



普通高等教育“十一五”计算机类规划教材



C#.NET程序设计

C#.NET Chengxu Sheji

■ 李旗 编著



机械工业出版社
CHINA MACHINE PRESS

TP312/2633

2008

普通高等教育“十一五”计算机类规划教材

C#.NET 程序设计

李 旗 编著

机 械 工 业 出 版 社

本书全面介绍了 C#编程的原理和方法, 涉及类与对象、控制语句、方法、数组、继承、多态、异常处理、数据库编程、网络编程、异步多线程、XML、Web 服务、泛型、集合等基本概念及应用方法。本书采用尽早接触类与对象的方法, 循序渐进地介绍面向对象的设计方法, 介绍.NET 框架 2.0 和 Visual C# 2005 的关键新特性。采用理论与实例结合、相互渗透、逐步引导的方法, 实例丰富、图文并茂、内容繁简得当, 由浅入深, 读者十分容易入门并逐步精通。

本书可作为高等院校相关专业的编程语言教材和 C#编程教材, 也可作为软件设计人员进行 C#程序开发的参考资料。为方便教师教学, 本书配有免费教学课件, 欢迎选用该书作为教材的老师索取, 索取邮箱: llm7785@sina.com。

图书在版编目(CIP)数据

C#. NET 程序设计/李旗编著. —北京: 机械工业出版社, 2008. 1
普通高等教育“十一五”计算机类规划教材
ISBN 978 - 7 - 111 - 22779 - 3

I. C… II. 李… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 174906 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
责任编辑: 刘丽敏 责任校对: 陈延翔 封面设计: 张 静
责任印制: 李 妍
北京蓝海印刷有限公司印刷
2008 年 1 月第 1 版第 1 次印刷
184mm × 260mm · 18 印张 · 441 千字
标准书号: ISBN 978 - 7 - 111 - 22779 - 3
定价: 27.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换
销售服务热线电话: (010) 68326294
购书热线电话: (010) 88379639 88379641 88379643
编辑热线电话: (010) 88379727
封面无防伪标均为盗版

前 言

1. 本书编写背景

目前,国内高校编程类课程的教学内容一般比较偏重于编程语言理论的教学,教学模式大多是传统的模式,结果导致学生的编程设计和应用的能力不够。笔者认为学习此类课程主要应掌握实际动手能力和分析问题、解决问题的能力。因此,很有必要改变课程的教学内容和教学模式,用先进的教学理念和方法培养一流人才。本教材就是在此改革思路下编写的。

2. 本书特点

1) 把课程教学目标纳入该专业技能培养目标中,目标具体、明确。课程的目标是使学生学完本书后便可以编制各种不同类型的企业级应用程序和系统。

2) 在内容的选择上,侧重讲述当前软件开发使用最多的技术,例如数据库系统、网络服务应用程序等。

3) 内容新,体现了.NET 框架 2.0 最新类库和 Visual Studio 2005 的新开发技术环境。对目前主流的开发技术,如跨平台的面向服务的 XML 和 Web 服务技术,给予详细的介绍。

4) 突出实际动手能力和实用性,突出案例(本书共有近百例应用程序实例),有些实例还有详细的步骤,根据书中的范例,可以完成类似的应用系统,进而触类旁通。

3. 本书内容提要

第 1 章 介绍 Visual Studio 2005 和 C#.NET,使用 Visual Studio 2005 创建简单的 C#应用程序, C#面向对象程序主要结构等内容。

第 2 章 介绍数据类型、类型转换、变量和表达式,常用的 string、StringBuilder 和 DateTime 数据类型。

第 3 章 介绍程序流程控制,包括选择控制、循环控制、跳转语句和异常处理语句。

第 4 章 介绍数组、集合和泛型集合,包括 Array、ArrayList 和 List 泛型类,Hashtable 类和 Dictionary 泛型类,队列 Queue 和堆栈 Stack 类及泛型类。

第 5 章 介绍常用的 Windows 窗体控件,使用控件创建 Windows 窗体应用程序。

第 6 章 介绍对象、类和结构,包括方法、静态类与静态成员、属性与索引器、继承、多态性、接口、抽象类、密封类、重载运算符。使用 Visual Studio 类关系图可视化创建对象。

第 7 章 介绍委托与事件、泛型类、结构、接口和泛型成员、可空类型,反射程序集、类型及其成员,调用实例方法。

第 8 章 介绍流(Stream)、文件(File)类、目录(Directory)等类,用于 XML 编程的 XmlReader、XmlWriter 和 XmlDocument 类,使用和管理应用程序配置文件信息。

第 9 章 介绍 ADO.NET 结构、.NET 框架数据提供程序、DataSet 数据集与 DataTable 相关类、TableAdapter 查询、Windows 窗体数据控件和数据访问可视化编程。

第 10 章 介绍进程、线程、线程池(ThreadPool)类、线程同步、使用 BackgroundWorker 实现后台操作的 Windows 应用程序、Windows 应用程序窗体控件线程安全调用。

第 11 章 介绍用于互联网编程的 WebClient 类等类、实现 TCP 服务的 TCPClient 和



TCPListener 类、创建和调用 Web 服务、使用 Web 服务创建多层分布式应用程序。

本书由齐吉泰教授担任主审，审阅了全书，为提高书稿质量付出了大量的精力和劳动，并提出了许多宝贵意见和建议，在此表示衷心的感谢。

全书紧密结合开发实例展开。初级读者可以系统学习本书，整体掌握使用 C# 语言进行应用程序开发的技术，对中、高级读者而言，本书可以作为一本开发手册使用。

编 者

目 录

前言

第 1 章 Visual Studio 2005 和 C#.NET

概述.....1

1.1 Visual Studio 2005、.NET 框架和 C#

简介.....1

1.1.1 Visual Studio 2005.....1

1.1.2 .NET 框架简介.....1

1.1.3 C#语言简介.....1

1.2 在 Visual Studio 2005 中创建 C#控制台

应用程序.....2

1.2.1 创建 HelloWorld 控制台应用程序

项目.....2

1.2.2 编写在控制台中显示消息“欢迎

使用 C#.NET 程序设计”的程序.....4

1.2.3 生成并运行控制台应用程序.....6

1.3 命名空间.....6

1.3.1 命名空间定义.....7

1.3.2 使用命名空间.....8

1.3.3 命名空间与程序集.....8

1.4 创建 Windows 窗体应用程序.....8

1.4.1 在解决方案中添加 Windows

应用程序新项目.....9

1.4.2 在 WelcomeWorld 新项目创建用户

界面.....9

1.4.3 Visual Studio 2005 自动生成代码

说明.....10

1.4.4 应用程序运行时, 窗体是如何

显示的?.....11

1.4.5 在 WelcomeWorld 项目中添加按钮

执行代码.....11

1.4.6 设置 WelcomeWorld 项目为启动

项目.....11

1.4.7 运行 WelcomeWorld 项目 Windows

应用程序.....12

1.4.8 多启动项目的设置.....12

1.5 面向对象 C#程序主要结构简介.....13

1.5.1 编写计算圆面积的程序.....13

1.5.2 C#程序的构成.....14

1.5.3 方法定义和调用.....14

1.6 程序调试.....16

1.6.1 使用调试器运行调试程序.....16

1.6.2 查看调试器中的数据.....17

1.7 编写简单面向对象 Windows 应用

程序.....17

1.7.1 创建计算圆面积的 Windows 应用

程序并设计界面.....18

1.7.2 添加自定义的计算圆面积的类.....18

1.7.3 添加 Windows 控件处理事件.....19

1.7.4 运行 CircleAreaW 项目.....20

1.8 作业.....20

第 2 章 数据类型、变量和表达式.....21

2.1 数据类型的值类型.....21

2.1.1 整数类型.....21

2.1.2 布尔类型.....21

2.1.3 字符类型.....21

2.1.4 实数类型.....21

2.1.5 结构类型.....22

2.1.6 枚举类型.....23

2.2 引用类型.....24

2.2.1 类.....24

2.2.2 数组.....26

2.2.3 委托.....27

2.2.4 接口.....29

2.3 类型转换.....31

2.3.1 隐式类型转换.....31

2.3.2 显式类型转换.....33

2.3.3 Convert 类.....35

2.4 装箱和拆箱.....35



2.4.1 内存“堆栈”和“堆”	35	3.3.1 break 语句	59
2.4.2 装箱转换	36	3.3.2 continue 语句	59
2.4.3 拆箱	36	3.3.3 return 语句	59
2.5 变量和常量	37	3.3.4 goto 语句	59
2.5.1 变量声明	37	3.4 异常处理	60
2.5.2 变量类型	37	3.4.1 异常处理概述	60
2.5.3 常量	39	3.4.2 异常处理语句 try-catch 语句	61
2.5.4 标识符命名规则	39	3.4.3 使用 finally 块	62
2.6 表达式	40	3.4.4 未处理的异常	62
2.6.1 操作符	40	3.4.5 异常层次结构	62
2.6.2 赋值运算符及其表达式	41	3.4.6 throw 语句	63
2.6.3 算术运算符及其表达式	41	3.4.7 通过实例说明程序处理异常的 流程	64
2.6.4 关系运算符及其表达式	42	3.5 作业	66
2.6.5 布尔运算符及其表达式	42	第 4 章 数组和集合	67
2.6.6 位运算符及其表达式	43	4.1 集合类型与集合类型的接口	67
2.6.7 条件运算符及其表达式	43	4.1.1 集合类型概述	67
2.6.8 自增、自减运算符	43	4.1.2 IEnumerable 接口	68
2.6.9 new 运算符	44	4.1.3 ICollection 接口	69
2.6.10 is 运算符	44	4.2 数组 Array、ArrayList 和 List 泛 型类	70
2.6.11 as 运算符	44	4.2.1 IList 接口	70
2.6.12 typeof 运算符	44	4.2.2 数组 Array 类	70
2.6.13 checked 和 unchecked 操作符	44	4.2.3 ArrayList 集合类型	72
2.6.14 sizeof 运算符	45	4.2.4 List 泛型类	74
2.7 String、StringBuilder 和 DateTime		4.2.5 数组 Array、ArrayList 和 List 泛型类比较	76
数据类型	45	4.3 Hashtable、Dictionary 和 SortedList 集合类	76
2.7.1 字符串 string	45	4.3.1 IDictionary 接口	76
2.7.2 StringBuilder 类	48	4.3.2 Hashtable 类	77
2.7.3 DateTime 结构	49	4.3.3 Dictionary 泛型类	78
2.8 作业	50	4.3.4 SortedList 类型	80
第 3 章 流程控制	51	4.3.5 SortedList 泛型类	81
3.1 选择语句	51	4.4 Queue 类和 Queue 泛型类, Stack 类和 Stack 泛型类	83
3.1.1 if 语句	51	4.4.1 队列 Queue 类和 Queue 泛型类	83
3.1.2 switch 语句	53	4.4.2 堆栈 Stack 类和 Stack 泛型类	85
3.2 循环语句	54	4.5 选择集合类	86
3.2.1 while 循环语句	54		
3.2.2 do-while 循环语句	55		
3.2.3 for 循环语句	56		
3.2.4 foreach 语句	58		
3.3 跳转语句	58		

4.6 作业	87	5.3.15 ListView 控件	106
第 5 章 创建 Windows 窗体应用		5.3.16 ProgressBar 进程条控件	107
程序	88	5.3.17 ToolStrip 控件	107
5.1 Windows 窗体	88	5.3.18 StatusStrip 状态栏控件	108
5.1.1 Windows 窗体简介	88	5.3.19 MenuStrip 菜单控件	108
5.1.2 窗体应用程序的创建与运行	88	5.3.20 Timer 定时器控件	108
5.1.3 Windows 窗体事件	90	5.3.21 控件命名规则	108
5.1.4 Windows 应用程序的委托与事件 处理	90	5.4 通用对话框	109
5.1.5 向 Windows 窗体添加控件	91	5.4.1 MessageBox 消息框	109
5.2 调整控件	92	5.4.2 文件对话框	110
5.2.1 调整控件的大小	92	5.4.3 例：创建标准菜单、工具栏和 状态栏	110
5.2.2 在窗体中定位控件	92	5.5 作业	113
5.2.3 对控件进行分层	93	第 6 章 面向对象程序设计	115
5.2.4 使用 Anchor 属性固定控件相对于 父控件的边框距离	93	6.1 对象、类和结构	115
5.2.5 Dock 属性将控件与窗体边缘 对齐	93	6.1.1 对象、类和结构概述	115
5.2.6 设置控件的 Tab 键顺序	93	6.1.2 类	115
5.2.7 使用“格式”子菜单调整界面 控件布局	94	6.1.3 结构	115
5.3 控件介绍	94	6.1.4 对象	116
5.3.1 TextBox 控件	96	6.1.5 类声明	116
5.3.2 Button 按钮控件	96	6.1.6 分部类定义	117
5.3.3 例：登录对话框和模式对话框的 应用	96	6.1.7 类继承定义	117
5.3.4 RadioButton 单选按钮控件	99	6.1.8 字段	118
5.3.5 CheckBox 复选框控件	99	6.1.9 构造函数	118
5.3.6 例：使用单选和复选按钮编写 Windows 应用程序	99	6.1.10 析构函数	119
5.3.7 ListBox 控件	101	6.2 方法	120
5.3.8 ComboBox 控件	102	6.2.1 方法声明	120
5.3.9 例：使用 ListBox 控件和 ComboBox 控件	102	6.2.2 方法的参数	121
5.3.10 GroupBox 控件	104	6.2.3 通过值传递值类型	121
5.3.11 Panel 控件	104	6.2.4 通过值传递引用类型	122
5.3.12 SplitContainer 控件	104	6.2.5 引用传递值类型	123
5.3.13 TabControl 选项卡控件	104	6.2.6 通过引用传递引用类型	123
5.3.14 Treeview 控件	105	6.2.7 参数数组 params	124
		6.3 静态成员与静态类	125
		6.3.1 静态成员	125
		6.3.2 静态类	126
		6.4 继承	126
		6.4.1 继承多态性	126
		6.4.2 使用 new 关键字	127



6.4.3	使用 virtual 和 override 关键字	128
6.4.4	abstract 抽象类和类成员	128
6.4.5	sealed 密封类和类成员	129
6.4.6	接口继承	130
6.4.7	重载运算符	131
6.5	属性与索引器	133
6.5.1	属性的定义	133
6.5.2	使用属性	134
6.5.3	接口属性	134
6.5.4	索引器	135
6.5.5	接口中的索引器	137
6.5.6	属性和索引器之间的比较	137
6.5.7	非对称访问器可访问性	137
6.5.8	.Net 框架类的索引器与属性 Item	138
6.6	使用 Visual Studio 类关系图可视化 创建对象	138
6.7	作业	144
第 7 章 委托与事件、泛型和反射 146		
7.1	委托与事件	146
7.1.1	委托	146
7.1.2	事件概述	148
7.1.3	使用事件	148
7.1.4	.NET 框架事件的委托类型	150
7.1.5	Windows 应用程序使用事件 处理机制	151
7.2	泛型	151
7.2.1	泛型概述	151
7.2.2	泛型类、结构、接口和泛型 成员	152
7.2.3	类型参数的约束	153
7.3	可空类型	155
7.3.1	可空类型概述	155
7.3.2	使用可空类型	155
7.4	反射	157
7.4.1	反射概述	157
7.4.2	反射获取和加载程序集 System.Reflection.Assembly	157
7.4.3	反射获取类型 System.Type	157

7.4.4	反射创建类型的成员	158
7.4.5	反射创建类型的实例	158
7.4.6	反射调用类型实例的方法	159
7.4.7	反射的性能	159
7.4.8	反射技术示例	160
7.5	作业	161
第 8 章 流、文件、XML 和配置		
文件 162		
8.1	流和文件	162
8.1.1	流 Stream 类	162
8.1.2	FileStream 类	162
8.1.3	BinaryReader 与 BinaryWriter 类	164
8.1.4	TextReader、StreamReader 和 StringReader 类	167
8.1.5	File 类	168
8.1.6	FileInfo 类	169
8.1.7	Directory 类	170
8.1.8	例：创建简单 Windows 资源管 理器应用程序	172
8.2	.NET 框架的 XML 编程	175
8.2.1	.NET 框架的 XML 编程概述	175
8.2.2	XML 文档对象模型	176
8.2.3	使用 XmlReader、XmlWriter 读写 XML 文件	177
8.2.4	使用 XmlDocument 类处理 XML 数据	180
8.3	应用程序配置文件	181
8.3.1	应用程序设置 Settings	181
8.3.2	使用 ConfigurationManager 管理 应用程序配置文件	183
8.4	作业	184
第 9 章 数据库编程 186		
9.1	ADO.NET 概述	186
9.1.1	设计 ADO.NET 的目的	186
9.1.2	ADO.NET 结构	186
9.1.3	.NET 框架数据提供程序	187
9.1.4	.Net 框架数据提供程序的 比较	188

9.2 使用 .NET 框架数据提供程序.....190	9.6.4 DataGridView 控件.....226
9.2.1 使用 Connection 类建立连接.....190	9.6.5 通过 ComboBox 和 DataGridView 控件创建查找表.....228
9.2.2 使用命令 Command 类.....191	9.7 作业.....230
9.2.3 DBDataReader 类.....193	第 10 章 进程与线程.....232
9.2.4 使用 DBCommand 执行存储 过程.....193	10.1 进程.....232
9.2.5 面向接口 IDbConnection、 IdbCommand、IDataReader 编程...195	10.1.1 创建和绑定到进程.....232
9.3 DataSet 数据集.....196	10.1.2 启动进程和停止进程.....233
9.3.1 DataSet 对象模型.....196	10.1.3 例：创建 Windows 的当前进 程列表应用程序.....233
9.3.2 可视化创建 DataSet 数据集 控件.....197	10.2 线程.....235
9.3.3 使用编程代码创建 DataSet 类.....199	10.2.1 线程概述.....235
9.3.4 DataTable 的架构.....200	10.2.2 创建线程.....236
9.3.5 DataColumn 列对象.....200	10.2.3 线程睡眠 Sleep、中断 Interrupt 和销毁 Abort.....237
9.3.6 创建 DataTable 架构.....201	10.2.4 使用线程池 ThreadPool 类.....238
9.3.7 创建 DataSet 表间关系 DataRelation 对象.....202	10.2.5 后台线程与前台线程.....239
9.3.8 DataRow 类.....202	10.3 线程同步.....239
9.3.9 DataTable 数据添加.....203	10.3.1 Interlocked 类互锁操作.....239
9.4 数据适配器.....203	10.3.2 lock 和 Monitor 类监视器.....242
9.4.1 数据适配器概述.....203	10.3.3 同步事件和等待句柄.....244
9.4.2 从 DataAdapter 填充 DataSet.....204	10.3.4 ReaderWriterLock 类.....246
9.4.3 使用 DataAdapter 更新数据源.....205	10.3.5 Mutex 类.....246
9.4.4 使用 CommandBuilder 类.....207	10.4 BackgroundWorker 与窗体控件线程 安全调用.....246
9.5 类型化数据集与 TableAdapter.....207	10.4.1 BackgroundWorker 组件.....246
9.5.1 数据源创建.....207	10.4.2 实现后台操作的 Windows 应用 程序.....246
9.5.2 类型化数据集.....208	10.4.3 对 Windows 窗体控件进行线程 安全调用.....250
9.5.3 使用类型化数据集创建显示相关 表 Windows 应用程序.....208	10.5 作业.....252
9.5.4 TableAdapter 类.....213	第 11 章 网络编程与 Web 服务.....253
9.5.5 添加 TableAdapter 查询.....215	11.1 网络编程.....253
9.5.6 添加 TableAdapter 全局查询.....219	11.1.1 网络编程概述.....253
9.6 Windows 窗体数据控件及可视化 编程.....221	11.1.2 WebClient 类.....253
9.6.1 BindingSource 组件.....221	11.1.3 使用 WebRequest 和 WebResponse 类请求数据和发送数据.....256
9.6.2 BindingNavigator 控件.....225	11.1.4 使用 FtpWebRequest 和 FtpWebResponse 类.....256
9.6.3 使用数据源设计 Windows 窗体 数据控件.....225	



11.1.5 TCP 服务.....	259	服务.....	266
11.1.6 Socket 类.....	261	11.2.4 使用 SOAP 头身份验证创建	
11.1.7 WebBrowser 控件.....	262	Web 服务安全.....	269
11.2 Web 服务.....	262	11.2.5 使用 Web 服务创建分布式应用	
11.2.1 Web 服务概述.....	262	程序.....	271
11.2.2 使用 ASP.NET 创建 Web		11.3 作业.....	273
服务.....	262	参考文献.....	275
11.2.3 使用事件驱动异步调用 Web			

第 1 章 Visual Studio 2005 和 C#.NET 概述

本章要点:

1. Visual Studio 2005 .NET 框架和 C#语言简介。
2. 使用 VS 2005 创建 C#控制台应用程序, Windows 窗体应用程序。
3. C# 面向对象程序主要结构: 类、对象、语句与方法等简介。
4. 使用 VS 2005 运行程序调试。

1.1 Visual Studio 2005、.NET 框架和 C# 简介

1.1.1 Visual Studio 2005

Visual Studio 是一套完整的开发工具集, 用于生成桌面应用程序、ASP.NET Web 应用程序、移动应用程序和基于微软 Office 的应用程序等多种应用程序, 是 Visual C#、Visual Basic、Visual C++和 Visual J# 等多种语言所共同使用的集成开发环境 (IDE)。利用此集成开发环境可以共享工具且有助于创建混合语言解决方案, 为熟悉不同语言的开发人员开展编程工作提供极大的便利。

Visual Studio 这些语言利用 .NET 框架的功能, 可简化 ASP Web 应用程序和 Web 服务开发的关键技术。可以非常轻松地创建具有自动伸缩能力的、功能强大的、性能可靠的各类应用程序。因此开发人员在开发类型不相同的 .NET 应用程序 (如基于 Windows 的应用程序和基于 Web 的应用程序) 时, 可以使用相似的开发技术。

1.1.2 .NET 框架简介

.NET 框架 (.NET Framework) 是微软支持下一代应用程序的内部 Windows 组件。.NET 框架由三个主要部分组成。

1. 公共语言运行库 (CLR)

公共语言运行库管理内存、线程执行、代码执行、代码安全验证、编译以及其他系统服务。

2. .NET 框架类库

.NET 框架类库开发人员提供了统一的、面向对象的、分层的和可扩展的类库集, 使用它可以方便地开发传统的命令行或图形用户界面 (GUI) 应用程序等多种应用程序。

3. ASP.NET

ASP.NET 建立在 .NET 框架的编程类的基础上, 为 Web 应用程序模型提供了一组可简化 Web 应用程序生成的控件和基础结构 (如 Web 窗体和 Web 服务)。

1.1.3 C# 语言简介

C#从 C 和 C++派生而来, 因为 C#是近期发展起来的, 所以吸取了 C、C++、Java 以及



Delphi 和 Visual Basic 等语言的优点,并解决了它们的问题。具有简单、强大、灵活、类型安全和面向对象等优点。C#是唯一为.NET 框架设计的语言,能使用.NET 框架代码库提供的各种功能。Visual Studio 2005、C# 和.NET 框架具有以下优点。

1. 统一的开发工具和.NET 框架

开发.NET 应用程序主要使用 Visual Studio 开发工具和.NET 框架,这与开发 J2EE 平台的应用程序大不相同,J2EE 平台有来自不同软件提供商的开发工具和不同的 J2EE 应用服务器,需要花时间熟悉和掌握不同的 J2EE 平台开发工具和应用服务器。

2. 开发.NET 技术简单和快速

Microsoft Visual 2005 提供高级代码编辑器、方便的用户界面设计器、集成调试器和许多其他工具,并在 C# 2.0 和.NET 框架的基础上加快各种不同类型功能强大的应用程序的开发。

3. 语言互操作性

由于 C# 编译器生成的 IL 代码符合公共类型规范 (CTS),因此从 C# 生成的 IL 代码可以与从 Visual Basic、Visual C++、Visual J# 的.NET 版本等 CTS 语言中的任何一种生成的代码进行交互。语言互操作性是.NET 框架的一个关键功能。

4. .NET 框架的库

.NET 框架包含一个由 4000 多个类组成的内容详尽的库,这些类被组织为命名空间,为从文件输入和输出到字符串操作、到 Windows 窗体控件的所有内容提供多种有用的功能。典型的 C# 应用程序使用 .NET 框架类库广泛地处理常见的“日常”任务。

1.2 在 Visual Studio 2005 中创建 C# 控制台应用程序

1.2.1 创建 HelloWorld 控制台应用程序项目

下面介绍如何使用 Visual Studio 2005 创建一个简单的 C# 控制台应用程序。程序的功能是在控制台显示“欢迎使用 C# .NET 程序设计”。本书的练习需要在安装了 Visual Studio 2005 的计算机上进行。下面用“|”表示上级菜单与其所属的下级菜单的关系,“|”的左边是上级菜单项,右边是其右边菜单项的子菜单项。

创建 HelloWorld 控制台应用程序步骤如下:

1) 在 Windows 中,选择“开始”|“程序”|“Microsoft Visual Studio 2005”。

2) 单击 Visual Studio 2005 图标,随后会启动 Visual Studio 2005。Visual Studio 2005 的起始页如图 1-1 所示。

3) 选择“文件”|“新建”|“项目”。随后会打开“新建项目”对话框,新建项目对话框如图 1-2 所示。这个对话框允许你使用各种模板来创建一个新项目。

4) 在“项目类型”选项区单击 Visual C# 选择项目的编程语言,在“模板”选项区,单击“控制台应用程序”图标,指定了要创建的应用程序的类型。模板包括 Windows 应用程序、类库以及控制台应用程序等。

5) 名称输入: HelloWorld; 位置输入: D:\C#.net 程序设计\chapter1。解决方案名称自动设为所输入的名称,解决方案名称可以改为其他。确定已勾选了“创建解决方案的目录”选项,然后单击“确定”。随后会打开新项目。

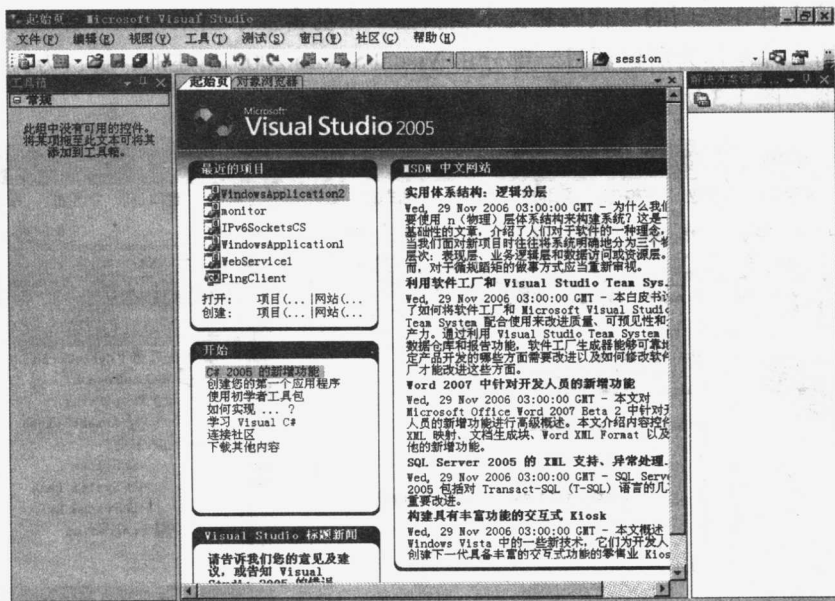


图 1-1 Visual Studio 2005 的起始页

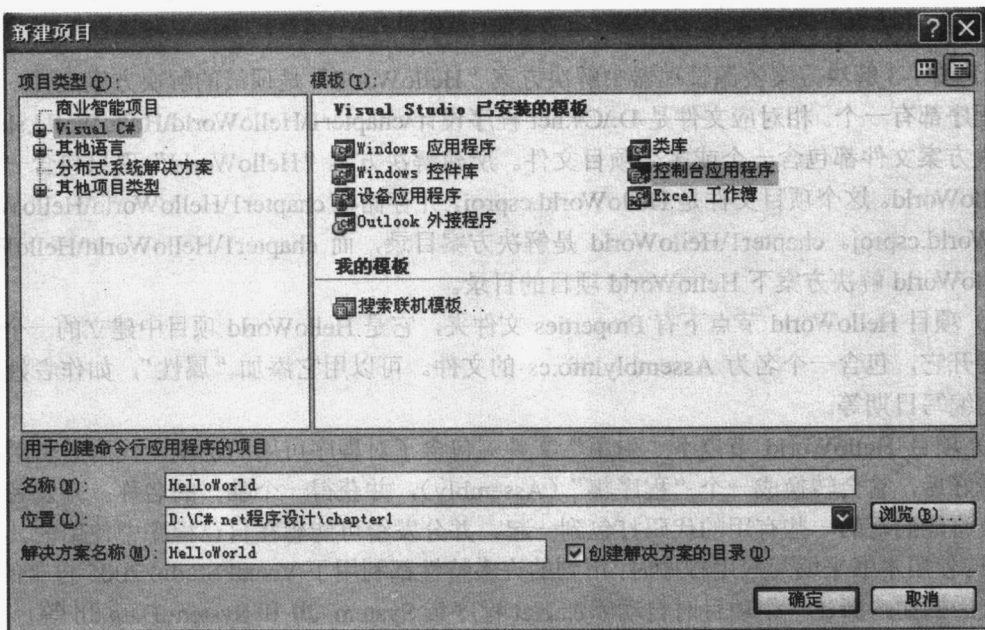


图 1-2 新建项目对话框

6) 通过屏幕顶部的菜单栏和菜单栏下方的工具栏的一系列快捷按钮, 可以访问编程环境所提供的各种功能。占据了 IDE 主要部分的“代码和文本编辑器”窗口显示了源文件的内容。在包含多个文件的项目中, 每个源文件都有它自己的一张“卡片”, 并用源文件的名称加上了标签。单击卡片标签, 即可将相应的源文件带到前台。在右侧的“解决方案资源管理器”中, 显示了项目相关文件以及其他内容的名称。在“解决方案资源管理器”中双击一个文件



名,即可使该文件在左侧的“代码和文本编辑器”中前台显示出来。在“解决方案资源管理器”中列出的文件是作为所建项目一部分由 Visual Studio 2005 创建的。解决方案文件与代码和文本编辑器如图 1-3 所示。

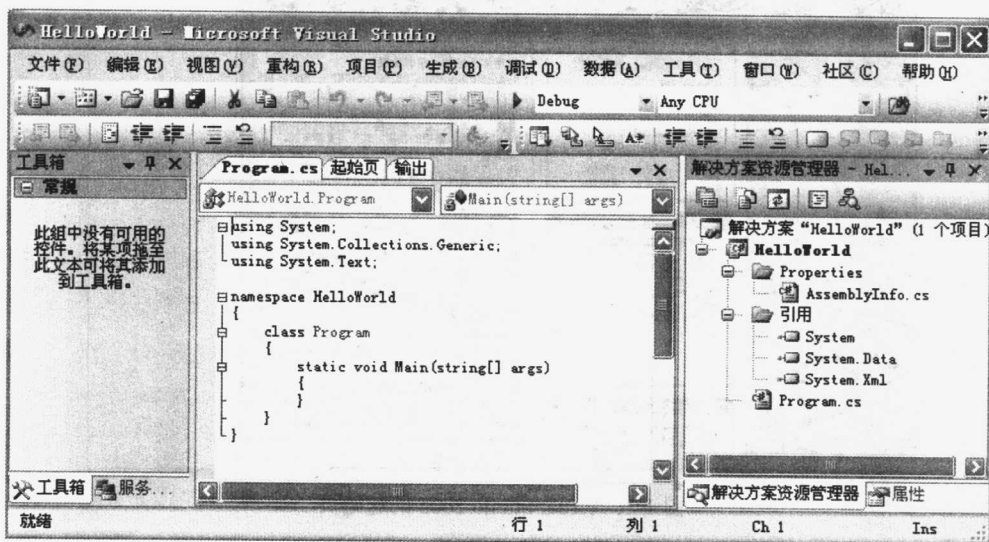


图 1-3 解决方案文件与代码和文本编辑器

7) 图 1-3 解决方案资源管理器中解决方案“HelloWorld”是顶级的解决方案文件,每个应用程序都有一个。相对应文件是 D:\C#.net 程序设计\chapter1\HelloWorld\HelloWorld.sln。每个解决方案文件都包含一个或多个项目文件。所建解决方案“HelloWorld”现只包含一个项目 HelloWorld。这个项目文件是 HelloWorld.csproj,并存储在 chapter1\HelloWorld\HelloWorld\HelloWorld.csproj。chapter1\HelloWorld 是解决方案目录,而 chapter1\HelloWorld\HelloWorld 是 HelloWorld 解决方案下 HelloWorld 项目的目录。

8) 项目 HelloWorld 节点下有 Properties 文件夹,它是 HelloWorld 项目中建立的一个文件夹。展开它,包含一个名为 AssemblyInfo.cs 的文件。可以用它添加“属性”,如作者姓名、程序的编写日期等。

9) 项目 HelloWorld 节点下“引用”文件夹包含了对程序可用的已编译代码的引用。代码编译好后,它会转换成一个“程序集”(Assembly),并获得一个唯一的名称。开发者使用程序集将他们写的一些有用的代码打包到一起,并分发给可能想在自己程序中使用的其他开发人员。使用本书来编写应用程序时,用到的许多特性都利用了 Visual Studio 2005 自带的 .Net 程序集。例如,新建一个项目时自动添加 .Net 程序集 System.dll 和 System.Data.dll 等。

10) 在图 1-3 解决方案资源管理器中双击 Program.cs 文件名,左侧的“代码和文本编辑器”内将显示其内容,Program.cs 是个 C#源代码文件,是项目最初创建时,在“代码和文本编辑器”窗口中显示的文件。它包含 Visual Studio 2005 自动提供的一些代码,可以在这个文件中写入自己的代码来实现程序的功能。

1.2.2 编写在控制台中显示消息“欢迎使用 C#.NET 程序设计”的程序

新建项目时在 Program.cs 文件定义了一个名为 Program 的类(class),在 Program 类中包

含一个名为 **Main** 的方法。C# 是一种对大小写敏感的语言，拼写 **Main** 时，首字母必须大写。**Main** 方法是一个特殊的方法——它指定了程序的入口。它必须是一个静态 (static) 方法 (静态方法将在后面深入讨论)。

在控制台显示消息“欢迎使用 C#.NET 程序设计”的程序编码步骤如下：

1) 在显示了 **Program.cs** 文件的“代码和文本编辑器”窗口中，将光标定位到 **Main** 方法的起始大括号之后，并输入 **Console**。输入单词 **Console** 的首字母 **C** 时，会自动显示一个智能感知 (IntelliSense) 列表。列表中包含所有有效的 C# 关键字，以及在这种情况下有效的数据类型。要么继续输入，要么在列表中滚动，并用鼠标双击 **Console**。除此之外，在输入了 **Console** 之后，智能感知列表会自动定位到 **Console**，可以按 **Tab** 键、回车键或者空格键来直接选择它。

2) 在单词 **Console** 之后，立即输入一个句点。随后，会出现另一个智能感知列表，其中显示了 **Console** 类的方法、属性和字段。

3) 在列表中向下滚动，直到选中 **WriteLine**，再按回车键。除此之外，可以继续输入，在 **WriteLine** 被自动选定后再按回车键。随后，智能感知列表会消失，而 **WriteLine** 方法将添加到源文件中。

4) **Console.WriteLine** 后输入一个起始的圆括号。随后将出现一条智能感知提示。这条提示中显示了 **WriteLine** 方法可用的参数。**WriteLine** 是一个重载的方法，在 **Console** 中，包含了多个名为 **WriteLine** 的方法。可以使用 **WriteLine** 方法的不同版本来输出不同类型的数据 (重载方法将在后面介绍)。单击提示中的上下箭头按钮可以在 **WriteLine** 方法的各个重载的版本中滚动。

5) 再输入一个结束圆括号，再加上一个分号。在左圆括号和右圆括号之间，输入字符串“欢迎使用 C#.NET 程序设计”。现在 **Program.cs** 如下所示。

```
using System;
namespace HelloWorld
{
    class Program
    {
        static void Main (string[] args)
        {
            Console.WriteLine ("欢迎使用 C#.NET 程序设计");
        }
    }
}
```

6) 可以在代码中添加注释。注释是在一些代码行中包含两个正斜杠，后跟一些纯文本。注释会被编译器忽略，但对开发者来说非常有用，因为它们有助于描述一个程序实际采取的操作。例如，下面代码从两个正斜杠开始一直到行末“//该方法是程序入口”都会被编译器忽略。

```
/*该方法是程序入口*/static void Main (string[] args) //该方法是程序入口
```



也可以使用/*来添加多行注释。编译器将跳过它之后的一切内容，直到遇到一个*/（可能出现在许多行之后）。

1.2.3 生成并运行控制台应用程序

完成编写程序后，需要对 C# 代码进行编译，如果编译出错，就要根据 Visual Studio 给出的错误信息，更改直至编译成功。编译可以通过使用生成菜单（F6 键）来完成，或通过调试运行菜单命令来同时完成编译和调试运行，下面是生成和调试运行控制台应用程序的步骤：

1) 在菜单选择“生成”|“生成解决方案”。其结果是对 C# 代码进行编译，生成一个可以运行的程序。在“代码和文本编辑器”窗口下方，会显示一个“输出”窗口。如果“输出”窗口没有出现，请在菜单选择“视图”|“输出”。在输出窗口中，会出现消息，它指出程序已经编译，并显示可能发生的任何错误的细节。就本例来说，没有任何错误或警告，程序成功生成。在“代码和文本编辑器”窗口中，文件名后面的星号(*)表明文件内容已被修改，而且尚未存盘。“生成解决方案”命令会自动保存文件。

2) 选择菜单“调试”|“开始执行(不调试)”。随后会打开一个命令窗口，程序将开始运行，会显示“欢迎使用 C#.NET 程序设计”消息，程序将等待用户按任意键继续。选择显示程序的命令窗口为当前窗口，然后按回车键关闭命令窗口，并返回 Visual Studio 2005 编程环境。如果使用“调试”|“启动调试”命令来运行程序，程序将在运行后立即关闭命令窗口，不会暂停下来等待用户按任意键。

3) 在“解决方案资源管理器”中，单击 HelloWorld 项目（而不是解决方案），再单击工具栏中的“显示所有文件”按钮。随后，会在 C#源文件名上方显示 bin 和 obj 条目。这些条目直接对应于项目文件夹中的 bin 和 obj 文件夹。这些文件夹是在生成应用程序时创建的，其中包含了程序的可执行版本和其他一些文件。单击 bin 左侧的+图标，随后会显示另一个名为 Debug 的文件夹。单击 Debug 左侧的+图标。随后会显示三个子项，分别是 HelloWorld.exe、HelloWorld.pdb 和 HelloWorld.vshost.exe。其中，文件 HelloWorld.exe 是编译好的程序。选择“调试”|“开始执行(不调试)”命令时，就会运行这个程序。另外两个文件包含了在调试模式中运行程序时（选择“调试”|“启动调试”）Visual Studio 2005 要使用的信息。

1.3 命名空间

规模很大的程序需要解决两个问题。首先，代码越多，就越难理解和维护；其次，更多的代码通常意味着更多的名称、更多的已命名数据、更多的已命名方法以及更多的已命名类。随着名称数量越来越多，极有可能出现两个或多个名称发生冲突的情况。

命名空间(Namespace)的宗旨就是解决这个问题，它能为各种标识符（如类名）创建一个已命名空间。同名的两个类如果在不同的命名空间中，相互是不会混淆的。好比名字叫张三的人可以有很多，仅通过“张三”无法确定何人，而当某个地方（例如某单位）只有一个“张三”时，那么“某个地方的张三”则是唯一的。

应该在命名空间中定义所有类型，Visual Studio 2005 环境默认使用项目名称作为顶级命名空间。例如，前例 HelloWorld 项目 Program.cs 中定义了一个命名空间 HelloWorld，并在