Введение

Я помню время, много лет тому назад, когда я предложил издательству Apress книгу, посвященную еще не выпущенному на тот момент пакету инструментальных средств разработки под названием Next Generation Windows Services (NGWS — сервисы Windows следующего поколения). Вы, наверное, знаете, что пакет NGWS в конечном счете стал тем, что сегодня называется платформой .NET. Параллельно с моими исследованиями в области разработки языка программирования C# и платформы .NET шло создание начального варианта соответствующей рукописи. Это был фантастический проект, но я должен признаться, что писать о технологии, которая претерпевала весьма значительные изменения в ходе своей разработки, было занятием, очень действующим на нервы. К счастью, после многих бессонных ночей, где-то к началу лета 2001 года, первое издание книги *C# and the .NET Platform* было опубликовано почти одновременно с выходом .NET 1.0 Beta 2.

С того времени я был чрезвычайно рад и благодарен тому, что эта книга очень благосклонно принимается прессой и, самое главное, читателями. За эти годы книга предлагалась в качестве номинанта на премию Jolt Award (я тогда "пролетел"...) и на премию Referenceware Excellence Award 2003 года в категории книг по программированию (и я счастлив, что на этот раз мне повезло).

Второе издание этой книги (*C# and the .NET Platform, Second Edition*) дало мне возможность включить в нее материал, соответствующий версии 1.1 платформы .NET. Хотя второе издание книги содержало обсуждение множества новых тем, ряд глав и примеров включить в окончательный вариант книги все же не удалось.

Теперь, когда книга представлена в третьем издании, я с удовлетворением могу заявить, что она содержит (почти) все темы и примеры, которые я не смог представить в предыдущих изданиях. Это издание не только учитывает все многочисленные усовершенствования, предлагаемые в .NET 2.0, но включает и ряд глав, которые, будучи давно написанными, оставались до сих пор неопубликованными — это касается, например, описания CIL (Common Intermediate Language — общий промежуточный язык).

Как и в предыдущих изданиях, в этом, третьем издании книги на основе простого и понятного материала представлены язык программирования C# и библиотеки базовых классов .NET. Я никогда не понимал склонность некоторых авторов к созданию технических книг, более похожих на руководство по подготовке к экзамену GRE, чем пригодный для чтения текст. Задачей этого нового издания попрежнему остается предоставление вам информации, необходимой для построения программных решений сегодня, а не расточительная трата времени на рассмотрение множества деталей, которые оказываются важными для очень узкого круга специалистов.

Вы и я — одна команда

Публикации разработчиков новых технологий предназначены для очень требовательной аудитории (я должен знать это не понаслышке — ведь я один из них). Построение программных решений для любой платформы требует чрезвычайной детализации и учета множества особенностей соответствующей отрасли, компании, клиентской базы, а также учета сути дела. Вы можете работать в электронном издательстве, разрабатывать системы для федерального или местного правительства, работать в NASA или каком-то оборонном ведомстве. Я, например, разрабатывал программное обеспечение для обучения детей, создавал различные N-звенные системы и участвовал в многочисленных проектах для медицинских и финансовых учреждений. С вероятностью почти 100 процентов тот программный код, который вы создаете на своем рабочем месте, не имеет никакой связи с программным кодом, который пишу я (если, конечно, нам случайно не приходилось работать вместе).

Поэтому в этой книге я сознательно избегаю примеров, в которых программный код связан со спецификой определенных отраслей производства или сфер программирования. Я пытаюсь описать возможности C#, объектно-ориентированного подхода, CLR и библиотек базовых классов .NET 2.0 с помощью примеров, не использующих такой специфики. Вместо того чтобы в каждом примере заполнять таблицы реальными данными, рассчитывать платежки или выполнять какие-то другие специальные вычисления, я буду рассматривать объекты, с которыми могут иметь дело все, — например, автомобили (с их геометрическими формами и служащими соответствующего предприятия, добавленными для полноты картины). И тут на сцену должны выйти вы.

Моей целью является как можно более понятное объяснение возможностей языка программирования C# и описание различных аспектов его применения в рамках платформы .NET. Я сделаю все, что будет в моих силах, чтобы вы, используя знания и навыки, полученные в процессе работы над этой книгой, могли продолжить дальнейшее освоение соответствующих технологий.

Вашей целью является освоение этой информации и применение ее к вашим конкретным задачам программирования. Я, конечно, понимаю, что ваши проекты вряд ли напрямую связаны с автомобилями, имеющими имена домашних любимцев, но так уж заведено в прикладных науках! Уверен, если вы поймете концепции платформы .NET, представленные в этой книге, то сможете предложить и реализовать решения, подходящие для вашей конкретной среды программирования.

Обзор содержимого книги

Книга *Язык программирования C# 2005 и платформа .NET 2.0, 3-е издание* делится на пять логически обособленных разделов, каждый из которых состоит из глав, тем или иным образом связанных между собой. Если вы имели возможность ознакомиться с одним из предыдущих изданий этой книги, вы сможете заметить некоторое сходство в названиях ряда глав, но знайте, что здесь почти на каждую страницу был добавлен новый материал и дополнительные примеры. Вы можете также заметить, что некоторые темы, также освещенные в первом и втором изда-

ниях (например, сериализация объектов и сборщик мусора .NET), здесь представлены в виде отдельных глав.

Кроме того, и вы вправе это ожидать, третье издание книги содержит несколько глав с совершенно новым материалом (в частности главу, посвященную синтаксису и семантике CIL) и подробное описание специфических возможностей .NET 2.0. Теперь, после всех сделанных оговорок, мы перейдем к краткой характеристике содержимого книги по частям и главам.

Часть I. Общие сведения о языке C# и платформе .NET

Целью этой части книги является описание базовых принципов функционирования платформы .NET, системы типов .NET и различных инструментальных средств разработки, используемых при создании приложений .NET (многие из таких инструментальных средств являются программными продуктами с открытым исходным кодом). Здесь же представлены базовые возможности языка программирования C#.

Глава 1. Философия .NET

Материал этой главы является фундаментом для понимания всего остального материала книги. Глава начинается с обсуждения традиционных возможностей разработки программ в среде Windows и указывает на недостатки этого, использовавшегося ранее подхода. Но главной целью данной главы является знакомство с набором "строительных блоков" .NET, таких как CLR (Common Language Runtime — общеязыковая среда выполнения), CTS (Common Type System — общая система типов), CLS (Common Language Specification — общеязыковые спецификации) и библиотеки базовых классов. Здесь же предлагается вводная информация о языке программирования C# и формате компоновочных блоков .NET, рассматриваются межплатформенная независимость, вытекающая из самой природы платформы .NET, и роль CLI (Common Language Infrastructure — общеязыковая инфраструктура).

Глава 2. Технология создания приложений на языке C#

В этой главе предлагается краткое описание процесса компиляции и отладки файлов исходного кода для программ, написанных на языке C#, а также обсуждаются соответствующие инструменты и технологии. Сначала вы узнаете, как использовать компилятор командной строки (`csc.exe`) и файлы ответных сообщений C#. После этого будут рассмотрены некоторые из множества пакетов, предлагающих интегрированную среду разработки, — это, в частности, TextPad, SharpDevelop, Visual C# 2005 Express и (конечно же) Visual Studio 2005. Тут же будет представлен ряд аналогичных пакетов с открытым исходным кодом (Vil, NAnt, NDoc и т.д.), которые на всякий случай должен иметь любой разработчик .NET.

Часть II. Язык программирования C#

В этой части исследуются основные возможности языка программирования C#, включая новые синтаксические конструкции, появившиеся с выходом .NET 2.0. Кроме того, часть II познакомит вас с элементами CTS (классы, интерфейсы, структуры, перечни и делегаты) и конструкциями общих типов.

Глава 3. Основы языка C#

В этой главе рассматриваются базовые конструкции языка программирования C#. Вы освоите технику построения классов, выясните разницу между типами, характеризуемыми значениями, и ссылочными типами, приведением к объектному типу и восстановлением из "объектного образа", а также роль любимого всеми базового класса `System.Object`. В этой же главе показано, как платформа .NET заставляет работать самые простые программные конструкции, такие как перечни, массивы и обработчики строк. Наконец, в этой главе рассматривается ряд специфических для версии 2.0 вопросов, включая типы данных, для которых имеется разрешение принимать значение `null`.

Глава 4. Язык C# 2.0 и объектно-ориентированный подход

Целью главы 4 является выяснение того, как язык C# сочетается с базовыми принципами ООП — инкапсуляцией, наследованием и полиморфизмом. После рассмотрения ключевых слов и синтаксиса, используемых при построении иерархии классов, будет выяснена роль XML-комментариев в программном коде.

Глава 5. Цикл существования объектов

В этой главе выясняется, как в рамках CLR с помощью сборщика мусора .NET организовано управление памятью. В этой связи вы узнаете о роли корней приложения, генераций объектов и типа `System.GC`. После изучения базовых вопросов в оставшейся части главы будут рассмотрены тема объектов, предусматривающих освобождение ресурсов (через интерфейс `IDisposable`), и процесс финализации (реализуемый с помощью метода `System.Object.Finalize()`).

Глава 6. Структурированная обработка исключений

В этой главе обсуждается структурированный подход к обработке исключений (т.е. исключительных ситуаций), которые могут возникать в среде выполнения. Вы узнаете о ключевых словах языка C# (`try`, `catch`, `throw` и `finally`), которые позволяют решать соответствующие проблемы, а также выясните разницу между исключениями системного уровня и уровня приложения. Дополнительно в главе обсуждаются специальные средства Visual Studio 2005, призванные упростить задачу выявления и обработки исключений, ускользнувших от вашего внимания.

Глава 7. Интерфейсы и коллекции

Материал этой главы опирается на понимание принципов разработки приложений с использованием объектов и охватывает вопросы программирования на базе интерфейсов. Вы узнаете, как определить типы, поддерживающие множество элементов поведения, как найти эти элементы поведения во время выполнения и как выборочно скрыть такие элементы поведения, используя *явную реализацию интерфейса*. В оставшейся части главы рассматривается пространство имен `System.Collections`, с помощью которого демонстрируется польза типов интерфейса.

Глава 8. Интерфейсы обратного вызова, делегаты и события

Цель главы 8 заключается в разъяснении типа делегата. Упрощенно говоря, *делегат* .NET — это шаблон, "указывающий" на методы в приложении. С помощью

такого шаблона вы получаете возможность строить системы, в которых множество объектов могут быть связаны двусторонним обменом. После выяснения возможностей использования делегатов .NET (включая множество возможностей, появившихся с версией 2.0, — например, анонимные методы) в главе рассматривается ключевое слово C# `event`, которое используется для того, чтобы упростить процесс программирования с помощью делегатов.

Глава 9. Специальные приемы построения типов

Эта глава позволит вам глубже понять возможности языка программирования C# путем изучения более совершенных методов программирования. Вы узнаете, как использовать перегрузку операций и создавать пользовательские программы преобразования (явного или неявного), как строить индексаторы типов и работать с указателями C-типа в файле `*.cs`.

Глава 10. Обобщения

В связи с разработкой .NET 2.0 язык программирования C# был расширен с целью поддержки новой возможности CTS, связанной с так называемыми *обобщениями* (generics). Вы увидите, что программирование с помощью обобщений ускоряет процесс создания приложений и обеспечивает типовую безопасность. Здесь рассмотрены различные обобщенные типы из пространства имен `System.Collections.Generic`, а также показано, как строить свои собственные обобщенные методы и типы (как с ограничениями, так и без таковых).

Часть III. Программирование компоновочных блоков .NET

В этой части рассматривается формат компоновочных блоков .NET. Вы узнаете, как развернуть и сконфигурировать библиотеки программного кода .NET, и выясните внутреннюю структуру бинарного образа .NET. В этой же части объясняется роль атрибутов .NET и конструкция многопоточных приложений. В последних главах этой части рассматриваются низкоуровневые вопросы (такие как, например, объектный контекст), а также синтаксис и семантика CIL.

Глава 11. Компоновочные блоки .NET

С точки зрения высокоуровневого подхода, *компоновочные блоки* — это файлы `*.dll` или `*.exe`. Но такая интерпретация компоновочных блоков .NET очень далека от "исчерпывающей". Вы узнаете, чем отличаются одномодульные и многомодульные компоновочные блоки и как строятся и инсталлируются такие объекты. Вы научитесь конфигурировать приватные и общедоступные компоновочные блоки, используя для этого XML-файлы `*.config` и компоновочные блоки политики поставщика. По ходу дела будут выяснены внутренняя структура GAC (Global Assembly Cache — глобальный кэш компоновочных блоков) и роль утилиты конфигурации .NET Framework 2.0.

Глава 12. Отображение типов, динамическое связывание и программирование с помощью атрибутов

В главе 12 изучение компоновочных блоков .NET продолжается — здесь рассматривается процесс обнаружения типов в среде выполнения с помощью простран-

ства имен System.Reflection. С помощью этих типов можно строить приложения, способные читать метаданные компоновочных блоков "на лету". Вы узнаете, как динамически активизировать и обрабатывать типы во время выполнения программы, используя *динамическое связывание*. Здесь же исследуется роль атрибутов .NET (как стандартных, так и создаваемых программистом). Чтобы иллюстрировать возможности применения обсуждавшихся подходов, в конце главы рассматривается конструкция расширяемого приложения Windows Forms.

Глава 13. Процессы, домены приложений, контексты и хосты CLR

Здесь выполняется более глубокий анализ структуры загруженного выполняемого файла .NET. Главная цель — иллюстрация взаимосвязи между процессами, доменами приложений и границами контекстов. Определив эти объекты, вы сможете понять, как обслуживается CLR в рамках операционной системы Windows, и расширить свои знания о mscoree.dll. Представленная здесь информация может оказаться очень полезной при освоении материала главы 14.

Глава 14. Создание многопоточных приложений

В этой главе объясняется, как строить многопоточные приложения, и иллюстрируется ряд приемов, которые вы можете использовать для создания программного кода, безопасного с точки зрения многопоточных приложений. В начале главы снова обсуждается тип делегата .NET, и это делается с целью выяснения внутренних механизмов делегата, используемых для поддержки асинхронного вызова методов. Затем исследуются типы пространства имен System.Threading. Здесь обсуждается множество типов (Thread, ThreadStart, и т.п.), позволяющих очень просто создавать дополнительные потоки.

Глава 15. Понимание CIL и роль динамических компоновочных блоков

В этой главе ставится две цели. В первой половине главы рассматриваются синтаксис и семантика CIL, намного более подробно, чем в предыдущих главах. Остаток главы посвящен выяснению роли пространства имен System.Reflection.Emit. С помощью соответствующих типов можно строить программное обеспечение, позволяющее генерировать компоновочные блоки .NET в памяти во время выполнения программы. Компоновочные блоки, определенные и выполняемые в памяти, формально называют *динамическими компоновочными блоками*.

Часть IV. Программирование с помощью библиотек .NET

К этому моменту вы уже имеете достаточно информации о языке C# и формате компоновочных блоков .NET. Часть IV предлагает расширить ваши новые знания и исследовать целый ряд пространств имен в рамках библиотек базовых классов, в частности файловый ввод-вывод, слой удаленного доступа .NET, конструкцию Windows Forms и доступ к базам данных с помощью ADO.NET.

Глава 16. Пространство имен System.IO

По названию указанного пространства имен можно догадаться, что System.IO обеспечивает взаимодействие со структурой файлов и каталогов соответствующей

машины. Из этой главы вы узнаете, как программными средствами можно создать (или уничтожить) систему каталогов, как размещать данные внутри различных потоков (файловых, строковых, в памяти и т.д.) и как выводить их оттуда.

Глава 17. Сериализация объектов

В этой главе рассматриваются сервисы сериализации объектов для платформы .NET. Упрощенно говоря, сериализация позволяет "консервировать" состояние объекта (или множества связанных объектов) в потоке для использования в будущем. Десериализация (как вы можете догадаться сами) является процессом извлечения объекта из потока для восстановления в памяти с целью использования этого объекта в приложении. Поняв базовые принципы этих процессов, вы сможете управлять процессами сериализации с помощью интерфейса `ISerializable` и множества новых атрибутов, предлагаемых .NET 2.0.

Глава 18. Удаленное взаимодействие .NET

Вопреки распространенному убеждению, Web-сервисы XML не являются единственным средством построения распределенных приложений для платформы .NET. Из этой главы вы узнаете о слое удаленного доступа .NET. Вы увидите, что CLR поддерживает простые возможности обмена объектами между приложениями из разных доменов и на разных машинах, используя семантику MBV (marshal-by-value — маршалинг по значению) и MBR (marshal-by-reference — маршалинг по ссылке). По ходу дела вы узнаете, как в декларативной форме во время выполнения можно изменить поведение распределенного .NET-приложения, используя XML-файлы конфигурации.

Глава 19. Создание окон с помощью System.Windows.Forms

В этой главе начинается ваше знакомство с пространством имен `System.Windows.Forms`. Подробно обсуждается вопрос построения традиционных приложений с графическим интерфейсом, поддерживающим системы меню, панели инструментов и строки состояния. Как и следует ожидать, здесь рассматриваются различные аспекты проектирования форм в Visual Studio 2005, а для .NET 2.0 — целый ряд типов Windows Forms (`MenuStrip`, `ToolStrip` и т.п.).

Глава 20. Визуализация графических данных средствами GDI+

В этой главе говорится о том, как реализовать динамическую визуализацию графических данных в приложении Windows Forms. Кроме обсуждения вопросов обработки шрифтов, цветовых данных, геометрических образов и файлов изображений, в этой главе рассматриваются вопросы проверки попадания в заданную область и техника перетаскивания объектов в рамках графического интерфейса. Вы узнаете о новом формате ресурсов .NET, который, как вы уже можете догадываться к этому моменту, основан на XML-представлении данных.

Глава 21. Использование элементов управления Windows Forms

Эта глава является последней из глав книги, связанных с обсуждением приложений для Windows, и здесь будет рассмотрено множество элементов графического интерфейса, предлагаемых в .NET Framework 2.0. Вы научитесь использовать раз-

личные элементы управления Windows Forms, узнаете о приемах разработки диалоговых окон и наследовании форм. В этой же главе рассматривается возможность построения пользовательских элементов управления Windows Forms, которые можно интегрировать в IDE (Integrated Development Environment — интегрированная среда разработки).

Глава 22. Доступ к базам данных с помощью ADO.NET

ADO.NET — это API (Application Programming Interface — интерфейс программирования приложений) доступа к данным для платформы .NET. Вы увидите, что с типами ADO.NET можно взаимодействовать как на связном уровне ADO.NET, так и несвязном. В этой главе будут рассмотрены оба эти режима ADO.NET, а также некоторые новые возможности, связанные с .NET 2.0, включая модель источника данных, построители строк соединений и асинхронный доступ к базам данных.

Часть V. Web-приложения и Web-сервисы XML

Эта часть книги посвящена созданию Web-приложений ASP.NET и Web-сервисов XML. Из материала первых двух глав этой части вы узнаете, что ASP.NET 2.0 является значительным шагом вперед по сравнению с ASP.NET 1.x и предлагает множество новых возможностей.

Глава 23. Web-страницы и Web-элементы управления ASP.NET 2.0

В этой главе начинается изучение Web-технологий, поддерживаемых в рамках платформы .NET с помощью ASP.NET. Вы увидите, что программный код сценариев серверной стороны теперь заменяется “реальными” объектно-ориентированными языками (такими как C#, VB .NET и им подобными). Здесь будут рассмотрены ключевые для ASP.NET вопросы, такие как работа с файлами, содержащими внешний программный код поддержки, роль Web-элементов управления ASP.NET, использование элементов управления, связанных с контролем ввода, и взаимодействие с новой моделью “шаблона страницы”, предлагаемой ASP.NET 2.0.

Глава 24. Web-приложения ASP.NET 2.0

Эта глава расширяет ваши знания о возможностях ASP.NET с помощью рассмотрения различных способов управления состоянием объектов в рамках .NET. Подобно классической модели ASP, приложение ASP.NET позволяет создавать файлы cookie, а также переменные уровня приложения или сеанса. Однако ASP.NET предлагает и новую технологию управления состояниями — это кэш приложения. Рассмотрев многочисленные способы обработки состояний в ASP.NET, вы сможете выяснить роль базового класса `System.HttpApplication` (скрытого в файле `Global.asax`) и научиться динамически менять поведение Web-приложения в среде выполнения, используя файл `Web.config`.

Глава 25. Web-сервисы XML

В этой последней главе книги выясняется роль Web-сервисов XML и рассматриваются возможности их создания в рамках .NET. Грубо говоря, *Web-сервис* — это компоновочный блок, активизируемый с помощью стандартных HTTP-запросов.

Преимущество этого подхода заключается в том, что HTTP является сетевым протоколом, применяемым почти повсеместно, поэтому он прекрасно подходит для использования в распределенных системах, нейтральных в отношении различных платформ и языков. Здесь же вы узнаете о множестве сопутствующих технологий (WSDL, SOAP и UDDI), которые обеспечивают гармонию взаимодействия Web-сервиса и внешнего клиента.

Исходный код примеров книги

Программный код всех примеров из этой книги (с точностью до встречающихся кое-где небольших фрагментов) доступен для загрузки из раздела исходного кода Web-узла издательства. Выполнив поиск по названию книги, перейдите на ее "домашнюю" страницу, откуда вы сможете загрузить файл `*.zip` с исходным кодом примеров. После распаковки содержимого этого файла вы обнаружите соответствующий программный код, разделенный по главам.

Обратите внимание на то, что в книге в разделах с названием *Исходный код* указано, где содержится программный код обсуждаемого примера. Этот программный код можно загрузить в Visual Studio 2005 для проверки и модификации.

Исходный код. В таком примечании указывается ссылка на каталог, содержащий исходный код соответствующего примера.

Для загрузки примера откройте файл `*.sln` в указанном подкаталоге.

Связь с автором

Если у вас возникнут вопросы в связи с исходным кодом примеров, потребность в дополнительных разъяснениях или просто желание поделиться своими идеями в отношении платформы .NET, без всякого стеснения пишите мне на мой адрес электронной почты `atroelsen@IntertechTraining.com` (чтобы гарантировать, что ваше сообщение не окажется в корзине моей почтовой системы, укажите "C# TE" в строке темы).

Я постараюсь сделать все возможное, чтобы ответить в приемлемые сроки, но прошу учесть то, что я, как и вы, время от времени бываю очень занят. Если я не отвечаю в течение недели или двух, то знайте, что это не из вредности и не потому, что я не хочу общаться с вами. Я просто занят (или, если выпало счастье, нахожусь где-то на отдыхе).

Так что, вперед! Спасибо за то, что вы купили эту книгу (или, как минимум, заглянули в нее в книжном магазине, обдумывая возможность ее покупки). Я надеюсь, что вам будет приятно читать ее, и вы сможете применить полученные знания в благородных целях.

Берегите себя,
Эндрю Троелсен