

Введение

Язык программирования С# является мощным, относительно новым потомком более ранних языков С, С++ и Java. Программирование на нем доставляет много удовольствия, в чем можно будет убедиться при работе с этой книгой.

Язык С# был создан Microsoft как главная часть ее инициативы .NET. Возможно, из соображений политики компания Microsoft направила спецификации языка С# в комитет международных стандартов ассоциации ECMA (произносится как “эк-ма”) летом 2000 года, задолго до внедрения платформы .NET. Теоретически любая компания может разработать свою собственную версию языка С#, предназначенную для работы в любой операционной системе и на любой машине, большей, чем калькулятор.

Когда вышло первое издание настоящей книги, компилятор языка С# Microsoft был единственным, и ее инструментальный пакет Visual Studio .NET предлагал единственную возможность программирования на языке С#. С тех пор, однако, Visual Studio претерпел два существенных изменения — появилась версия Visual Studio 2003 и, совсем недавно, Visual Studio 2005. И, по крайней мере, еще два игрока вступили в игру С#.

В настоящее время можно писать и компилировать программы на языке С# на множестве машин, работающих под управлением Unix, при помощи реализаций Mono или Portable .NET платформы .NET и языка С#.

- ✓ Mono (www.go-mono.com) является программным проектом с открытым исходным кодом, финансируемым компанией Novell Corporation. Версия 1.1.8 вышла в июне 2005 года. Хотя проект Mono и отстает от платформы .NET компании Microsoft (версия 1.1 Microsoft выпустила пару лет назад), он быстро развивается.
- ✓ Проект Portable .NET фирм Southern Storm Software и DotGNU (www.dotgnu.org/pnet.html) также является проектом с открытым исходным кодом. Во время написания этой книги текущей версией проекта Portable .NET была 0.7.0.

Оба проекта предназначены для выполнения программ С# в Windows и различных операционных системах семейства Unix, включая Linux и Macintosh компании Apple. Когда писалась эта книга, проект Portable .NET работал на большем количестве платформ, в то время как проект Mono гордится более полной реализацией платформы .NET. Так что выбор между ними может быть затруднен, в зависимости от вашего проекта, платформы и целей. (Книги по программированию для этих платформ уже становятся доступны. Посетите сайт www.amazon.com.)



Программное обеспечение с открытым исходным кодом создается сотрудничающими группами программистов-добровольцев и обычно является бесплатным для всех.

Переносимость языка С# и других языков платформы .NET выходит далеко за рамки настоящей книги. Но можно ожидать, что в течение нескольких лет программы С# для Windows, которые можно научиться создавать по этой книге, будут работать на различном аппаратном обеспечении и для всех типов операционных систем, что соответствует требованиям для языка Java компании Sun Microsystems — о работе на любой машине. Это, несомненно, хорошая вещь, даже для Microsoft. Переносимость — вопрос, над ко-

торым в настоящее время идет интенсивная работа, так что нет никаких сомнений, что все трудности и препятствия на пути к истинной универсальной переносимости языка C# будут преодолены. Но этот путь является уже не только путем Microsoft.

Однако в настоящий момент пакет Visual Studio компании Microsoft содержит наиболее развитые версии языка C# и платформы .NET, а также набор инструментов к ним с богатыми возможностями для программирования.



Если вам нужен только C#, то в одной из дополнительных глав вы узнаете, как практически бесплатно написать код C#. (Вы потеряете множество удобств, включая отличные средства визуального дизайна, обеспечиваемые Visual Studio 2005, но сможете создавать код Windows и без них — в особенности такой простой код, как рассматриваемый в этой книге.)

Что нового в C# 2.0

Хотя в версию 2.0 языка C# был внесен ряд изменений, в основном C# остается практически таким же, как и в предыдущей версии. В этой книге рассматриваются следующие значительные нововведения.

- ✓ **Блоки итераторов:** *итератор* представляет собой объект, который позволяет пройти по всем элементам *набора*, или *коллекции* объектов. Сделать это можно было всегда, но C# 2.0 значительно упрощает использование итераторов. Коллекции рассматриваются в главе 15, “Обобщенное программирование”.
- ✓ **Обобщенное программирование** является важным нововведением! Новые возможности позволяют создавать более обобщенный и гибкий код, что является мечтой любого программиста. Из главы 15, “Обобщенное программирование”, вы узнаете, как создавать более простой код с улучшенной *безопасностью типов* с помощью обобщенного программирования.

Оставляя в стороне более сложные нововведения, следует упомянуть о нескольких более простых элементах в соответствующих разделах книги. (Не нужно беспокоиться, если что-то в этом *Введении* вам непонятно. Все станет ясно в процессе чтения соответствующих глав книги.)

Об этой книге

Цель книги заключается в объяснении языка программирования C#, но для реального написания программ необходима специальная среда кодирования. Мы уверены, что большинство читателей будет использовать Microsoft Visual Studio, хотя предусмотрены и другие варианты. Основывая книгу на Visual Studio, мы попытались сделать долю Visual Studio минимально необходимой. Можно было бы просто сказать: “Запускайте свою программу каким угодно образом”, но вместо этого мы говорим: “Запускайте свою программу C# в Visual Studio нажатием клавиши <F5>”. Мы хотим, чтобы вы могли сосредоточиться на самом языке C#, а не на том, как работают простые вещи.

Вполне понятно, что многие читатели, если не большинство, захотят использовать C# для создания графических приложений Windows, поскольку язык C# является мощным средством разработки подобных программ, но это всего лишь одна из областей

применения C#. Данная же книга должна в первую очередь обращать внимание на C#, как на язык программирования. Графические программы Windows будут кратко рассмотрены в первой главе, но вы должны хорошо понять основы C#, прежде чем переходить к программированию для Windows. Также вполне понятно, что некоторые опытные пользователи будут применять C# для создания сетевых, распределенных приложений. Однако из-за издательских ограничений невозможно включить эту тему в данную книгу. В книге *C# для чайников* распределенное программирование не рассматривается. В ней совсем кратко рассмотрена платформа .NET — по той простой причине, что могущество языка C# во многом исходит из библиотек классов .NET Framework, которые используются этим языком.

Что необходимо для чтения книги

Для того чтобы просто запустить программы, сгенерированные C#, нужна, как минимум, *общезыковая исполняющая среда* (Common Language Runtime — CLR). Visual Studio 2005 копирует систему CLR на вашу машину во время процедуры установки. В качестве альтернативы можно загрузить весь пакет .NET, включая компилятор языка C# и множество других полезных инструментов, зайдя на Web-сайт компании Microsoft по адресу <http://msdn.microsoft.com>. Ищите там *набор инструментальных средств разработки программного обеспечения .NET* (Software Development Toolkit — SDK).



Большинство программ, приведенных в этой книге, можно при необходимости создавать и в среде Visual Studio 2003. Исключениями являются программы, содержащие новые возможности, доступные только в языке C# 2.0, прежде всего обобщения и блоки итераторов. Имеется также более дешевая версия системы Visual Studio 2005 — C# Express 2005, и другие недорогие альтернативы, рассматриваемые в дополнительных главах.

Как использовать книгу

При создании настоящей книги авторами преследовалась цель сделать ее максимально легкой в использовании, поскольку изучение нового языка и так достаточно трудное. Зачем же излишне его усложнять? Книга разделена на части. В первой части представлено введение в программирование на C# с использованием Visual Studio. В ней пошагово излагается создание двух различных типов программ. Авторы настоятельно рекомендуют начать с этой части и прочесть данные две главы, прежде чем перейти к другим частям книги. Даже если вы программировали ранее, базовая структура программы, созданная в первой части, постоянно применяется во всей книге.

Главы в частях со второй по пятую являются самостоятельными. Они написаны так, чтобы можно было открыть книгу на любой из них и начать чтение. Если вы новичок в программировании, то должны полностью прочесть вторую часть, прежде чем идти далее. Но если просто возвращаетесь назад, чтобы освежить свою память по некоторой определенной теме, у вас не возникнет никаких проблем при переходе к разделу, и вам не нужно будет повторно перечитывать предыдущие 20 страниц.

И, конечно же, книгу завершают традиционная часть о “великолепных десятках” и дополнительные главы; много интересного можно найти и на компакт-диске, прилагаемом к книге.

Структура книги

Ниже приведено краткое описание каждой части книги.

Часть I, “Создание ваших первых программ на C#”

В этой части шаг за шагом рассматривается написание минимального графического приложения Windows с использованием интерфейса Visual Studio 2005. В ней также показывается, как создать базовую структуру консольной программы C#, которая используется в других частях книги.

Часть II, “Основы программирования в C#”

На базовом уровне пьесы Шекспира — это всего лишь набор слов, связанных вместе. С этой же точки зрения 90% любой программы C#, которую вы когда-либо напишете, состоит из создания переменных, выполнения арифметических действий и управления ходом выполнения программы. Во второй части внимание уделяется этим основным операциям.

Часть III, “Объектно-основанное программирование”

Одно дело — объявлять переменные где-либо в программе, добавлять и убирать их. И совсем другое — создавать реальные программы для реальных людей. В третьей части внимание уделяется тому, как организовывать данные так, чтобы их было легче использовать при создании программы.

Часть IV, “Объектно-ориентированное программирование”

Вы можете соединять части самолета так, как пожелаете, но пока вы не сложите их правильно, у вас нет ничего, кроме кучи деталей. И только когда вы соберете самолет так, что сможете запустить двигатели и использовать подъемную силу крыла — только тогда вы сможете лететь куда угодно.

В четвертой части объясняется, как превратить набор данных в реальный объект — объект, который имеет внутренние члены и может моделировать свойства реальных вещей. В этой части представлена сущность объектно-ориентированного программирования.

Часть V, “За базовыми классами”

После того как самолет оторвется от земли, он должен куда-нибудь лететь. Изучение классов и основ объектно-ориентированного программирования — это только начало. В данной части сделан следующий шаг: в ней представлены структуры, интерфейсы и обобщения, открывающие доступ к более мощным объектно-ориентированным концепциям.

Часть VI, “Великолепные десятки”

Язык C# силен в поиске ошибок в ваших программах — иногда кажется, что он даже слишком хорошо указывает на недостатки. Однако верите вы в это или нет, но C# все же пытается принести вам пользу. Каждая проблема, им обнаруженная, могла бы привести к другим проблемам, которые вам пришлось бы находить и локализовывать самостоятельно.

К сожалению, сообщения об ошибках могут сбивать с толку. В одной из глав этой части представлено десять наиболее общих сообщений об ошибках C# времени компиляции, их значение, и как от них избавиться.

Многие читатели переходят в C# из других языков программирования. Во второй главе этой части описаны десять основных отличий между C# и его предком C++.

О прилагаемом CD-ROM

На прилагаемом компакт-диске содержится масса хороших вещей. Прежде всего на нем можно найти все исходные тексты из этой книги. Кроме того, на компакт-диске содержится набор полезных утилит. Утилита SharpDevelop не рекомендуется для полномасштабной разработки коммерческих программ, но она весьма полезна для написания небольших приложений или быстрого внесения изменений, чтобы не ждать, пока загрузится Visual Studio. Она полностью подходит для компиляции всех исходных текстов данной книги. Редактор TextPad представляет собой существенно усиленную версию стандартного Блокнота. Он предоставляет прекрасную дешевую платформу для программирования на C#. Инструмент тестирования NUnit, очень популярный среди программистов на C#, проводит проверку вашего кода легче, чем из Visual Studio, SharpDevelop или TextPad. Не пренебрегайте компакт-диском и имеющимися на нем программами.

И, конечно, не забудьте о файле README, содержащем всю наиболее свежую информацию.

Пиктограммы, используемые в книге

В книге используются следующие пиктограммы для выделения важной информации.



Этой пиктограммой помечен технический материал, который можно пропустить при первом чтении.



Данной пиктограммой выделены места, которые могут сохранить много вашего времени и усилий.



Это необходимо запомнить, так как это важно.



Это также следует запомнить. Иначе оно настигнет вас тогда, когда вы меньше всего ожидаете, и создаст одну из действительно трудно находимых ошибок.



Данная пиктограмма указывает код, который можно найти на прилагаемом к этой книге компакт-диске. Эта возможность предназначена, чтобы избавить вас от лишнего набора, если ваши пальцы начали дрожать. Но не злоупотребляйте ею — вы лучше поймете C#, если будете набирать текст программ самостоятельно.

Соглашения, используемые в книге

Чтобы помочь вам, в книге используется несколько соглашений. Термины, которые не являются “настоящими словами”, такие как имена переменных, напечатаны таким шрифтом. Листинги программ выделены из текста следующим образом:

```
use System;
namespace MyNameSpace
{
    public class MyClass
    {
    }
}
```

Каждый листинг сопровождается ясным и понятным пояснением. Полные исходные тексты программ помещены на прилагаемый компакт-диск, в отличие от небольших фрагментов.

Наконец, вы увидите стрелки, как, например, во фразе: “Выберите команду меню File⇒Open With⇒Notepad”. Это означает, что необходимо выбрать меню File. Затем из появившегося раскрывающегося меню выбрать Open With, и наконец, из следующего подменю выбрать Notepad.

Что делать дальше

Очевидно, что первым шагом должно быть изучение языка C# (в идеале используя для этого книгу *C# 2005 для чайников*, конечно). Вы можете потратить несколько месяцев на написание простых программ C#, прежде чем сделать следующий шаг — освоить создание приложений Windows. Вам придется потратить еще много месяцев на приложения Windows, прежде чем вы начнете создавать программы, предназначенные для распространения через Интернет.

Тем временем вы можете поддерживать свои знания языка C# несколькими способами. Прежде всего, обратитесь к официальному источнику <http://msdn.microsoft.com/msdn>. Кроме того, на различных Web-сайтах для программистов имеется обширный материал по языку C#, включая живые обсуждения разных вопросов — от того, как сохранить исходный файл, и до сравнения свойств детерминистической и недетерминистической сборки мусора. Вот список нескольких больших сайтов по C#:

- ✓ www.gotdotnet.com, официальный Web-сайт группы разработчиков платформы .NET;
- ✓ <http://msdn.microsoft.com>, который направит вас на соответствующие сайты групп разработчиков, включая C# и платформу .NET;
- ✓ <http://blogs.msdn.com/csharpfaq>, блог “Часто задаваемые вопросы по C#”;
- ✓ <http://msdn.microsoft.com/vcsharp/team/blogs>, который содержит личные блоги членов группы разработки C#;
- ✓ www.cs2themax.com.

Один из авторов книги поддерживает Web-сайт www.chucksphar.com, содержащий ряд часто задаваемых вопросов (FAQ). Если вы столкнетесь с чем-то, чего не смо-

жете понять, попробуйте посетить этот сайт — возможно, в FAQ уже есть ответ на ваш вопрос. Кроме того, сайт содержит список ошибок, которые могли пробраться в книгу. И наконец — имеется в виду действительно крайний случай — вы можете найти ссылку на адреса электронной почты авторов и написать им, если не сможете найти ответ на свой вопрос на сайте.