

Visual C++ .NET



Создание основных типов приложений

Концепция “документ/представление”

Расширенные возможности Visual C++

Разработка справочной системы

Создание библиотек динамической компоновки



Дискета содержит файлы примеров и описание классов и функций Visual C++ .NET

Мощная среда программирования для разработки Windows-приложений

Николай Секунов

САМОУЧИТЕЛЬ
VISUAL C++ .NET

Санкт-Петербург

«БХВ-Петербург»

УДК 681.3.06

Книга посвящена методам объектно-ориентированного программирования для 32-разрядных операционных систем Windows. Рассмотрен широкий круг вопросов разработки диалоговых и многооконных приложений. Обсуждаются вопросы создания различных типов справочных систем приложений, их русификации и преобразования из одного типа в другой. Большое внимание уделено применению механизма исключений для обработки ошибок, работе с шаблонами, многозадачности и взаимодействию потоков. Отдельная глава посвящена созданию библиотек динамической компоновки (DLL). Приведен обширный справочный материал по функциям и классам Visual C++.

Для начинающих программистов

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зав. редакцией	<i>Анна Кузьмина</i>
Редактор	<i>Владислав Борисов</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Секунов Н. Ю.

Самоучитель Visual C++ .NET. — СПб.: БХВ-Петербург, 2002. — 736 с.: ил.

ISBN 5-94157-032-5

© Н. Ю. Секунов, 2002

© Оформление, издательство "БХВ-Петербург", 2002

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.02.02.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 59,34.

Тираж 4000 экз. Заказ №

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.1.953.П.950.3.99 от 01.03.1999 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ФГУП ордена Трудового Красного Знамени "Техническая книга"
Министерства Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.
198005, Санкт-Петербург, Измайловский пр., 29.

Содержание

Введение	1
Для кого предназначена эта книга?	2
Структура данной книги	3
Соглашения, принятые в данной книге	7
Требования к аппаратным средствам и к программному обеспечению	9
Часть I. СОЗДАНИЕ ПРОСТЕЙШИХ ПРИЛОЖЕНИЙ	11
Глава 1. Использование мастера создания приложений	13
Диалоговое приложение	13
Многооконное приложение Windows	17
Другие виды приложений MFC	19
Консольное приложение	20
Глава 2. Классы приложений, документов и представлений	25
Класс документа	25
Класс представления	29
Класс приложения	35
Дизайн элементов управления и системный реестр	39
Шаблон документа	40
Создание окон	41
Работа с несколькими типами документов	43
Глава 3. Диалоговые окна и простейшие элементы управления	55
Диалоговое окно	55
Формирование ресурсов диалогового окна	55
Внесение изменений в класс диалогового окна	69
Вкладки и мастера	82
Создание вкладок диалогового окна	82
Создание мастера	87
Некоторые модификации окна мастера	93
Часть II. ПРОГРАММИРОВАНИЕ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ	99
Глава 4. Классы элементов управления	101
Класс списка	101

Классы линейного регулятора и линейного индикатора.....	110
Создание пользовательского линейного индикатора.....	119
Обработка даты и времени.....	125
Глава 5. Сообщения и команды	136
Обработка сообщений	136
Карта сообщений.....	138
Сообщения в Windows 3.x.....	141
Сообщения в Win32.....	141
Сообщения, посылаемые всеми новыми элементами управления Windows.....	142
Обработка извещений	143
Обработка отраженных сообщений.....	145
Макросы карты сообщений и заготовки функций обработки отраженных сообщений.....	146
Пример обработки отраженных сообщений.....	148
Использование карты сообщений приложением	153
Создание функций обработки сообщений.....	154
Диалоговое окно <i>Properties</i>	154
Список сообщений	159
Обновление команд	159
Глава 6. Вывод информации на экран	162
Интерфейс графических устройств (GDI)	162
Контекст устройства	163
Отображение текста	165
Шрифты	166
Вывод текста	173
Форматирование текста	173
Вывод текста	175
Программа вывода текста.....	176
Перерисовка окна	179
Использование перьев	181
Работа с кистью.....	185
Использование диалогового окна для настройки параметров.....	189
Работа с битовыми образами	195
Аппаратно-зависимые битовые образы	195
Аппаратно-независимые битовые образы	202
Глава 7. Работа с файлами документов.....	208
Работа с архивом	208
Непосредственное чтение и запись файлов.....	218
Использование объектов класса <i>CFile</i> при работе с классом <i>CArchive</i>	218
Автономное использование класса <i>CFile</i>	222
Работа с системным реестром.....	232

Глава 8. Работа с текстовыми документами	236
Создание простейшего текстового редактора	236
Создание более сложного редактора	238
Форматирование документов	244
Задание пользовательского шрифта	250
Глава 9. Панели инструментов и строка состояния	254
Работа с панелью инструментов	255
Удаление кнопок из панели инструментов	256
Добавление кнопок в панель инструментов	258
Работа со строкой состояния	272
Глава 10. Печать документов и организация прокрутки в окне	285
Организация прокрутки в окне	286
Режимы отображения	293
Распечатка и предварительный просмотр	294
Работа с окном предварительного просмотра печати	294
Распечатка многостраничного документа	299
Использование функций библиотеки MFC при печати	309
Часть III. ОСОБЕННОСТИ ПРОГРАММИРОВАНИЯ В СРЕДЕ VISUAL C++	315
Глава 11. Исключения, шаблоны и новые возможности Visual C++	317
Работа с исключениями	317
Аргументы исключений	318
Механизмы исключений Visual C++	320
В каких случаях следует вызывать исключения	322
Перехват и уничтожение исключений	324
Уничтожение объектов в исключениях	326
Вызов исключений из функций пользователя	328
Преобразование макросов исключений в операторы C++	329
Совместное использование макросов и операторов C++	333
Шаблоны	335
Понятие шаблона	336
Шаблоны функций	338
Шаблоны классов	341
Работа с классами коллекций	344
Виды классов коллекций	344
Классы коллекций, использующие шаблоны	347
Классы коллекций, не использующие шаблоны	352
Доступ к элементам классов коллекций	353
Удаление элементов классов коллекций	355
Использование классов коллекций	357

Глава 12. Многозадачность на основе потоков Windows.....	360
Независимая работа потоков	361
Создание рабочего потока.....	361
Создание интерфейсных потоков.....	363
Прекращение работы потока	365
Взаимодействие между потоками	366
Взаимодействие между потоком и приложением	367
Использование классов синхронизации	367
Простейший пример работы с потоками	369
Более сложный пример работы с потоками.....	377
Глава 13. Справка в приложении	387
Описание справочной системы приложения.....	387
Способы доступа к справочной системе	388
Способы представления справочной информации	389
Формы представления справочной информации	403
Программирование справочной системы	403
Компоненты справочной системы	404
Обработка сообщений справочной системы	407
Русификация файла ресурсов	412
Создание системы командной справки	437
Создание системы контекстной справки.....	441
Подготовка справочных текстов.....	446
Русификация справочной системы приложения	446
Добавление новых тем	473
Форматирование текстовых файлов справки	477
Использование графики и гиперграфических ссылок	480
Использование макросов справочной системы	481
Внесение изменений в оглавление справочной системы	481
Использование справочной системы HTML	485
Преобразование справочной системы приложения	486
Диалоговое окно <i>О программе</i>	511
Глава 14. Отладка приложения.....	513
Средства отладки, предоставляемые интерфейсом пользователя.....	513
Точки останова	515
Анализ исполнения программы.....	519
Настройка уровня предупреждений транслятора	524
Программные средства отладки	526
Макросы <i>ASSERT</i> и <i>TRACE</i>	526
Отладочные функции.....	528
Глобальные диагностические функции	529
Отладка распространяемой версии приложения	530
Устранение утечки памяти	531
Основные причины возникновения утечек памяти	531
Отладочные версии операторов <i>new</i> и <i>delete</i>	535

Глава 15. Создание пользовательской библиотеки	538
Использование библиотек динамической компоновки	539
Главная функция библиотеки динамической компоновки	539
Экспорт и импорт функций	540
Неявная компоновка	542
Явная компоновка	542
Использование для работы с библиотеками динамической компоновки	543
Регулярные библиотеки динамической компоновки MFC	544
Библиотеки динамической компоновки расширения MFC	545
Функция <i>DllMain</i>	547
Экспорт классов и функций	548
Передача ресурсов	549
Использование файлов DEF	550
Пример создания и использования библиотеки динамической компоновки	555
Создание регулярной библиотеки динамической компоновки	555
Создание библиотеки динамической компоновки расширения MFC	562
Создание демонстрационного приложения	579
Описание созданных проектов	583
Глава 16. Создание простейшего приложения Internet	587
Классы WinInet	587
Чтение Web-страницы	588
Часть IV. ПРИЛОЖЕНИЯ.....	599
Приложение 1. Объектно-ориентированное программирование и классы	601
Обзор объектно-ориентированных методов программирования	601
Инкапсуляция	602
Наследование	603
Полиморфизм	604
Классы	605
Классы как типы данных	605
Файлы заголовков и файлы реализации	605
Когда следует, а когда не следует использовать классы	612
Перегрузка функций и операторов	612
Использование виртуальных функций	614
Область действия класса	615
Приложение 2. Интерфейс пользователя Visual C++	617
Первая страница	618
Панели инструментов	620
Панель инструментов <i>Standard</i>	623
Система меню	625
Меню <i>File</i>	625
Команда <i>File / New</i>	625
Команда <i>File / Open</i>	628

Команда <i>File / Close</i>	630
Команды <i>File / Add New Item</i> и <i>File / Add Existing Item</i>	630
Команда <i>File / Add Project</i>	630
Команда <i>File / Open Solution</i>	631
Команда <i>File / Close Solution</i>	631
Команда <i>File / Save</i>	632
Команда <i>File / Save As</i>	632
Команда <i>File / Advanced Save Options</i>	632
Команда <i>File / Save All</i>	633
Команда <i>File / Source Control</i>	633
Команда <i>File / Page Setup</i>	634
Команда <i>File / Print</i>	634
Команда <i>File / Recent Files</i>	635
Команда <i>File / Recent Projects</i>	635
Команда <i>File / Exit</i>	635
Меню <i>Edit</i>	636
Команда <i>Edit / Undo</i>	636
Команда <i>Edit / Redo</i>	637
Команда <i>Edit / Cut</i>	637
Команда <i>Edit / Copy</i>	638
Команда <i>Edit / Paste</i>	638
Команда <i>Edit / Delete</i>	638
Команда <i>Edit / Select All</i>	638
Команда <i>Edit / Find and Replace</i>	638
Команда <i>Edit / Go To</i>	647
Команда <i>Edit / Insert File As Text</i>	647
Команда <i>Edit / Advanced</i>	648
Команда <i>Edit / Bookmarks</i>	650
Команда <i>Edit / Outlining</i>	651
Команда <i>Edit / IntelliSense</i>	655
Меню <i>View</i>	657
Команда <i>View / Open</i>	657
Команда <i>View / Open With</i>	657
Команда <i>View / Solution Explorer</i>	657
Команда <i>View / Class View</i>	659
Команда <i>View / Server Explorer</i>	659
Команда <i>View / Resource View</i>	659
Команда <i>View / Properties Window</i>	659
Команда <i>View / Toolbox</i>	661
Команда <i>View / Web Browser</i>	661
Команда <i>View / Other Windows</i>	661
Команда <i>View / Show Tasks</i>	662
Команда <i>View / Toolbars</i>	662
Команда <i>View / Full Screen</i>	662
Команда <i>View / Navigate Backward</i>	664
Команда <i>View / Navigate Forward</i>	664
Команда <i>View / Property Pages</i>	664

Меню <i>Project</i>	664
Команда <i>Project / Add Class</i>	666
Команда <i>Project / Add Function</i>	666
Команда <i>Project / Add Variable</i>	666
Команда <i>Project / Add Resource</i>	668
Команда <i>Project / Add New Item</i>	668
Команда <i>Project / Add Existing Item</i>	669
Команда <i>Project / New Folder</i>	669
Команда <i>Project / Unload Project</i>	669
Команда <i>Project / Add Web Reference</i>	670
Команда <i>Project / Set as StartUp Project</i>	670
Команда <i>Project / Properties</i>	671
Меню <i>Build</i>	671
Команда <i>Build / Build</i>	671
Команда <i>Build / Rebuild All</i>	671
Команда <i>Build / Clean</i>	672
Команда <i>Build / Batch Build</i>	672
Команда <i>Build / Configuration Manager</i>	673
Команда <i>Build / Compile</i>	673
Команда <i>Build / Deploy</i>	673
Меню <i>Debug</i>	673
Команда <i>Debug / Windows</i>	673
Команда <i>Debug / Start</i>	675
Команда <i>Debug / Break All</i>	675
Команда <i>Debug / Stop Debugging</i>	675
Команда <i>Debug / Detach All</i>	675
Команда <i>Debug / Restart</i>	675
Команда <i>Debug / Apply Code Changes</i>	675
Команда <i>Debug / Processes</i>	675
Команда <i>Debug / Exceptions</i>	676
Команда <i>Debug / Step Into</i>	676
Команда <i>Debug / Step Over</i>	676
Команда <i>Debug / Step Out</i>	677
Команда <i>Debug / Quick Watch</i>	677
Команда <i>Debug / New Breakpoint</i>	677
Команда <i>Debug / Clear All Breakpoints</i>	679
Команда <i>Debug / Disable Breakpoint</i>	679
Команда <i>Debug / Save Dump As</i>	679
Меню <i>Tools</i>	679
Команда <i>Tools / Debug Processes</i>	680
Команда <i>Tools / Connect to Database</i>	680
Команда <i>Tools / Connect to Server</i>	680
Команда <i>Tools / Customize Toolbox</i>	680
Команда <i>Tools / Add-in Manager</i>	681
Команда <i>Tools / Build Comment Web Pages</i>	683
Команда <i>Tools / Macros</i>	684

Команда <i>Tools / ActiveX Control Test Container</i>	684
Команда <i>Tools / Create GUID</i>	684
Команда <i>Tools / Error Lookup</i>	686
Команда <i>Tools / MFC/ATL Trace Tool</i>	686
Команда <i>Tools / OLE/COM Object Viewer</i>	687
Команда <i>Tools / Spy++</i>	687
Команда <i>Tools / External Tools</i>	687
Команда <i>Tools / Customize</i>	689
Команда <i>Tools / Options</i>	689
Меню <i>Window</i>	691
Команда <i>Window / New Window</i>	691
Команда <i>Window / Split</i>	691
Команда <i>Window / Dockable</i>	692
Команда <i>Window / Hide</i>	692
Команда <i>Window / Floating</i>	692
Команда <i>Window / Auto Hide</i>	692
Команда <i>Window / Auto Hide All</i>	692
Команда <i>Window / New Horizontal Tab Group</i>	694
Команда <i>Window / New Vertical Tab Group</i>	694
Команда <i>Window / Move to Next Tab Group</i>	695
Команда <i>Window / Move to Previous Tab Group</i>	695
Команда <i>Window / Close All Documents</i>	695
Список открытых окон.....	696
Команда <i>Window / Windows</i>	696
Меню <i>Help</i>	696
Команда <i>Help / Dynamic Help</i>	697
Команда <i>Help / Contents</i>	698
Команда <i>Help / Index</i>	698
Команда <i>Help / Search</i>	699
Команда <i>Help / Index results</i>	699
Команда <i>Help / Search results</i>	700
Остальные команды меню.....	700
Окна Visual Studio.NET.....	701
Окно <i>Solution Explorer</i>	701
Окно <i>Class View</i>	701
Окно <i>Properties</i>	704
Окно <i>Watch</i>	704
Окно <i>Breakpoints</i>	705
Приложение 3. Описание дискеты	707
Предметный указатель	709

Введение

Windows изначально разрабатывалась как графическая среда для "домохозяйек", которая впоследствии превратилась в операционную систему для "домохозяйек", сохранив при этом основной подход к данной категории программных продуктов. Отличительной чертой подобных программных продуктов является то, что они устанавливаются на компьютер до его передачи пользователю специалистом высокой квалификации, способным устранить все конфликты, возникающие в установленной программной конфигурации. После того как компьютер передается пользователю, на него не может быть установлена ни одна новая программа, поскольку это может привести к краху всей системы. И, действительно, о какой надежности может идти речь в операционной системе, при установке в которую нового приложения может быть замещена добрая половина системных библиотек. Следствием данного подхода является недопустимость создания в данной операционной системе новых программ.

Поскольку писать программы под Windows все-таки приходится, для этого лучше всего использовать систему программирования, разработчики которой хорошо знакомы с используемой операционной системой и которая требует от программиста минимального вмешательства в процесс написания программ. Этим требованиям как нельзя лучше удовлетворяет визуальная среда программирования Visual Studio.NET, разработанная корпорацией Microsoft. Наиболее весомым доводом в пользу использования именно этой системы является тот факт, что она разработана в той же корпорации, что и операционная система, в которой ей предстоит работать.

Однако всегда следует помнить, что даже разработка операционной системы и среды программирования в одной корпорации не гарантирует их нормальной совместной работы, особенно под управлением новых версий операционных систем, которые отсутствовали на момент создания среды разработки. Например, среда программирования Visual Studio 6.0, нормально работавшая под управлением Windows 95, под управлением Windows 98 уже работает некорректно. После завершения работы с созданным ею простейшим диалоговым приложением, обрабатывающим мультимедийный файл и не работающим с панелями инструментов, помещение указателя мыши на панель управления (панель инструментов задачи или Панель задач рабочего стола) считается недопустимой операцией, после которой необходимо прекратить выполнение задачи. Отсюда следуют два вывода: во-первых, среду программирования нужно обновлять одновременно с операционной системой и, во-вторых, нужно пользоваться продуктами Microsoft, поскольку информация об операционной системе является недоступной разработчикам других сред программирования и их продукты будут работать крайне ненадежно.

Среда программирования Visual Studio.NET, являющаяся мощным средством для разработки 32-разрядных приложений в операционных системах Windows 95,

Windows 98 и Windows NT. Она содержит полный набор средств для быстрой разработки Internet-приложений масштаба предприятия, средства упрощения совместной разработки приложений группой программистов и средства распространения созданных приложений. Этот достаточно сложный продукт позволяет создавать намного более объемные и более сложные приложения, чем его предшественники, разработанные для 16-разрядного Windows или для DOS.

В принципе, программу любой сложности можно написать на любом языке программирования и с использованием любой среды программирования. Это, конечно, так, но хорошая среда программирования отличается от плохой тем, что в ней на программирование той же самой задачи будет затрачено меньше времени, а получившаяся в результате программа будет работать быстрее и надежнее. Для создания подобной среды программирования необходимо досконально изучить среду, в которой будет работать скомпилированная программа, а кто изучил эту среду лучше ее разработчика. При этом следует особенно учесть, что вместо подробного описания корпорация Microsoft предоставила обществу схематический план своего творения.

Пользователь, скорее всего, оценит возможности, предоставляемые ему Visual Studio.NET. Основой этой среды является новый компонентно-ориентированный язык программирования C#, обеспечивающий надежную совместную работу компонентов, написанных с использованием различных языков программирования. Однако наиболее полно возможности этого языка используются при работе с компонентами, написанными на языке C++. Язык Visual C++, включенный в Visual Studio.NET, сохранил все основные особенности своей предыдущей версии 6.0, в том числе связь с библиотекой базовых классов Microsoft Foundation Classes или MFC и использование мастеров, позволяющих получать заготовки больших фрагментов исходного кода программы путем выбора соответствующих опций в панелях специальным образом организованных диалоговых окон.

Для кого предназначена эта книга

Данная книга посвящена языку программирования Visual C++, а не C++. В ней будут рассмотрены не конструкции данного языка, а классы библиотеки MFC, позволяющие выполнить ту или иную задачу, встающую перед разработчиком полноценного приложения.

В настоящее время остается еще достаточно много программистов, предпочитающих работать в среде DOS. Работа в этой среде позволяет более эффективно использовать системные ресурсы и создавать компактные и быстродействующие приложения. Доказательством этому служит активное использование подобных приложений самой корпорацией Microsoft. Приложения, работающие в среде DOS, называются в среде Windows консольными приложениями. В Visual Studio.NET имеются средства для разработки этого типа приложений, правда, не такие удобные, как в предыдущей версии данного продукта.

Несмотря на то, что Microsoft позиционирует среду Visual Studio.NET как конкурента среды программирования C++ Builder фирмы Inprise, успешно используемого для разработки приложений, работающих с базами данных, в ней еще осталось много свойств от ее предыдущей версии, которая была хорошо приспособлена для создания небольших приложений Windows. Данная книга посвящена описанию возможностей, предоставляемых данной средой в указанной области. Другими потенциальными читателями этой книги являются программисты, имеющие опыт программирования в среде Windows 3.x и продолжающие его использовать при работе в Visual C++, не желая изучать особенности данной среды программирования. Конкуренция и требования рынка уже заставляют их не просто использовать некоторые заготовки, предоставляемые им мастером AppWizard, но и более полно применять описанные в данной книге возможности, предоставляемые им библиотекой MFC.

У каждого пользователя среды программирования Visual C++, пытавшегося разобраться в ней самостоятельно, накопилось множество вопросов, и он надеется получить в одной книге ответы на большинство из них. Как показывает опыт, уровень подготовки, даже у достаточно опытных программистов, существенно отличается: некоторые до сих пор писали исключительно в среде DOS и теперь, перейдя к работе в среде Windows, нуждаются в простейшем компиляторе для своих задач, не использующих и не требующих использования сложного графического интерфейса, другие работали в среде Windows 3.x и не привыкли следовать при написании программ требованиям, предъявляемым 32-разрядными приложениями, но хотели бы это делать, и, наконец, любому программисту нужен достаточно подробный справочник, в котором он мог бы найти решение стандартных проблем, возникающих при программировании.

Поскольку нельзя объять необъятное, данная книга не является учебником по алгоритмическому языку C++ и операционной системе Windows. Предполагается, что читатель уже написал, по крайней мере, пару программ на этом языке. Однако, как выяснилось, многие программисты, работавшие под DOS и, даже, использовавшие библиотеку классов Turbo Vision фирмы Borland, имеют о классах достаточно смутное представление. Поэтому в данную книгу включена глава, посвященная концепции объектного программирования и классам. Те, кто считает, что он достаточно хорошо знаком с этим вопросом, может пропустить эту главу.

Структура данной книги

Поскольку предполагается, что читатель данной книги может не иметь опыта программирования в среде Windows, то в данной книге содержится весь необходимый материал, позволяющий начать программирование приложений практически с нуля. В приведенном ниже списке разделов и глав дано краткое их описание, позволяющее пользователю лучше ориентироваться в структуре книги.

□ Часть I. Создание простейших приложений.

В данной части рассмотрено создание простейших приложений, использующих основные элементы интерфейса Windows. Здесь же рассмотрена кон-

цепция Документ/Представление, лежащая в основе программирования с использованием библиотеки MFC.

- Глава 1. Использование мастера создания приложений.

В данной главе рассмотрен мастер создания приложений, позволяющий автоматически создать заготовку приложения, определив в нем некоторые ключевые его особенности. Здесь же рассмотрено создание консольного приложения, работающего в окне MS DOS.

- Глава 2. Классы приложений, документов и представлений.

В данной главе рассмотрена концепция Документ/Представление. Кроме того, в ней описаны классы шаблонов документов.

- Глава 3. Диалоговые окна и простейшие элементы управления.

В данной главе рассмотрено создание диалоговых окон с использованием редактора ресурсов и описано включение в них простейших элементов управления.

□ Часть II. Программирование интерфейса пользователя.

В данной части содержится описание того, каким образом пользователь может выводить на экран необходимую информацию и управлять работой приложения.

- Глава 4. Классы элементов управления.

В данной главе рассмотрены достаточно сложные элементы управления, работа с которыми требует создания объектов классов данных элементов управления и вызова содержащихся в них методов обработки информации.

- Глава 5. Сообщения и команды.

В данной главе рассмотрен механизм обработки сообщений, используемый в операционной системе Windows для обеспечения взаимодействия различных процессов и потоков в приложении.

- Глава 6. Вывод информации на экран.

В данной главе рассмотрены принципы работы интерфейса графических устройств Windows (GDI).

- Глава 7. Работа с файлами документов.

В данной главе содержится описание работы с файловой системой Windows.

- Глава 8. Работа с текстовыми документами.

Операционная система Windows создавалась преимущественно для работы с текстовыми документами, и оптимизирована исключительно для выполнения данной задачи. Поэтому, вопросы создания приложения, работающего с текстовой информацией, вынесены в отдельную главу.

- Глава 9. Панели инструментов и строка состояния.

Панели инструментов и строка состояния являются неизменным атрибутом любого законченного приложения Windows. В данной главе описаны

способы создания и программирования работы с панелями инструментов и строкой состояния.

- Глава 10. Печать документов и организация прокрутки в окне.

Печать документов и прокрутка изображения в окне большей частью реализуются с использованием стандартных средств среды программирования Visual C++, однако, для обеспечения работы данных функций в приложении необходимо произвести его настройку. Этому вопросу и посвящена данная глава.

□ Часть III. Особенности программирования в среде Visual C++.

Средств, описанных в первых двух частях данной книги, достаточно для создания работоспособного приложения. Однако среда программирования Visual C++ предоставляет пользователю возможности, позволяющие уменьшить размеры исходных текстов программ, повысить их надежность и рационально распределить предоставляемые приложению вычислительные ресурсы. Описание этих возможностей содержится в данной части.

- Глава 11. Исключения, шаблоны и новые возможности Visual C++.

В данной главе рассматриваются новые возможности, предоставляемые средой программирования Visual C++. Исключения представляют собой механизм, существенно упрощающий процесс обработки ошибок. Без использования этого механизма практически невозможно создать надежно работающее приложение. Шаблоны представляют собой метод задания функций обработки для переменных абстрактных классов. Это позволяет существенно сократить количество практически идентичных по тексту функций, выполняющих одну и ту же операцию для переменных разных типов. Использование шаблонов позволяет сократить размер исходного текста программ и обеспечивает гарантированное внесение изменений во все функции обработки различных типов данных, выполняющих над ними одну и ту же операцию.

- Глава 12. Многозадачность на основе потоков Windows.

Поток Windows представляет собой основную программную единицу, которой операционная система Windows предоставляет квант процессорного времени. Создание нескольких потоков в одном приложении позволяет более рационально использовать выделяемое данному приложению процессорное время. Создание фоновых потоков и правильная расстановка приоритетов позволяет обеспечить быструю реакцию приложения на запрос пользователя при максимально возможной загрузке процессора.

- Глава 13. Справка в приложении.

Любое серьезное приложение, независимо от того, создается ли оно для внутреннего использования или для продажи, должно содержать обширную справочную систему, позволяющую получить справку по любому вопросу, связанному с данным приложением. Для коммерческих приложений, а также для приложений, рассчитанных на пользователя со слабым

знанием английского языка, встает вопрос русификации приложения и содержащейся в нем справочной системы. Поэтому в данной главе рассмотрены вопросы не только связанные с созданием справочной системы, но и с ее русификацией. Параллельно рассмотрен вопрос русификации всего приложения.

- Глава 14. Отладка приложения.

Отладка приложения является неременным этапом его создания. Практически ни одно достаточно серьезное приложение не заработало безошибочно с первого раза. В данной главе рассмотрены возможности, предоставляемые средой программирования Visual C++ для отладки приложения.

- Глава 15. Создание пользовательской библиотеки.

Любой программист, написавший хотя бы одно полноценное приложение, хочет использовать его отдельные компоненты в других своих приложениях. То же самое относится и к любому специалисту в некоторой области, которому хотелось бы применять в своих приложениях некоторый инструментарий, содержащий стандартные методы данной области. Все эти вопросы могут быть решены с использованием пользовательских библиотек, принципы создания которых описаны в данной главе.

- Глава 16. Создание простейшего приложения Internet.

Microsoft позиционирует Visual Studio.NET, прежде всего, как средство создания многоуровневых систем в Internet. Поэтому в данной главе приведено краткое описание классов WinInet и продемонстрировано их использование для создания простейшего приложения для чтения Web-страниц.

□ Приложения

В приложениях изложены вопросы, не связанные напрямую с программированием приложений, но изучение которых может способствовать успешной работе в Visual C++.

- Приложение 1. Объектно-ориентированное программирование и классы.

Как уже отмечалось выше, опыт показывает, что не все программисты, использующие классы, хорошо понимают их назначение. Поскольку использование библиотеки MFC построено на использовании классов и для эффективного ее использования пользователь должен хорошо разбираться в основах ООП, то в данном приложении дан краткий обзор данной концепции и связанного с ней понятия класса.

- Приложение 2. Интерфейс пользователя Visual C++.

Visual C++ представляет собой достаточно сложную среду программирования. Хотя ее интерфейс и задумывался как "интуитивно ясный", но многие функции данной среды таковыми не являются. Кроме того, пользователь может и не догадываться насколько ему необходима данная функция, пока он не прочтет о ней и не попробует ею воспользоваться. Данное приложе-

ние содержит описание интерфейса пользователя, достаточное для начала работы с ним абсолютно неподготовленного пользователя.

- Приложение 3. Описание дискеты.

Помимо самораспаковывающегося архива SAMPLES.EXE, на дискете находится файл с описанием классов и функций Visual C++. Содержимое этого файла представляет собой выборочный перевод информации, находящейся в библиотеке MSDN. Необходимость размещения данного файла на дискете обусловлена тем, что далеко не все пользователи настолько хорошо знают английский язык, чтобы быть уверенными в том, что они поймут все тонкости использования функций, описанных в данной библиотеке. В этом файле пользователь может найти полное описание большинства функций, используемых в программах, приведенных в качестве примеров.

Все функции, классы и структуры в данном файле расположены в алфавитном порядке. Сначала идет описание глобальных функций, а затем описание классов и структур.

Соглашения, принятые в данной книге

В данной книге использовалось специальное форматирование текста для выделения некоторых текстов или фрагментов текста. Ниже приведены основные принципы выделения текста.

- Исходные тексты фрагментов программ, представляющие собой одну или более строк текста, выделяются специальным шрифтом, как это показано ниже:

```
void CMainFrame::Dump(CDumpContext& dc) const
{
    CMDIFrameWnd::Dump(dc);
}
```

- Когда на эти фрагменты программ имеются ссылки из различных фрагментов текста, они оформляются в виде листинга, как показано ниже.

Листинг 1. Заголовок листинга

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CMDIFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    if (!m_wndToolBar.CreateEx(this, TBSTYLE_FLAT, WS_CHILD | WS_VISIBLE |
        CBRS_TOP | CBRS_GRIPPER | CBRS_TOOLTIPS | CBRS_FLYBY |
        CBRS_SIZE_DYNAMIC) || !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
```

```

{
    TRACE0("Failed to create toolbar\n");
    return -1; // fail to create
}

if (!m_wndStatusBar.Create(this) ||
    !m_wndStatusBar.SetIndicators(indicators,
    sizeof(indicators)/sizeof(UINT))
{
    TRACE0("Failed to create status bar\n");
    return -1; // fail to create
}

// TODO: Delete these three lines if you don't want the toolbar to
// be dockable
m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
EnableDocking(CBRS_ALIGN_ANY);
DockControlBar(&m_wndToolBar);

return 0;
}

```

- ❑ Если в тексте встречается имя класса, функции, имя типа переменной или фрагмент текста длиной менее строки, он выделяется специальным шрифтом. Например: `int`, `CMainFrame`, `DockControlBar` и т. д.
- ❑ Имена функции, обычно, даются в полной форме. Это значит, что если функция является членом класса, то перед ней указывается имя класса, членом которого она является. При этом указывается имя того класса в иерархии, в котором впервые появилась данная функция. Например: `Object::Serialize`. Если перед именем функции не стоит имя класса, то это означает, что это либо глобальная функция, либо рассматриваемая в настоящее время функция пользовательского класса, либо часто встречающаяся функция, полное описание которой было только что приведено. Это позволяет сравнительно просто отыскать описание данной функции на дискете, прилагаемой к этой книге.
- ❑ Заголовок диалогового окна, имя кнопки или команды меню выделяются полужирным шрифтом. Например: **Open**, команда меню **File | New** и т. д.
- ❑ Имя клавиши заключается в угловые скобки (<>). Например: <Ctrl>, <F5> и т. д.
- ❑ Если требуется одновременно нажать несколько клавиш, они объединяются знаком (+). Например: <Ctrl>+<Alt>+.
- ❑ При первом появлении нового термина он выделяется курсивом. Например: *новый термин*.
- ❑ Ссылка на другую главу содержит ее название и номер. Например: см. главу 5.

Требования к аппаратным средствам и к программному обеспечению

Для работы в среде программирования Visual Studio.NET требуется следующее аппаратное и программное обеспечение:

- IBM совместимый PC с процессором Intel Pentium II 450 МГц или лучшим;
- не менее 256 Мбайт оперативной памяти;
- жесткий диск объемом не менее 3 Гбайт;
- дисплей SVGA и соответствующий видеоадаптер, обеспечивающие разрешение не менее 800×600 точек; дисковод CD-ROM;
- совместимая с Microsoft мышь; операционная система Windows 98, Windows ME, Windows NT или Windows 2000.

Естественно, эти требования являются минимальными и работа в такой конфигурации не доставит вам большого удовольствия.



ЧАСТЬ I

СОЗДАНИЕ ПРОСТЕЙШИХ ПРИЛОЖЕНИЙ

Глава 1. Использование мастера создания приложений

Глава 2. Классы приложений, документов и представлений

Глава 3. Диалоговые окна и простейшие элементы управления

Глава 1



Использование мастера создания приложений

Среда программирования Visual Studio.NET используется не только для редактирования, компиляции и отладки программ, но и для генерации их заготовок. Эти заготовки представляют собой работоспособные программы, реализующие основные элементы интерфейса Windows, необходимые в любой пользовательской программе, и, естественно, не выполняющие никакой обработки. Мастер приложений позволяет создать такую заготовку за несколько минут, однако, чтобы получить на ее основе полноценное коммерческое приложение, уйдут недели и месяцы работы.

Программисту, начинающему работу в среде Visual Studio.NET, необходимо, прежде всего, ознакомиться с простейшими типами приложений, генерируемых мастерами. К ним следует отнести приложения, генерируемые мастером MFC (Microsoft Foundation Classes) Application Wizard. Эти типы приложений являются базовыми для языка Visual C++ и позволяют ознакомиться с его основными особенностями.

Библиотека MFC является одной из основных библиотек языка Visual C++. Она используется для работы с элементами управления окон Windows и обработки поступающих от них сообщений. Применение данной библиотеки существенно облегчает написание программ на языке Visual C++.

Мастер создания приложений MFC позволяет создавать четыре вида приложений:

- диалоговые приложения;
- однооконные приложения;
- многооконные приложения;
- приложения с несколькими документами верхнего уровня.

Первые три типа приложений хорошо знакомы программистам, использовавшим предыдущие версии языка Visual C++.

Диалоговое приложение

Диалоговые приложения рассматриваются корпорацией Microsoft как простейший тип приложения, которое не должно иметь никаких меню, кроме системного, а также не может открывать или сохранять информацию в файле. Оба эти ограничения могут быть обойдены, однако при этом возникают определенные сложности. Другое неудобство, возникающее при работе с диалоговыми прило-

жениями, связано с тем, что, по мнению разработчиков Visual C++, считывание из него информации должно производиться только при его закрытии, что исключает взаимодействие его элементов управления. Поэтому очень часто возникают сложности при необходимости вывести в одном элементе управления изменения, произведенные в другом элементе управления. Особые сложности возникают в том случае, если изменения вносятся в текстовое поле.

Диалоговое приложение является полноценным приложением Windows, в котором может быть использована полноценная поддержка технологии ActiveX, что позволяет решить вопрос с передачей данных в приложение и из приложения. С другой стороны, это приложение может быть оформлено в виде самостоятельного модуля, что позволит задать ему при выполнении уровень приоритета, недоступный для потока основного приложения. Поэтому диалоговое приложение может использоваться для реализации критичных ко времени исполнения фрагментов программы.

Текст диалогового приложения можно найти в каталоге Dialog на прилагаемой к данной книге дискете.

Чтобы самостоятельно создать заготовку диалогового приложения с использованием мастера создания приложений MFC:

1. Закройте все открытые проекты и файлы и выберите команду **File | New | Project** (Файл | Создать | Проект) или нажмите кнопку **New Project** (Новый проект) на панели инструментов **Standard** (Стандартная). Появится диалоговое окно **New Project** (Новый проект), изображенное на рис. 1.1.

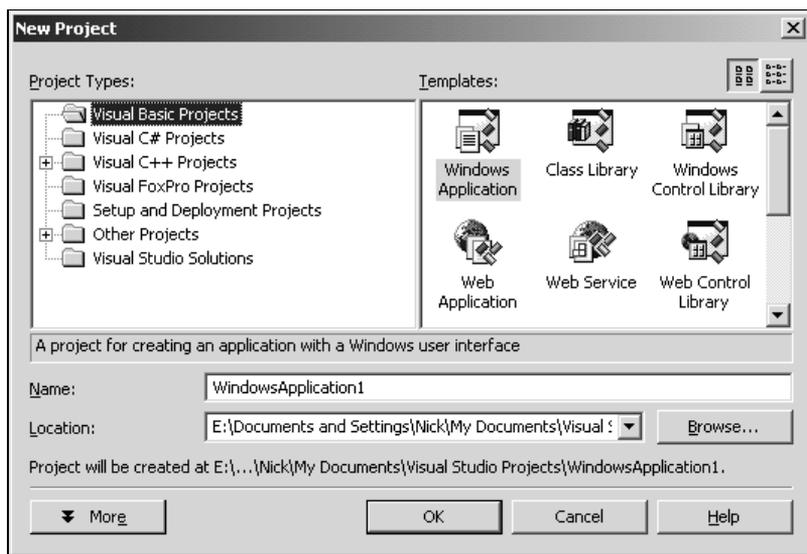


Рис. 1.1. Диалоговое окно **New Project**

- В окне иерархического списка **Project Types** (Типы проектов) раскройте папку **Visual C++ Projects** (Проекты Visual C++). Диалоговое окно **New Project** (Новый проект) примет вид, изображенный на рис. 1.2.

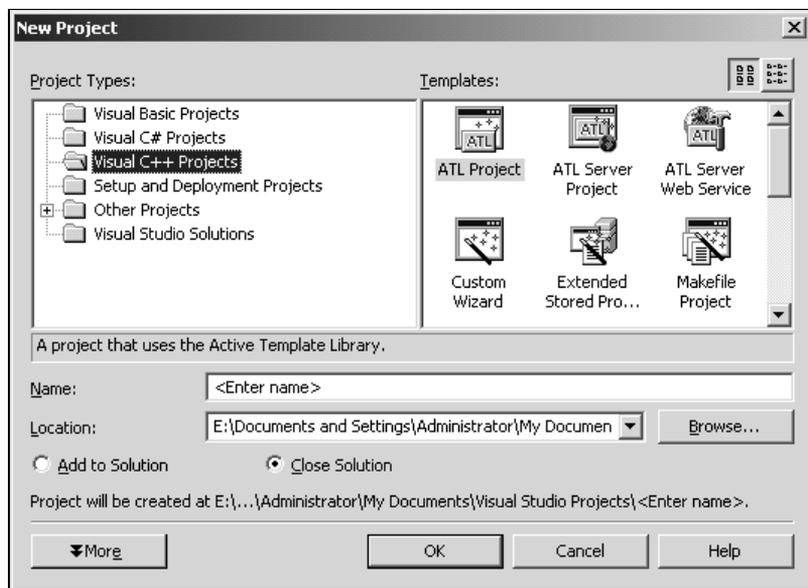


Рис. 1.2. Раскрытая папка **Visual C++ Projects**

- В окне списка **Templates** (Шаблоны) выделите значок **MFC Application** (Приложение MFC), в текстовое поле **Name** (Имя) введите имя приложения "Dialog" и нажмите кнопку **OK**. Появится диалоговое окно **MFC Application Wizard - Dialog** (Мастер создания приложений MFC), изображенное на рис. 1.3.
- Раскройте вкладку **Application Type** (Тип приложения). Диалоговое окно **MFC Application Wizard** (Мастер создания приложений MFC) примет вид, изображенный на рис. 1.4.
- Установите переключатель **Application type** (Выбор типа приложения) в положение **Dialog based** (Диалоговое) и нажмите кнопку **Finish** (Готово). Мастер MFC Application Wizard создаст заготовку диалогового приложения, как это показано на рис. 1.5.

В правой части окна Visual Studio.NET расположен ярлык вкладки **Toolbox** (Инструментарий), позволяющей автоматизировать процесс создания диалогового окна. Элементы управления диалогового окна и использование вкладки **Toolbox** (Инструментарий) будут более подробно описаны в главе 3, а попытка написания примера программы на базе этого приложения наталкивается на специфичность области его использования.



Рис. 1.3. Диалоговое окно **MFC Application Wizard - Dialog**

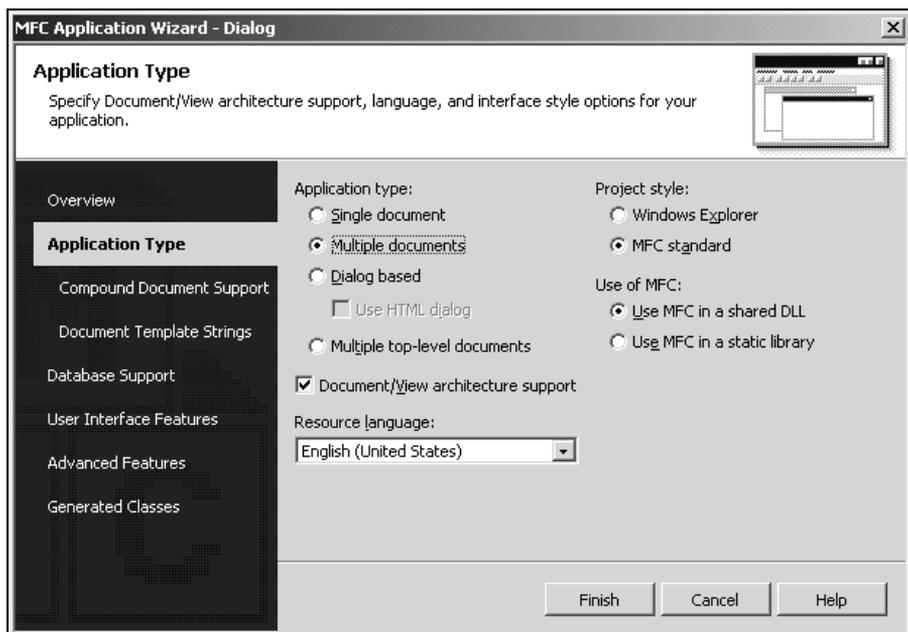


Рис. 1.4. Вкладка **Application Type**

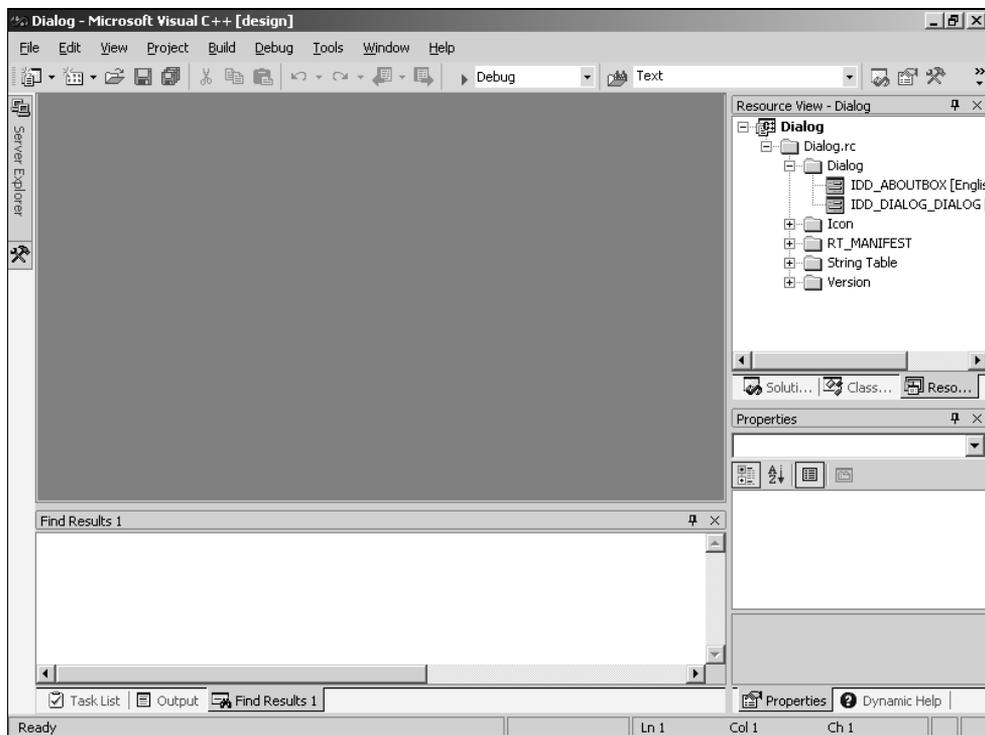


Рис. 1.5. Заготовка диалогового окна

Многооконное приложение Windows

Под многооконным приложением Windows мы будем понимать приложение, использующее MDI (Multiple Document Interface, Многооконный интерфейс приложения). Подобные приложения являются основным типом приложений в Windows, поэтому в дальнейшем мы будем в основном рассматривать приложения этого вида.

Демонстрационное многооконное приложение можно найти в папке MDI на прилагаемой к данной книге дискете.

Чтобы самостоятельно создать заготовку многооконного приложения Windows с использованием мастера MFC Application Wizard:

1. Закройте все открытые проекты и файлы и выберите команду **File | New | Project** (Файл | Новый | Проект) или нажмите кнопку **New Project** (Новый проект) на панели инструментов **Standard** (Стандартная). Появится диалоговое окно **New Project** (Новый проект).
2. В окне иерархического списка **Project Types** (Типы проектов) раскройте папку **Visual C++ Projects** (Проекты Visual C++).

3. В окне списка **Templates** (Шаблоны) выделите значок **MFC Application** (Приложение MFC), в текстовое поле **Name** (Имя) введите имя приложения "MDI" и нажмите кнопку **OK**. Появится диалоговое окно **MFC Application Wizard** (Мастер создания приложений MFC).
4. Раскройте вкладку **Application Type** (Тип приложения).
5. Оставив переключатель в положении **Multiple documents** (Многооконные приложения), нажмите кнопку **Finish** (Готово). Мастер MFC Application Wizard создаст заготовку многооконного приложения, но при этом не будет открыто ни одного окна редактирования файлов.
6. Запустите приложение на исполнение. Для чего выберите команду меню **Debug | Start** (Отладка | Запуск) или нажмите клавишу <F5>. Появится окно **Microsoft Development Environment** (рис. 1.6), сообщающее, что текущая конфигурация проекта устарела и предлагающее перекомпилировать проект.

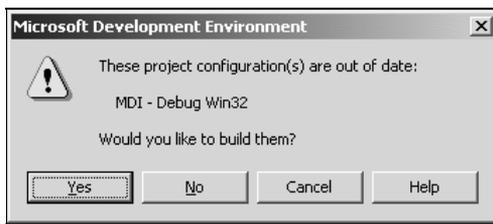


Рис. 1.6. Диалоговое окно **Microsoft Development Environment**

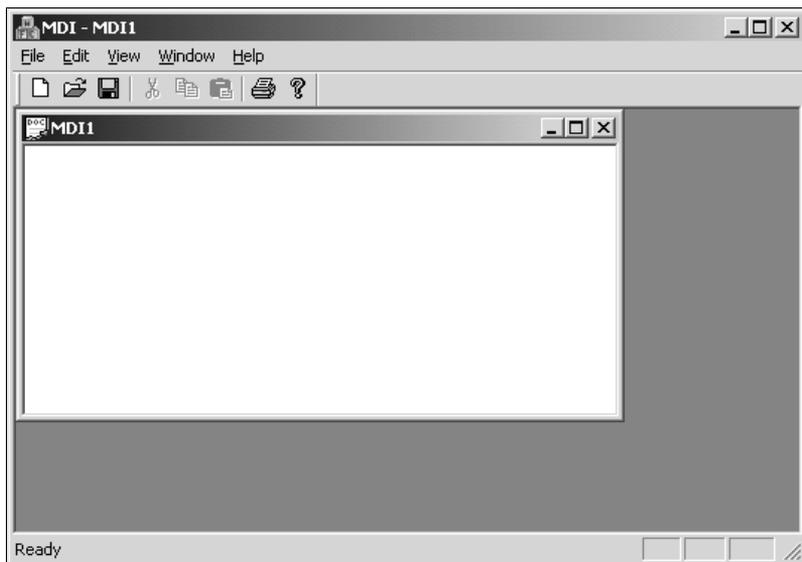


Рис. 1.7. Многооконное приложение

7. Нажмите кнопку **Yes** (Да). Visual C++ произведет трансляцию приложения и запустит его на исполнение. При этом на экране появится окно, изображенное на рис. 1.7.

Это приложение позволяет создавать новые окна, изменять их размер и уничтожать их. Оно может работать с меню и панелью инструментов, но не может выполнять никаких полезных действий. Структура этого проекта будет подробно рассмотрена в следующей главе.

Другие виды приложений MFC

На вкладке **Application Type** (Тип приложения) диалогового окна **MFC Application Wizard** (Мастер создания приложений MFC) остались нерассмотренными два положения переключателя: **Single document** (Однооконное приложение) и **Multiple top-level documents** (Несколько окон верхнего уровня).

Однооконное приложение является одним из базовых типов приложений Windows. Оно использует однооконный интерфейс приложения. Это приложение может работать только с одним документом и, по сути своей, занимает промежуточное положение между диалоговым приложением и многооконным приложением. Поскольку процесс его создания отличается от процесса создания многооконного приложения только установкой переключателя в соответствующее положение, то здесь будет приведен только внешний вид его окна (рис. 1.8).

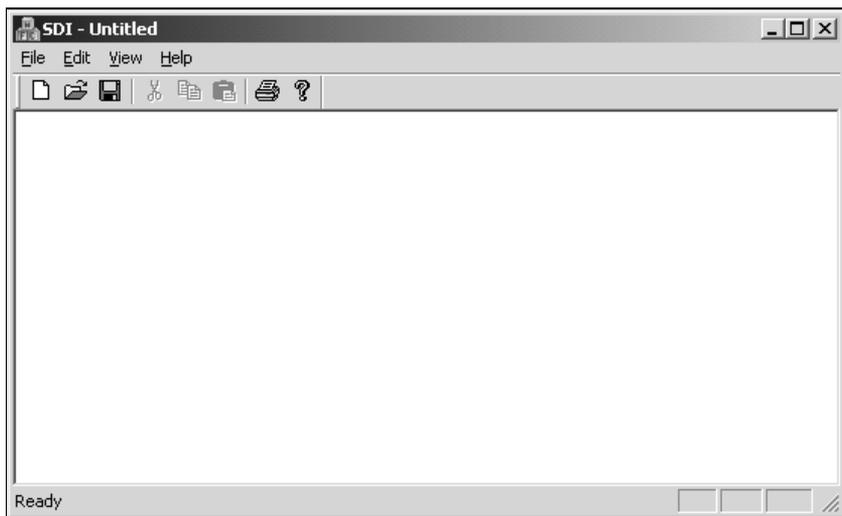


Рис. 1.8. Однооконное приложение

Данный тип приложения может использоваться для обработки конкретного типа документа, при этом предполагается, что не требуется использования других

документов. Подобные задачи, но без использования документов, решает диалоговое приложение.

Обычно обработка одного документа предполагает возможность одновременной работы со связанными с ним документами, например, для внесения в них соответствующих изменений по ходу работы или получения из них необходимой информации. Поэтому в Visual Studio.NET введен новый тип приложения: приложение с несколькими окнами верхнего уровня. Этот тип приложения, по сути, представляет собой несколько взаимосвязанных однооконных приложений. Первое из созданных окон является главным и его закрытие влечет за собой закрытие всех остальных окон. Остальные окна являются подчиненными и их закрытие не влечет за собой закрытие других окон. Окно, создаваемое из подчиненного окна, становится подчиненным окном главного окна и никак не зависит от окна, в котором было создано. При открытии файлов для них открываются свои подчиненные окна.

Единственным видимым отличием главного окна является то, что в его меню **File** (Файл) присутствует команда **Exit** (Выход). В то время как во всех остальных окнах она заменена командой **Close** (Заккрыть).

Консольное приложение

Помимо приложений MFC, особое внимание следует обратить на консольное приложение. Раньше использование этого типа приложений можно было рассматривать как попытку завоевать симпатии приверженцев программирования в среде DOS, оставшихся без средств программирования. По мере перехода от версии к версии возможности заготовок консольных приложений, создаваемых средой программирования Visual C++, постоянно росли. Так было по версию 6.0. С появлением пакета Visual Studio.NET, который иногда называют Visual Studio 7.0, все изменилось. Хотя в нем и остались средства создания консольных приложений, средства их создания не обеспечивают того удобства работы, которое было присуще версии 6.0.

Консольные приложения создаются, как правило, в тех случаях, когда для приложения требуется создать простейшую оболочку, обладающую самым простым интерфейсом. Информация консольному приложению передается в аргументах командной строки и через стандартный поток ввода. Для вывода информации консольное приложение использует стандартный поток вывода. Поэтому одной из основных областей использования консольных приложений является создание внепроцессных серверов, приложение которых должно иметь возможность работать с аргументами командной строки, что несколько затруднено при использовании других типов приложений.

С точки зрения программиста, консольное приложение является обычной программой DOS, использующей все стандартные операторы, но позволяющей, однако, использовать всю доступную оперативную память. Этот тип приложения запускается в особом типе окна, называемом "Окно MS DOS", в такое окно