



工业和信息化部普通高等教育“十二五”规划教材立项项目
普通高等学校计算机教育“十二五”规划教材

Visual C#.NET 程序设计教程 (第2版)

VISUAL C#.NET PROGRAMMING
(2nd edition)

罗福强 白忠建 杨剑 ◆ 编著



普.通.高.等.学.校
计算机教育“十二五”规划教材

Visual C#.NET 程序设计教程

(第2版)

VISUAL C#.NET PROGRAMMING
(2nd edition)

罗福强 白忠建 杨剑 ◆ 编著

人 民 邮 电 出 版 社
北 京

图书在版编目 (CIP) 数据

Visual C#.NET程序设计教程 / 罗福强, 白忠建, 杨剑编著. -- 2版. -- 北京: 人民邮电出版社, 2012.8
普通高等学校计算机教育“十二五”规划教材
ISBN 978-7-115-27760-2

I. ①V… II. ①罗… ②白… ③杨… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第066937号

内 容 提 要

C#是微软公司推出的新一代编程语言。它在 C++的基础之上重新打造, 成为一门全新的完全面向对象的程序设计语言, 能够提供更高的可靠性和安全性, 不仅能用于开发传统的控制台应用程序和 Windows 应用程序, 还用于开发 Web 应用程序、Silverlight 动画和 XNA 游戏。

本书第2版分为3个部分, 共14章。第1章~第3章为第1部分, 主要介绍C#的基本语法; 第4章~第8章为第2部分, 重点介绍面向对象的程序设计方法和基于事件的程序设计方法; 第9章~第14章为第3部分, 主要介绍C#的高级实用技术, 包括多线程编程技术、Windows 程序设计技术、数据库访问与编程技术(包含LINQ编程)、文件操作与编程技术(包含XML编程)、TCP/IP 网络应用编程技术、GDI+与多媒体编程技术等。本书内容丰富, 可操作性强, 叙述简洁流畅, 没有晦涩的术语, 所有案例精心设计, 能够使学生轻松、愉快地掌握C#的基本语法、编程方法和应用技巧。

本书可作为高等院校计算机相关专业学生的教材, 也可作为初、中级读者和相关专业培训学员学习的参考用书。

普通高等学校计算机教育“十二五”规划教材

Visual C#.NET 程序设计教程 (第2版)

- ◆ 编 著 罗福强 白忠建 杨 剑
责任编辑 刘 博
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
- ◆ 开本: 787×1092 1/16
印张: 23.5 2012年8月第2版
字数: 592千字 2012年8月河北第1次印刷

ISBN 978-7-115-27760-2

定价: 45.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前 言

C#是微软公司推出的基于.NET Framework 之上的新一代编程语言。在保持了C++强大功能的同时,又整合了Java语言的所有优点,成为一种全新的完全面向对象的高级语言。它简单、安全、灵活、功能强大,能够快速地开发各种应用软件。它解决了存在于许多程序设计语言中的问题,如安全问题、可靠性问题、跨平台、跨语言的兼容性问题。相对于C++来说,C#更容易理解和接受;相对于Java来说,C#更容易使用,开发软件的效率更高。C#既能用于开发传统的控制台应用程序、Windows 应用程序、Web 应用程序,还能开发 COM+组件、XML Web 服务等。特别是C#与Web技术、WPF技术、WCF技术、Silverlight技术和XNA技术紧密结合,使之成为如今Web 2.0时代最强大的、最高效的程序设计语言。

2009年,我们结合当时的教学和应用开发需求,编写了本书第1版,受到广大师生的欢迎。在此基础之上,我们开展了“C#程序设计”精品课程项目建设,并通过省级精品课程评估。随着Visual Studio.NET的不断升级,C#功能越来越强,应用越来越广。结合广大师生的反馈意见以及新的教学和应用开发经验,我们制订了全新的修订方案,重新编写了本教材的第2版,并成功入选四川省高等教育质量工程建设项目。

本书第2版以Visual Studio.NET 2010和C# 4.0为蓝本,以第1版为基础进行全面调整、修改和优化,并遵循以下5点基本思想:第一,面向应用型本科院校学生,立足于把C#的语法讲透彻、讲清楚,文字叙述要简练;第二,紧紧围绕面向对象思想和可视化设计方法展开教材内容;第三,书中所有案例均精心设计,不仅代码完整,还贴近学生实际生活;第四,坚持零起点原则,学生可以在没有C/C++基础的情况下使用本书;第五,坚持应用为纲,全面展示C#在各方面的编程应用。

针对第1版,本书第2版修订的关键点包括:一、将第1版的第2章拆分为两章,使C#的基础语法更加全面,更有助于无C/C++基础的师生学习C#,同时扩展了C#基础语法的深度;二、由于有关委托的概念与基于事件驱动编程的思想紧密相连,为此将第1版第4章有关委托的内容与第6章合并,整合之后放到第8章完整介绍;三、第1版将程序的调试和异常处理放到第9章,根据广大师生的建议,我们把它提前到第7章;四、多线程编程技术在网络通信、数据访问与存储、图像处理、云计算、动漫与游戏开发等众多应用领域中的作用越来越重要,为此我们增强了多线程编程的介绍;五、第1版设计的教学课时为68学时,重点为基础语法,而应用技术不突出,为此我们将教学课时设计为96课时,增长1/3的教学内容,使之具有更强的可裁剪性,实际教学时可更灵活地安排教学内容;六、所有教学案例和实训任务被优化组合,甚至重新设计,以突出编程方法和技巧的实用性和完整性,例如新版的第10章、第11章的案例就完整地实现了“学生成绩管理系统”的开发。

本书第2版分为3个部分,共14章。第1章~第3章为第1部分,主要介绍C#的基本语法;第4章~第8章为第2部分,重点介绍面向对象的程序设计方法和基于事件的程序设计方法;第9章~第14章为第3部分,主要介绍C#的高级实用技术,包括多线程编程技术、Windows 程序设计技术、数据库访问与编程技术(包含LINQ编程)、文件操作与编程技术(包含XML编程)、TCP/IP 网络应用编程技术、GDI+与多媒体编程技术等。

本书第2版具有以下4个鲜明特点:第一,知识结构完整,根据循序渐进的认识规律设计编写内容及顺序;第二,提供了大量的实例,所有实例程序都是完整的,都是通过Visual Studio.NET 2010调试的,并给出了运行效果,其中部分复杂的实例还有详细的分析,以帮助读者理解程序算法,并学会程序设计;第三,全书配备了丰富的、符合教学实际的、能真正培养学生编程能力的实训任务;第四,全书各章节配备了丰富的标准化习题,以单项选择题和判断题为主,形成了数百道题库,因此特别方便教学和考试。

本书可作为高等院校C#程序设计课程的教材或参考资料,也可供软件开发人员参考使用。

本书为省级精品课程指定教材,并入选四川省质量工程建设项目,获电子科技大学成都学院教材建设项目资助。参与本书编写工作的有:罗福强、白忠建、杨剑、胡德坤等老师。其中,罗福强编写了第10章、第11章、第12章、第13章、第14章,白忠建编写第9章,杨剑编写了第2章、第3章、第4章、第5章、第6章、第7章和第8章,胡德坤编写了第1章和所有上机实验。本书由罗福强负责主编工作,制定了修改方案和编写大纲,进行了全书统稿和审校。

由于时间仓促,书中难免有不妥之处,我们殷切地期望读者提出中肯的意见,联系方式:LFQ501@sohu.com。

编者

2012年1月

目 录

第 1 章 C#概述.....1

1.1 C#简介.....1

1.1.1 .NET 概述.....1

1.1.2 C#语言的发展.....3

1.1.3 C#语言的特点.....3

1.2 C#程序的开发环境.....5

1.2.1 Visual Studio.NET 的基本操作.....5

1.2.2 Visual Studio.NET 解决方案和 项目文件的组织结构.....8

1.3 C#程序入门.....8

1.3.1 一个简单的 C#控制台应用程序.....8

1.3.2 一个简单的 Win 32 应用程序.....11

1.3.3 一个具有输入功能的 Win 32 应用程序.....13

1.3.4 一个简单的 Web 应用程序.....15

1.3.5 C#程序的特点.....16

习题.....17

上机实验 1.....19

第 2 章 C#程序设计基础.....20

2.1 常量与变量.....20

2.1.1 常量.....21

2.1.2 变量.....22

2.2 C#的数据类型.....24

2.2.1 简单类型.....24

2.2.2 枚举型 enum.....25

2.2.3 结构型 struct.....26

2.2.4 数据类型转换.....27

2.3 运算符与表达式.....29

2.3.1 算术运算符与表达式.....29

2.3.2 赋值运算符与表达式.....30

2.3.3 关系运算符与表达式.....32

2.3.4 逻辑运算符与表达式.....32

2.3.5 运算符优先级.....33

2.4 数组和字符串.....33

2.4.1 一维数组.....33

2.4.2 多维数组.....36

2.4.3 数组型的数组.....37

2.4.4 字符串 string.....38

习题.....40

上机实验 2.....42

第 3 章 C#程序的流程控制.....44

3.1 C#程序的分支语句.....44

3.1.1 if 语句.....44

3.1.2 多分支 if...else if 语句.....46

3.1.3 switch 语句.....48

3.1.4 分支语句的嵌套.....50

3.2 C#程序的循环语句.....51

3.2.1 while 语句.....52

3.2.2 do/while 语句.....53

3.2.3 for 语句.....54

3.2.4 foreach 语句.....56

3.2.5 循环语句的嵌套.....57

3.3 跳转语句.....58

3.3.1 break 语句.....59

3.3.2 continue 语句.....60

习题.....61

上机实验 3.....64

第 4 章 面向对象程序设计入门.....66

4.1 面向对象的基本概念.....66

4.1.1 对象.....67

4.1.2 事件与方法.....67

4.1.3 类与对象.....68

4.1.4 抽象、封装、继承与多态.....68

4.2 类的定义.....70

4.2.1 类的声明和实例化.....70

4.2.2 类的数据成员和属性.....72

4.2.3 类的可访问性	74	6.2 索引器	139
4.2.4 值类型与引用类型	75	6.2.1 索引器的定义与使用	139
4.3 类的方法	76	6.2.2 索引器与属性的比较	143
4.3.1 方法的声明与调用	76	6.3 泛型	143
4.3.2 方法的参数传递	78	6.3.1 泛型概述	144
4.3.3 方法的重载	85	6.3.2 泛型集合	144
4.4 构造函数	86	6.3.3 泛型类、泛型方法和泛型接口	146
4.4.1 构造函数的声明和调用	87	习题	151
4.4.2 构造函数的重载	87	上机实验 6	152
4.4.3 构造函数与只读字段	89	第 7 章 程序调试与异常处理	154
4.4.4 对象的生命周期和析构函数	91	7.1 程序错误	154
习题	92	7.1.1 程序错误分类	154
上机实验 4	94	7.1.2 调试程序错误	156
第 5 章 面向对象的高级程序设计	99	7.2 程序的异常处理	158
5.1 静态成员与静态类	99	7.2.1 异常的概念	158
5.1.1 静态成员	99	7.2.2 异常类	159
5.1.2 静态构造函数	101	7.2.3 try-catch 语句	160
5.1.3 静态类	102	7.2.4 finally 语句	161
5.2 类的继承性与多态性	102	7.2.5 throw 语句与抛出异常	162
5.2.1 类的继承性	103	习题	164
5.2.2 类的多态性	107	上机实验 7	165
5.3 抽象类与接口	110	第 8 章 基于事件驱动的程序设计技术	168
5.3.1 抽象类	110	8.1 基于事件的编程思想	168
5.3.2 接口	113	8.2 委托	170
5.3.3 抽象类与接口的比较	119	8.2.1 委托的概述	170
5.4 嵌套类、分部类与命名空间	119	8.2.2 委托的声明、实例化与使用	170
5.4.1 嵌套类	119	8.2.3 多路广播与委托的组合	172
5.4.2 分部类	121	8.3 事件	173
5.4.3 命名空间	122	8.3.1 声明事件	173
习题	124	8.3.2 订阅事件	175
上机实验 5	126	8.3.3 触发事件	176
第 6 章 集合、索引器与泛型	131	8.4 基于事件的 Windows 编程	178
6.1 集合	131	8.4.1 Windows 应用程序概述	179
6.1.1 集合概述	131	8.4.2 Windows 窗体与事件驱动编程	180
6.1.2 ArrayList	132	习题	184
6.1.3 哈希表 Hashtable	136	上机实验 8	185
6.1.4 栈和队列	138		

第 9 章 多线程和异步编程.....191

- 9.1 多线程的概念.....191
 - 9.1.1 线程和进程.....191
 - 9.1.2 线程的生命周期和状态.....192
 - 9.1.3 线程的优先级.....192
- 9.2 线程创建与控制.....193
 - 9.2.1 创建和启动线程.....193
 - 9.2.2 控制线程.....194
- 9.3 多线程的同步.....197
 - 9.3.1 线程安全.....197
 - 9.3.2 线程同步策略.....198
- 9.4 线程池.....202
 - 9.4.1 线程池管理.....203
 - 9.4.2 ThreadPool 类.....203
- 9.5 异步编程.....204
 - 9.5.1 异步编程和多线程.....204
 - 9.5.2 异步编程模式.....204
- 习题.....208
- 上机实验 9.....209

第 10 章 Windows 程序的
界面设计.....211

- 10.1 窗体与控件概述.....211
 - 10.1.1 Windows 窗体.....211
 - 10.1.2 窗体的控件.....213
- 10.2 简单输入与输出控件.....216
 - 10.2.1 文本显示控件.....216
 - 10.2.2 文本编辑控件.....216
 - 10.2.3 按钮控件.....218
 - 10.2.4 图片显示控件.....221
- 10.3 列表与选择类控件.....222
 - 10.3.1 选项与选项组.....223
 - 10.3.2 列表类控件.....224
 - 10.3.3 其他选择类控件.....226
 - 10.3.4 历与计时器控件.....226
- 10.4 对话框及其选项卡设计.....230
 - 10.4.1 对话框概述.....230
 - 10.4.2 对话框的选项卡.....231
 - 10.4.3 消息框.....231

- 10.4.4 通用对话框.....236
- 10.5 菜单、工具栏和状态栏.....238
 - 10.5.1 菜单.....239
 - 10.5.2 工具栏.....240
 - 10.5.3 状态栏.....241
- 10.6 SDI 和 MDI 应用程序.....244
 - 10.6.1 创建 SDI 应用程序.....244
 - 10.6.2 创建 MDI 应用程序.....245
- 习题.....246
- 上机实验 10.....248

第 11 章 C#数据库编程技术.....251

- 11.1 ADO.NET 概述.....251
 - 11.1.1 ADO.NET 的架构.....251
 - 11.1.2 ADO.NET 的一般使用步骤.....252
- 11.2 使用 ADO.NET 访问数据库.....255
 - 11.2.1 使用 Connection 连接数据库.....255
 - 11.2.2 使用 Command 对象访问数据库.....256
 - 11.2.3 使用 DataReader 对象访问
数据库.....260
 - 11.2.4 使用 DataAdapter 与 DataSet
对象操作数据库.....264
- 11.3 LINQ to SQL 编程.....269
 - 11.3.1 LINQ 概述.....269
 - 11.3.2 LINQ 的查询子句与表达式.....270
 - 11.3.3 LINQ to SQL 的应用.....272
- 习题.....281
- 上机实验 11.....283

第 12 章 文件操作与编程技术.....285

- 12.1 文件的输入/输出.....285
 - 12.1.1 文件 I/O 与流.....285
 - 12.1.2 读写文本文件.....286
 - 12.1.3 读写二进制文件.....287
 - 12.1.4 对象的序列化.....289
- 12.2 文件操作控件.....292
 - 12.2.1 SaveFileDialog 与 OpenFileDialog
控件.....292
 - 12.2.2 FolderBrowseDialog 控件.....294
 - 12.2.3 应用实例：简易的写字板程序.....295

12.3 XML 文档编程.....	298	13.4.2 FTP 客户端的实现.....	334
12.3.1 XML 概述.....	298	习题.....	339
12.3.2 XML 文档的创建.....	300	上机实验 13.....	341
12.3.3 XML 文档的查询.....	303	第 14 章 多媒体编程技术	342
12.3.4 XML 文档的编辑.....	304	14.1 GDI+绘图.....	342
习题.....	307	14.1.1 GDI+概述.....	342
上机实验 12.....	308	14.1.2 创建 Graphics 对象.....	344
第 13 章 网络应用编程技术	311	14.1.3 颜料、钢笔和画笔.....	345
13.1 网络编程基础.....	311	14.1.4 线条与图形的绘制.....	346
13.1.1 计算机网络的概述.....	311	14.1.5 图像和文本的绘制.....	353
13.1.2 计算机网络的通信协议.....	312	14.1.6 坐标系统及变换.....	356
13.1.3 System.Net 概述.....	313	14.2 Windows Media Player 组件的使用.....	361
13.2 Socket 编程.....	316	14.2.1 Windows Media Player 组件的 介绍.....	361
13.2.1 Socket 编程概述.....	316	14.2.2 Windows Media Player 组件的 使用.....	363
13.2.2 TCP 应用编程.....	319	习题.....	365
13.2.3 UDP 应用编程.....	323	上机实验 14.....	367
13.3 电子邮件的发送和接收.....	326	参考文献	368
13.3.1 发送邮件.....	326		
13.3.2 接收邮件.....	328		
13.4 FTP 与文件传输编程.....	334		
13.4.1 WebRequest 和 WebResponse 类.....	334		

第 1 章

C#概述

总体要求

- 了解 .NET Framework 以及 C# 语言的特点及其发展。
- 熟悉 Visual Studio.NET 2010 的操作方法。
- 掌握 C# 程序的创建、编辑、编译、运行等基本操作过程。
- 理解 C# 程序的结构及其特点。

学习重点

- .NET Framework、C#、C# 程序的特点。
- C# 程序在 Visual Studio.NET 2010 中的操作方法。

1.1 C#简介

1.1.1 .NET 概述

.NET 平台是由 Microsoft 推出的全新的应用程序开发平台，可用来构建和运行新一代 Microsoft Windows 和 Web 应用程序。它建立在开放体系结构基础之上，集 Microsoft 在软件领域的主要技术成就于一身。

.NET 平台的核心是 .NET Framework，它为 .NET 平台下应用程序的运行提供基本框架，如果把 Windows 操作系统比作一幢摩天大楼的地基，那么 .NET Framework 就是摩天大楼中由钢筋和混凝土搭成的框架。

Visual Studio.NET 是 .NET 平台的主要开发工具，由于 .NET 平台是建立在开放体系结构基础之上的，因此应用程序开发人员也可以使用其他开发工具。

1. .NET Framework 的结构

.NET Framework 以微软的 Windows 操作系统为基础，由不同的组件组成，如图 1-1 所示，能够与 Windows 的各种应用程序服务组件（如消息队列服务、COM+ 组件服务、Internet 信息服务（IIS）、Windows 管理工具等）整合，以开发各种应用程序。

在 .NET Framework 的最顶层是程序设计语言，.NET Framework 支持诸如 VB、C#、C++、J#、Perl 等几十种高级程序设计语言。在 Visual Studio.NET 开发环境中，可直接使用 VB、C#、C++、F#、Jscript 共 5 种语言开发应用程序。

.NET Framework 具有两个主要组件：公共语言运行时 CLR（Common Language Runtime）

和 .NET Framework 类库,除此之外还包括 ADO.NET、ASP.NET、XML Web 服务等。

公共语言运行时 CLR 是 .NET Framework 的基础,是应用程序与操作系统之间的“中间人”,它为应用程序提供内存管理、线程管理和远程处理等核心服务。在 .NET 平台上,应用程序无论使用何种语言编写,在编译时都会被语言编译器编译成 MSIL (微软中间语言代码),在运行应用程序时 CLR 自动启用 JIT 编译器 (Just In Time) 把 MSIL 再次编译成操作系统能够识别的本地机器语言代码 (简称本地代码),然后运行并返回运行结果,如图 1-2 所示。因此,CLR 是所有 .NET 应用程序的托管环境。这种运行在 .NET 之上的应用程序被称为托管应用程序,而传统的直接在操作系统基础中运行的应用程序,则被称为非托管应用程序。

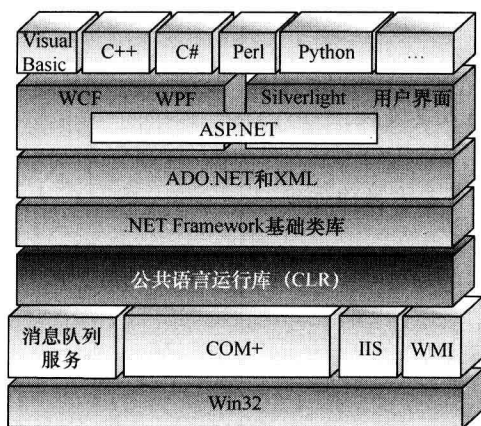


图 1-1 .NET Framework 的结构

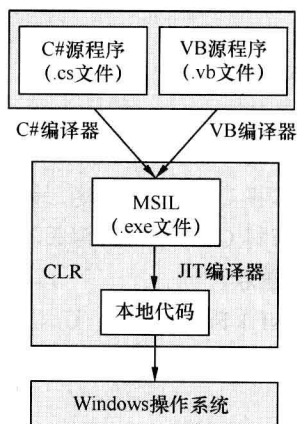


图 1-2 CLR 的工作机制

.NET Framework 类库是一个综合性的面向对象的可重用类型集合,利用它可以开发多种应用程序,包括传统的命令行或图形用户界面 (GUI) 应用程序,也包括 Web 应用程序。

ADO.NET 是 .NET Framework 提供的微软新一代的面向对象的数据处理技术,利用它可以简便、快捷地开发数据库应用程序。

ASP.NET 是 .NET Framework 提供的全新的 Web 应用程序开发技术,利用它开发 Web 应用程序,如同开发 Windows 应用程序一样简单。

WCF (Windows Communication Foundation)、WPF (Windows Presentation Foundation) 以及 Silverlight 等技术是微软推出的全新 .NET 技术。WCF 可以理解 Windows 通讯接口,它整合了 TCP/IP、XML、SOAP 等技术,因此简化了 XML Web 服务的设计与实现。WPF 为用户界面、2D/3D 图形、文档和媒体提供了统一的描述和操作方法。Silverlight 为开发具有专业图形、音频和视频处理的 Web 应用程序提供了全新的解决方案。

2. .NET Framework 的优点

在 .NET 平台诞生之前,虽然 Internet 已经出现,但很少有应用程序能运行在各种不同类型的客户端上,也不能和其他应用程序进行无缝集成。这种局限性导致开发人员花费大量的时间去改写应用程序,以保证它们能在各种客户端和平台上运行,而不是利用这些时间去设计新的应用程序。.NET Framework 的最大特色就在于它为应用程序开发人员提供了一个真正与平台无关的开发环境。使用 .NET Framework 开发应用程序有以下优点。

(1) 基于 Web 的标准

.NET Framework 完全支持现有的 Internet 技术,包括 HTML (超文本标记语言)、HTTP (超

文本传输协议)、XML(可扩展标记语言)、SOAP(简单对象访问协议)、XSLT(可扩展样式表语言转换)、XPath(XML 路径语言)和其他 Web 标准。

(2) 使用统一的应用程序模型

任何与 .NET 兼容的语言都可以使用 .NET Framework 类库。.NET Framework 为 Windows 应用程序、Web 应用程序和 XML Web 服务提供了统一的应用程序模型,因此同一段代码可被这些应用程序无障碍地使用。

(3) 便于开发人员使用

在 .NET Framework 中,代码被组织在不同的命名空间和类中,而命名空间采用树形结构,以便开发人员引用。当开发人员企图调用 .NET Framework 类库的类时,只需将该类属性命名空间添加到引用解决方案中即可。

(4) 可扩展类

.NET Framework 提供了通用类型系统,它根据面向对象的思想把一个命名空间或类中代码的实现细节隐藏,开发人员可以通过继承来访问类库中的类,也可以扩展类库中的类,甚至于构建自己的类库。

1.1.2 C#语言的发展

在过去的二十年里,C 和 C++ 已经成为在商业软件的开发领域中使用最广泛的语言。它们为程序员提供了十分灵活的操作,不过同时也牺牲了一定的效率。与 Visual Basic 等语言相比,同等级别的 C/C++ 应用程序往往需要更长时间来开发。由于 C/C++ 语言的复杂性,许多程序员都试图寻找一种新的语言,希望能在功能与效率之间找到一个更为理想的平衡点。

目前有些语言,以牺牲灵活性的代价来提高效率。可是这些灵活性正是 C/C++ 程序员所需要的。这些解决方案对编程人员的限制过多(如屏蔽一些底层代码控制的机制),其所提供的功能难以令人满意。这些语言无法方便地同原来的系统交互,也无法与当前的网络编程很好地结合。

对于 C/C++ 用户来说,最理想的解决方案无疑是在快速开发的同时,又可以调用底层平台的所有功能。他们想要一种和最新的网络标准保持同步,并且能和已有的应用程序良好整合的环境。另外,一些 C/C++ 开发人员还需要在必要的时候进行一些底层的编程。

C#(读作 C Sharp)是微软对这一问题的解决方案。C# 是一种最新的、面向对象的编程语言。它是一种简单但功能强大的编程语言,使得程序员可以快速地编写各种基于 Microsoft.NET 平台的应用程序。

C# 是从 C 和 C++ 语言演化而来。它在语句、表达式和运算符方面使用了许多 C++ 功能。它在类型安全性、版本转换、事件和垃圾回收等方面进行了相当大的改进和创新。它提供对常用 API(如 .NET Framework、COM+ 等)的访问。

C# 自推出以来,已得到不断的改进和优化,通常同 .NET Framework 一起,随 Visual Studio.NET 一起发布。目前,比较流行的 C# 主要有 4 个版本,它们分别是:C# 1.0、C# 2.0、C# 3.0 和 C# 4.0。C# 1.0 同 .NET Framework 1.1 一起,随 Visual Studio.NET 2003 一起发布。C# 2.0 同 .NET Framework 2.0 一起,随 Visual Studio.NET 2005 一起发布。C# 3.0 同 .NET Framework 3.5 一起,随 Visual Studio.NET 2008 一起发布。而 C# 4.0 同 .NET Framework 4.0 一起,随 Visual Studio.NET 2010 一起发布。

本书以 C# 4.0、.NET Framework 4.0 和 Visual Studio.NET 2010 为范本,所有案例均是在 Visual Studio.NET 2010 中经过调试运行无误的。

1.1.3 C#语言的特点

C# 是一款简洁、类型安全的面向对象的语言,开发人员可以使用它来构建在 .NET Framework

上运行的各种安全、可靠的应用程序,包括控制台应用程序、Windows 窗体应用程序、WPF 应用程序、Web 应用程序、Silverlight 应用程序、WCF 服务、分布式组件、客户端/服务器应用程序、数据库应用程序等。

作为一款面向对象的语言,C#支持封装、继承和多态性的概念。所有的变量和方法,包括 Main 方法(应用程序的入口点),都封装在类定义中。C#程序的生成过程比 C 和 C++简单,比 Java 更为灵活。没有单独的头文件,也不要求按照特定顺序声明方法和类型。C#源文件可以定义任意数量的类、结构、接口和事件。

相对其他计算机程序设计语言来说,C#具有如下优点。

(1) 语法简洁

C#最大特色是抛弃了 C/C++的指针,不允许代码直接操作内存,C#自动计算数组或集合的长度,有效避免了内存地址或数组下标越界的问题。C#统一了对结构型、类及其成员的引用操作符,只有一个“.”,使代码书写更简单。C#没有全局函数,也没有全局变量,这使代码具有更好的可读性,也减少了因命名而造成的冲突。

(2) 完全面向对象设计

C#使用 Object(根类型)来统一所有数据类型,通过装箱和拆箱机制来完成对象操作或数据类型转换,即使 bool、byte、char、int、float、double 等简单数据类型都封装为 Boolean、Byte、Char、Int32、Single、Double 等结构型。C#只允许单一继承,不允许一个类从多个基类派生,从而根本上避免了类型定义的混乱问题。C#的类模型建立在 .NET Framework 之上,借助于 Framework 的公共类库,使之成为构建各种应用程序组件的理想之选——无论是高级的商业对象,还是系统级的应用程序。

(3) 与 Web 紧密结合

设计 C#的一个重要因素就是使应用程序的解决方案与 Web 标准相统一,使程序设计与 HTML、XML、SOAP、DOM、XPath 等 Web 技术紧密结合。C#首先统一了传统的命令行或 Windows 应用程序,以及 Web 应用程序的开发模式,在此基础之上,微软公司又推出了 WPF、WCF、Silverlight、Xbox 等新技术。这些技术使得 C#不仅能开发普通应用程序,还能在网络通信、动画制作、游戏开发、图像处理、多媒体应用、移动设备领域等发挥重要作用。

(4) 完善的安全性和错误处理

程序设计语言的安全性和错误处理能力是衡量一种语言是否优秀的重要依据。任何人都会犯错误,特别是在程序编写过程中,出错总是难免的,因此有经验的程序员总是不断地调试程序,并加以改进,直到完成。C#借助 Visual Studio.NET 的智能感知技术,可以消除在程序编写过程中的许多常见错误。另外,C#还提供统一的异常类 Exception 来管理程序在运行过程中产生的错误。

在安全性方面,C#提供了完整的类型安全机制,例如对象的成员变量由编译器负责初始化,而其他局部变量未经初始化则不允许使用,编译器也会进行自动检查并提示。CLR 提供垃圾回收、类型安全检查、内部代码信任机制等,允许管理员或用户根据自己的 ID 来配置安全等级,借助 CLR 这一特性,可以进一步确保 C#应用程序的安全性。

(5) 良好的可扩展性

C#具有生成持久系统级组件的能力,提供 COM+、MSMQ(微软消息队列服务)或其他技术平台支持以集成现有代码,利用 .NET Framework 的通用类型系统能够与其他程序设计语言交互操作。C#应用程序能跨语言、跨平台、跨互联网互相调用。使用 C#语言可实现具有不同专业技术背景的人员协同工作,完成软件系统的设计和开发。

C#语言允许自定义数据类型,以扩展元数据。这些元数据可以应用于任何对象。项目构建者

可以定义领域特有的属性,并把他们应用于任何语言元素——类、接口等。然后,开发人员可以编程检查每个元素的属性。这样,很多工作都变得方便多了,比如编写一个小工具来自动检查每个类或接口是否被正确定义为某个抽象商业对象的一部分,或者只是创建一份基于对象的领域特有属性的报表。定制的元数据和程序代码之间的紧密对应,有助于加强程序的预期行为和实际实现的之间的对应关系

1.2 C#程序的开发环境

Visual Studio.NET 是 C#程序的主要集成开发环境。目前,使用较多的 Visual Studio.NET 有 4 个版本: Visual Studio.NET 2003、Visual Studio.NET 2005、Visual Studio.NET 2008 和 Visual Studio.NET 2010。本节将主要介绍 Visual Studio.NET 2010 的基本操作方法。

1.2.1 Visual Studio.NET 的基本操作

当计算机安装了 Visual Studio.NET 2010 后,用户只需选择“开始→所有程序→Microsoft Visual Studio 2010→Microsoft Visual Studio 2010”系统菜单命令即可启动。

刚启动的 Visual Studio.NET 2010 的窗口由菜单栏、工具栏、工具箱、起始页、解决方案资源管理器等组成,如图 1-3 所示。其中,菜单栏列出 Visual Studio.NET 2010 的所有操作命令,工具栏则列出常用的操作命令,解决方案资源管理器用于显示将要创建的应用程序项目的文件夹结构以及文件列表,工具箱用于显示在设计应用程序操作界面时所要使用的可视化控件。



图 1-3 Visual Studio.NET 2010 窗口

1. 应用程序项目的创建与打开

(1) 新建项目

Visual Studio.NET 2010 是一个高度集成的开发工具。它集 Visual Basic、C++、C#和 F# 4 种程序设计语言为一体,可以创建这 4 种语言的应用程序。它具有创建控制台应用程序、Windows 应用程序、类库、设备应用程序、Windows 控件库、安装项目、Web 应用程序(ASP.NET 网站)、WCF 服务等功能。因此,在创建新项目之前应该先做好选择。

如果选择“文件→新建→项目”菜单命令,则先弹出“新建项目”对话框,然后根据需要选

择所有创建的项目即可,如图 1-4 所示。例如,要创建 Windows 窗体应用程序,首先在“已安装的模板”列表框中选择“Visual C#”及其子节点“Windows”,然后在中间的模板列表框中选择“Windows 窗体应用程序”,接着在“名称”文本框中输入应用程序项目的名字(如 MyFirstApp),在“位置”组合框中可设置应用程序项目的创建位置(如 D:\Demo),在“解决方案名称”文本框中输入解决方案的名称(如 MyFirstApp),最后单击“确定”按钮即可。



图 1-4 “新建项目”对话框

如果选择“文件→新建→网站”菜单命令,则先弹出“新建网站”对话框,然后根据需要进行选择所有创建的项目即可,如图 1-5 所示。

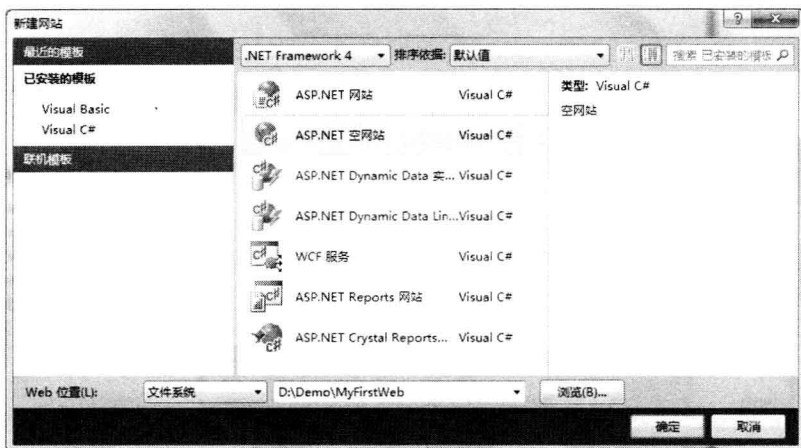


图 1-5 “新建网站”对话框

在创建网站时,应注意以下几点。

① 在列表框中选择“ASP.NET 网站”,表示创建 Web 应用程序,而选择“WCF 服务”,表示创建能通过互联网远程调用的组件。

② “位置”组合框中有 3 种选择,其中“文件系统”表示将网站创建到磁盘文件夹中,“HTTP”表示通过超文本传输协议把网站直接创建到某一个 Web 服务器中,“FTP”表示通过文件传输协议把网站创建到某一个 FTP 服务器。由于选择“HTTP”或“FTP”要求开发人员具有操作远程服务器的权限,因此通常选择“文件系统”。

(2) 打开项目

要打开一个已经存在的应用程序项目，有多种操作方法。

选择“文件→打开→项目/解决方案”菜单命令，以打开控制台应用程序、Windows 窗体应用程序等。选择“文件→打开→网站”菜单命令，以打开 Web 应用程序项目。

当然，用户也可以通过“文件→最近使用的项目和解决方案”菜单命令，来打开已存在的项目，也可以通过“起始页”中选择最近的项目来打开。

2. 应用程序项目的设计与编辑

在创建或打开项目之后，即可在 Visual Studio.NET 2010 窗口的工作区中对应用程序项目进行设计和编辑。Visual Studio.NET 2010 窗口的工作区分为 3 部分，左边是工具箱，中间是设计区，右边是解决方案资源管理器和属性窗口，如图 1-6 所示。



图 1-6 设计或编辑应用程序

通过工具箱，可设计或修改应用程序的操作界面。只需将工具箱中的控件按钮拖放到设计视图区即可。

通过解决方案资源管理器，可修改项目文件名、添加或删除项目文件。具体方法是：先在解决方案资源管理器中右击操作目标，再选择相应的快捷菜单命令。

通过“属性”窗口，可设置或修改控件或对象的各属性值。

需要编辑或修改程序源代码，可先右击“设计区”，并选择快捷菜单“查看源代码”命令，把 Visual Studio.NET 2010 的设计视图切换到源代码编辑视图即可。反过来，在源代码编辑视图中，右击文本编辑窗口，然后选择快捷菜单“查看设计器”命令，即可切换到设计视图。

当应用程序的设计和编辑操作完成之后，单击工具栏上的“全部保存”，即可把相应的操作结果保存到磁盘中。

3. 应用程序项目的调试与生成

在应用程序开发过程中或开发结束后，可能需要看看设计的某个部分的运行结果，或看看程序的运行情况，此时只需选择“调试→启动调试”菜单命令，也可以直接按快捷键<F5>，进行调试。当程序中有错误时，Visual Studio.NET 2010 在“错误列表”窗口中显示语言编译器的错误提示信息，程序员可根据错误提示信息修改程序。值得注意的是，语言编译器只能发现源程序中的语法错误，并不能发现程序逻辑错误。

当设计已经全部完成后，可选择“生成→生成×××”菜单命令以生成解决方案（其中，

“×××”为项目名)。此时，语言编译器会在应用程序项目所在文件夹下的 bin\debug 文件中生成一个可执行文件(.exe)。这样应用程序项目的设计就基本完成了。

注意，Visual Studio.NET 2010 会自动根据在创建应用程序项目时所选择的语言来选择相应的语言编译器对源程序进行编译。通过语言编译器编译之后，源代码被翻译成.NET 公共语言运行库(CLR)能够识别的中间语言代码。因此，Visual Studio.NET 生成的可执行文件(.exe)与传统的可执行文件(.exe)虽然扩展名相同，但有很大的区别，传统的.exe 文件可直接在操作系统平台上运行，而 Visual Studio.NET 生成的.exe 文件离不开.NET Framework，必须先安装.NET Framework 之后才能运行。Windows 7 自带.NET Framework，因此不需要单独安装.NET Framework。

1.2.2 Visual Studio.NET 解决方案和项目文件的组织结构

Visual Studio.NET 2010 通过解决方案和项目来管理一个正在开发的软件项目。在 Visual Studio.NET 2010 中，一个解决方案代表一个正在开发的异常庞杂的软件系统，一个项目可能只是正在开发的软件系统中的一个子系统。因此，一个解决方案可以把多个项目组织起来，而一个项目可以把一个子系统的所有文件管理起来。Visual Studio.NET 2010 支持多种文件类型及与它们相关的扩展类型。表 1-1 列出了.NET 应用程序特有的一些常用的文件类型。

表 1-1 Visual Studio.NET 中的常用文件类型

扩展名	名称	描述
.sln	Visual Studio.NET 解决方案文件	.sln 文件为解决方案资源管理器提供显示管理文件的图形接口所需的信息。打开.sln 文件，能快捷地打开整个项目的所有文件
.csproj	Visual C#项目文件	一个特殊的 XML 文档，主要用来控制项目的生成
.cs	Visual C#源代码文件	表示 C#源程序文件、Windows 窗体文件、Windows 用户控件文件、类文件、接口文件等
.resx	资源文件	包括一个 Windows 窗体、Web 窗体等文件的资源信息
.aspx	Web 窗体文件	表示 Web 窗体，由 HTML 标记、Web Server 控件、脚本组成
.asmx	XML Web 服务文件	表示 Web 服务，它链接一个特定.cs 文件，在这个.cs 文件中包含了供 Internet 调用的方法函数代码

1.3 C#程序入门

使用 C#语言，可创建多种应用程序，包括控制台应用程序、Windows 窗体应用程序、WPF 应用程序、Web 应用程序和 Silverlight 应用程序等。在 Visual Studio.NET 2010 中，这些应用程序的操作模式基本上相同。本节将使用 4 个实例来展现 C#程序的特点及其一般操作方法。

1.3.1 一个简单的 C#控制台应用程序

【实例 1-1】设计一个 C#控制台应用程序，实现如图 1-7 所示的效果。

【操作步骤】详细操作步骤如下。

- (1) 启动 Visual Studio.NET 2010
- (2) 新建项目

首先，选择“文件→新建→项目”菜单命令，弹出“新建项目”对话框后，在“已安装的模

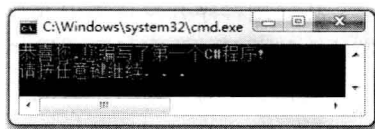


图 1-7 控制应用程序的运行效果