значения свойств компонента ListBox**

Свойство	Значение
Columns	2
Font	Tahoma, bold, 10
Height	30
Left	120
MultiSelect	0
Top	3000
Width	3135

7. В первом приближении дизайн программы завершен. В дизайнерах форм он будет выглядеть так, как это показано на

рис. 3.28.

## Учим английские фразы

Принять

Сброс

<-

Подсказка

**List1**

## Рис. 3.28. Дизайн программы.

8. Чтобы фрейм, который имеет размер больший, чем размер экрана, можно было прокручивать при помощи вертикальной полосы прокрутки, нужно написать обработчик события Change для компонента VScroll. Код этого обработчика приведен в листинге 3.12.

### Листинг 3.12

```
Private Sub VScroll1_Change()  
Frame1.Top = -200 - VScroll1.Value  
End Sub
```

Теперь можно запустить проект в эмуляторе и посмотреть, как при помощи полосы вертикального скроллинга происходит прокрутка экрана. Граничные значения свойства Value равны тому значению, на которое надо будет сдвигать фрейм в окне, и поэтому обработчик события получился простым. Начальное значение -200 нужно для того, чтобы край фрейма не отображался на форме и не портил общий внешний вид приложения.

9. Чтобы создать в приложении основное меню, необходимо сначала подключить к приложению соответствующую библиотеку. Для этого надо выполнить команду Project ► Components и установить флажок напротив компонен-

та Microsoft Pocket PC MenuBar Control. Следует также задать настройки для автоматической регистрации компонента на устройстве при помощи команды Project ► Project Properties|Update Components ► Always|Component to Updates ► Project Components.

10. После того как библиотека будет добавлена к проекту, соответствующий значок появится на палитре компонентов, и компонент MenuBar можно будет поместить на форму. Для свойства NewButton нужно установить значение false. Запуск проекта в таком состоянии не покажет никаких изменений. Чтобы меню отобразилось на форме, необходимо настроить и создать объекты, соответствующие тому, что должно отображаться на форме. Это делается программным путем в обработчике события Form\_Load().

11. Прежде чем писать обработчик события, нужно разобраться с компонентом MenuBar. Сам компонент представляет собой коллекцию команд меню (объект MenuBarMenu) и кнопок (объекты MenuBarButton), которая создается и отображается на нижней панели экрана Pocket PC. Кроме свойств объекта MenuBar, которые отображены в инспекторе, есть еще свойство Collection, содержащее индексированную коллекцию меню и кнопок, и свойство ImageList, содержащее ссылку на коллекцию изображений, которые будут отображаться на кнопках. События ButtonClick, MenuClick и NewClick предназначены для обработки событий щелчка на кнопках, пунктах меню и кнопки New соответственно.

Добавление компонентов к коллекции производится при помощи методов `AddButton` и `AddMenu` объекта `Collection`. Доступ к элементам коллекции может производиться как по индексу, так и по символьному ключу, который задается для каждого элемента коллекции в момент его создания.

12. Добавить к коду приложения обработчик события `Form_Load()`. Код обработчика приведен в листинге 3.13. В него добавлены комментарии, которые можно не вводить в код приложения. Они потребуются лишь для удобства рассмотрения этого фрагмента.

### Листинг 3.13

```
Rem 1
Dim mnuFile As MenuBarItem
Rem 2
Dim mnuSep, mnuClrl, mnuClr2, mnuClr3 As
MenuBarButtor
Rem 3
Set mnuFile = MenuBar1.Controls.AcШе-
пиС'Файл", «mnuFile»)
Rem 4
mnuFile.Items.Add 1, «itFont», «Редакти-
рование шрифта»
Rem 5
mnuFile.Items.Item(1). SubItems.Add 1,
```

```
«itBold», «Полужирный»
    mnuFile.Items.Item(1). SubItems.Add 2,
«itItal», «Наклонный»
    mnuFile.Items.Item(1). SubItems.Add 3,
«itSize», «Размер»
    Rem 6
    mnuFile.Items.Item(1). SubItems.Item(3).
SubItems.Add 1, «it8», «* 8»
    mnuFile.Items.Item(1). SubItems.Item(3).
SubItems.Add 2, «it10», «10»
    mnuFile.Items.Item(1). SubItems.Item(3).
SubItems.Add 3, «it12», «12»
    Rem 7
    mnuFile.Items.Add 2, "", ""
    mnuFile.Items.Item(2). Style =
mbrMenuSeparator
    Rem 8
    mnuFile.Items.Add 3, «itOpen», «Открыть»
    Rem 9
    mnuFile.Items.Add 5, "", ""
    mnuFile.Items.Item(5). Style =
mbrMenuSeparator
    Rem 10
    mnuFile.Items.Add 6, «itExit», «Завер-
шить работу»
    Rem 11
```

```
Set mnuSep =
MenuBar1.Controls.AddButton («mnuSep»)
mnuSep.Style = mbrSeparator
Rem 12
Set mnuClr1 =
MenuBar1.Controls.AddButton ("1")
mnuClr1.Style = mbrButtonGroup
mnuClr1.Caption = " 1»
Set mnuClr2 =
MenuBar1.Controls.AddButton ("2")
mnuClr2.Style = mbrButtonGroup
mnuClr2.Caption = " 2»
Set mnuClr3 =
MenuBar1.Controls.AddButton ("3")
mnuClr3.Style = mbrButtonGroup
mnuClr3.Caption = " 3»
Rem 13
Frame1.Top = -200
List1.Clear
End Sub
```

13. Теперь этот листинг надо детально рассмотреть. В качестве ссылок будут использоваться номера комментариев в тексте.

1) Объявляется переменная типа MenuBarItem. Переменная одна, значит, на панели команд устройства будет

Один пункт меню.

# Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.