

Введение

На протяжении почти 10 лет автор разрабатывал программное обеспечение, используя Visual C++ и библиотеку MFC. Весь этот период он внимательно наблюдал за его эволюцией. И с каждым годом увеличивалось количество новых программистов, желающих изучать и работать с флагманом *Microsoft* — средой разработки C++. Для этих будущих гуру MFC, а также для тех, кто уже имеет определенный опыт использования данной среды разработки, и была создана эта книга.

При написании книги, которая должна на многие годы стать основным руководством по среде Visual C++, трудней всего было гарантировать, что не будет пропущено что-либо на самом деле важное и не общеизвестное для большинства пользователей. Эта цель стала основной задачей проекта. Поэтому, чтобы быть в курсе насущных потребностей пользователей и последних новостей в данной области, автор постоянно участвовал в различных телеконференциях по Visual C++ и MFC, а также принимал участие в обсуждениях этих вопросов на страницах соответствующих Web-сайтов, что позволило лучше понять проблемы, с которыми чаще всего сталкиваются разработчики, и должным образом отобразить их в книге. В результате, книга содержит ответы на более чем 75% вопросов, регистрируемых в среднем за день на соответствующих Web-сайтах. Такой подход позволил автору организовать книгу, специально предназначенную для изучения MFC, таким образом, чтобы материал излагался в систематизированном, логически взаимосвязанном виде, одновременно обеспечивая ответы на вопросы, которые традиционно считаются наиболее трудными.

Для кого написана книга

Обратите внимание, данная книга посвящена прежде всего *библиотеке MFC* (Microsoft Foundation Classes — библиотека базовых классов), а вовсе не платформе .NET. В нескольких главах в конце книги описано создание *расширений управляемого кода* (managed extension) (расширений компилятора Visual C++, позволяющих преобразовывать код для среды выполнения .NET). Но эти главы не более чем вводный курс новой среды разработки, и если необходимо изучить ее более подробно, то имеет смысл раскошелиться на следующую книгу, посвященную именно разработке в среде .NET с использованием расширений управляемого кода.

Далее. Эта книга предназначена как для начинающих программистов, желающих узнать что такое библиотека MFC, так и для более опытных разработчиков, решивших расширить свои познания. Таким образом, здесь содержится информация полезная как для начинающих программистов, создающих свои первые приложения MFC, так и для более квалифицированных пользователей, нуждающихся в основном в советах на специфические темы и конкретных технологиях программирования. Как будет изложено в следующем разделе, книга организована так, чтобы можно было без затруднений перейти к разделам, соответствующим интересам и уровню квалификации читателя. Новичкам, вероятно, имеет смысл начинать с первой главы, изучение которой не предполагает наличия у читателя предварительных знаний о создании проектов и классов в Visual Studio. Каждая глава начинается вводным разделом, содержащим краткую информацию по обсуждаемой теме, затем следует пример довольно

простого демонстрационного приложения, а в конце главы содержится дополнительный материал и более подробное обсуждение темы. Например, в начале глав, посвященных DLL, изложены основы разработки библиотек DLL, а также сведения об импорте и экспорте классов. Затем, по мере изложения, сложность демонстрационных примеров нарастает, иллюстрируя способы размещения в библиотеке DLL необходимых документов и их представлений, способы создания *ловушек Windows* (Windows hook), а также способы организации совместного использования одного экземпляра объекта C++ вызовами из нескольких библиотек DLL при помощи нового малоизвестного оператора `placement new`.

В этой книге предполагается, что читатель обладает определенным опытом работы на языке C++. Несмотря на то, что основы MFC здесь изложены, обучить с нуля программированию на C++ с применением библиотеки MFC в одной книге практически невозможно. Следовательно, будем считать, что читателю известны такие базовые концепции языка C++, как конструкторы, списки инициализатора, виртуальные методы и т.д.

Кроме того, предполагается, что читатель имеет доступ к среде разработки Visual C++ .NET, поскольку все демонстрационные проекты данной книги были созданы именно для этой среды. Хотя в основе большей части кода примеров лежит библиотека MFC, и, в принципе, они должны компилироваться в среде разработки Visual C++ 6, но на практике этого никто не проверял.

Главы, посвященные базам данных, описывают ODBC, DAO, ADO и классы MFC для данных. Все главы снабжены демонстрационными приложениями, которые используют базу данных Microsoft Access. Microsoft Access (а не SQL Server) выбран специально, поскольку работать с ним намного проще, да и установлен он практически на каждой машине. Однако, для изучения этой книги вовсе не обязательно устанавливать на компьютере систему Microsoft Access, для правильного функционирования демонстрационных примеров достаточно самой базы данных.

Заключительные главы нашей книги представляют собой введение в мир платформы .NET и применения расширений управляемого кода. Кроме того, для изучения этих глав на компьютере следует установить комплект SDK .NET Framework.

Как организована книга

Чрезвычайно трудно написать достаточно большую книгу, которая подходила бы как для новичков, так и для более квалифицированных программистов. Поэтому организация книги играет весьма важную роль. Весь материал книги разбит на разделы, освещающие отдельные направления разработки, объединенные в логически упорядоченные категории. Новички, читая книгу шаг за шагом, получают систематизированные знания и не запутаются в огромном объеме информации. В тоже время квалифицированный программист легко найдет интересные разделы без необходимости пролистывать множество страниц, предназначенных для новичков.

Книга начинается с части I “Пользовательский интерфейс”. Как и все творческие личности, программисты народ импульсивный. Приобретая новую книгу, человек полон надежд, и никому не понравится, если его энтузиазм будет растоптан бесконечными главами теории и синтаксиса прежде, чем он сможет перейти к созданию какого либо кода! Именно поэтому первая часть начинается главой, позволяющей быстро ознакомиться со средой разработки Visual Studio и немедленно перейти непосредственно к созданию приложений Visual C++. Эта часть содержит несколько глав, позволяющих обучиться основам работы с пользовательским интерфейсом при разработке приложений MFC. Включая описание приложений SDI и MDI, концепции архитектуры документ/представление, работу с меню, клавиатурой и мышью, а также пару глав по введению в графическое программирование.

Рассмотрев способы создания пользовательского интерфейса, перейдем к части II, “Диалоги”, излагающей основы применения диалоговых окон в разрабатываемых приложениях. Вначале рассмотрим, как с помощью DDX организовать взаимодействие с элементами управления диалогового окна, как манипулировать переменными-членами, выводить на экран и управлять модальными диалоговыми окнами, осуществлять маршрутизацию команд и создавать наиболее часто используемые элементы управления. Следующие две главы посвящены более серьезным проблемам разработки диалоговых окон: работе с немодальными диалоговыми окнами, обмену сообщениями между диалоговыми окнами, применению акселераторов и перехвату ввода с клавиатуры, а также работе с формами представлений. Кроме того, здесь описано создание собственных классов, производных от класса `CCtrlView`, что позволяет сочетать возможности элементов управления с гибкостью архитектуры документ/представление. Заключительная глава этой части перекликается с главой, описывающей использование в приложениях классов вкладок и окон свойств, содержащих множество дополнительных возможностей: отключение вкладок, изменение шрифтов вкладок и использования мнемоники.

К моменту изучения части III, “Профессиональное программирование”, читатель уже будет иметь представление об основах программирования с использованием библиотеки MFC, поэтому можно приступить к более сложным темам. В данном разделе излагается, как можно расширить базовые функциональные возможности классов и элементов управления MFC. Часть начинается главой, описывающей создание собственных элементов управления, их кода и изображения (с помощью элементов графики), а также их применение в приложениях Windows. Далее изложен ряд способов (включая обычное наследование C++, множественное наследование, карты сообщений и шаблоны), позволяющих изменить стандартное поведение пользовательских элементов управления. В следующей главе описано применение печати и предварительного просмотра, использование библиотек DLL и многопоточное выполнение. Завершается часть подробным обсуждением обработки исключений, а также описанием их применения для повышения отказоустойчивости как простых систем, так и достаточно сложных.

Вполне очевидно, что от приложения не много пользы, если оно не обладает средствами хранения данных. Именно эту тему и раскрывает часть IV “Ввод и вывод данных”. В начале части описывается применение буфера обмена Windows. Большинство из нас повседневно пользуется буфером обмена, совершенно не задумываясь над тем, насколько просто применить его функциональные возможности в разрабатываемых приложениях, даже если необходимо использовать нестандартные форматы данных или объекты. Далее переходим к весьма серьезным главам, посвященным разработке баз данных с помощью ODBC и классов баз данных библиотеки MFC. Хотя читатель может сразу перейти к разделам, излагающим использование классов баз данных библиотеки MFC, прочитать имеет смысл все же обе главы, поскольку рассматриваемые классы инкапсулируют API ODBC, описываемые в первой из глав. Следовательно, чтобы лучше понять применение баз данных, автор рекомендует читать главы по порядку. В последней главе этой части обратимся к атрибутам. Атрибуты позволяют поднять применение компонентов (или объектов) на совершенно новый уровень. Поэтому, чтобы использовать их наилучшим образом, весьма рекомендуем прочесть эту главу.

Ни одна из книг по программированию на языке Visual C++ не могла бы претендовать на полноценность без раздела, посвященного COM и ATL. Нас бы просто не поняли! В части V “COM и ATL” приведен настолько полный обзор технологий разработки COM, что ему могли бы позавидовать некоторые специальные издания для новичков, посвященные исключительно COM. Вначале рассматриваются основы интерфейс-ориентированного программирования, потом — разработка и применение компонентов. Затем перейдем к более сложным концепциям COM: библиотекам типов и потоковым моделям. Изучив основы COM, можно приступить к использованию ATL (Active Template Library — библиотека активных шаблонов). Однако, в отличие от многих других книг, которые традиционно ограничиваются приложением

Hello World, примеры для этой разрабатывал известный программист в области ATL Эндрю Уайтчепел. Здесь охвачены такие темы, как работа с окнами, события и автоматизация. (Данная часть содержит небольшой приз — главу по WTL (Windows Template Library — библиотека шаблонов Windows).) Тот, кто полагает, что ATL упростила программирование COM, будут приятно удивлен, узнав, насколько применение WTL упростило программирование с использованием ATL!

Следуя логике постепенного усложнения, следующей темой данной книги будет “Применение COM+”. И хотя первая глава носит вводный характер, пропускать ее не стоит, поскольку в отличие от традиционных вводных глав она не только содержит описание COM+, но и достаточное количество примеров. Практически, в самом начале главы приведен код приложения COM+, демонстрирующего возможность разработки на C++ приложения с *фиксацией состояния* (stateful), которое можно затем использовать в клиентских сценариях. Представлена даже его демонстрационная версия, доступная по сети в качестве службы DCOM. Следующие главы иллюстрируют наиболее существенные аспекты разработки COM+: применение асинхронных событий и *компонентов очереди* (queued components).

Следующая часть, “Сервер ATL”, является естественным продолжением темы ATL и COM+. *ATL Server* (сервер ATL) — это набор расширений ATL, облегчающий разработчикам C++ создание Web-приложений, выполняющихся на сервере IIS. Как обычно, эта часть включает вводную главу, посвященную самой технологии, ее терминологии и способам применения. Затем описывается создание Web-служб, использующих ATL Server. В заключительной главе данной части приведено демонстрационное приложение Tip of the Day (Полезный совет), иллюстрирующее простоту создания реального Web-приложения при помощи C++ и ATL Server.

Завершает книгу часть VIII “Применение .NET”, в которой описывается достаточно новая область создания *управляемого кода* (managed code) в среде .NET. Изучение этой среды программирования начинается с введения в технологии .NET (включая составляющие их компоненты, их роль для перспектив разработки под Windows, а также их применение в настоящий момент). Затем излагаются основы разработки управляемого кода для платформы .NET при помощи Visual C++, включая создание многопоточковых приложений .NET, а также применение пакета форм Windows Forms.

Соглашения, принятые в этой книге

Подобно всем программистам, наравне с собственным стилем программирования, каждый автор имеет свои собственные любимые термины. Ниже приведен краткий перечень некоторых из них.

- *Предметная область* (problem domain) — автор узнал этот термин очень давно, еще при изучении методологии объектно-ориентированного анализа и проектирования. Предметная область — это обобщающий термин, который означает совокупность проблем, подлежащих решению.
- *Потребитель* (consumer) и *клиент* (client) — эти термины употребляются как синонимы, означая любой код, который использует определенный класс или тип.
- *Сервер* (server) — означает фрагмент кода (обычно класс), который используется клиентом или потребителем.
- *Аргумент* (argument) и *параметр* (parameter) — подобно большинству программистов, автор использует данные термины как синонимы для описания значений, передаваемых методу.

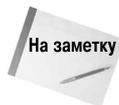
- *Прототип метода* (method prototype) и *сигнатура метода* (method signature) — здесь эти термины используются как синонимы.
- *Разрыв строк кода*. К сожалению, формат книги не всегда позволяет разместить на странице строку исходного кода полностью или перенести ее так, как того требуют правила переноса строк. Это прискорбно, но неизбежно. Пожалуйста, помните об этом, когда увидите странно оформленные строки кода.

Кроме того, в настоящей книге использованы следующие типографские соглашения.

- Новые термины в тексте выделяются *курсивом*, который также применяется с целью обратить внимание читателя на определенные участки текста.
- Выводимые на экран сообщения, примеры команд, имена переменных, любой код, присутствующий в тексте, а также унифицированные локаторы ресурсов (URL) Web-страниц представлены моноширинным шрифтом.
- Все, что придется вводить с клавиатуры, представлено **полужирным моноширинным** шрифтом.
- Знакомство в описаниях синтаксиса выделяется курсивом, что указывает на необходимость замены знакоместа фактическим именем переменной, параметром или другим элементом, который должен находиться на этом месте `BINDSIZE=(максимальная ширина колонки) * (номер колонки)`.
- Пункты меню и названия диалоговых окон представлены следующим образом: **Menu Option** (Пункт меню).

Если текст некоторых строк кода оказывается слишком длинным и не помещается на странице, то для разрыва строки применяется специальный символ ↵, указывающий, что данная строка кода является продолжением предшествующей.

В книге встречаются некоторые абзацы, текст которых выделен специальным шрифтом. Это примечания, советы и предостережения, которые, не являясь частью излагаемого текста, содержат важную информацию.



На заметку

Примечания содержат полезную и интересную информацию, которая позволяет более подробно изложить отдельные детали.



Совет

Советы содержат уникальную информацию, обеспечивающую наиболее эффективное решение поставленных задач.



Внимание!

Предостережения содержат информацию, которая поможет избежать возможных неприятностей.

Как связаться с автором

Для связи с автором лучше всего использовать его Web-сайт www.theCodeChannel.com, где кроме всего прочего размещены его книги и статьи.