



高等职业教育“十二五”规划教材
计算机类核心课程教改项目成果系列教材

C#程序设计案例教程

(第二版)

王明福 主编



科学出版社

免费提供教学资源

高等职业教育“十二五”规划教材
计算机类核心课程教改项目成果系列教材

C#程序设计案例教程

(第二版)

王明福 主编

科学出版社

北 京

内 容 简 介

本书以微软 Visual Studio 2010 作为开发平台, 全书共分 12 章。其中, 前 8 章介绍 C# 语言面向对象程序设计基础和编程环境, 主要内容包括 Visual Studio 2010 开发平台介绍、C# 语言基础、流程控制、数组与结构, 以及类与对象、继承与多态、接口、委托、事件、泛型和文件等; 第 9~12 章介绍 Visual C# 窗体应用开发, 通过开发计算器、记事本和学生信息管理系统等 Windows 应用程序, 详细介绍了包括窗体与常用控件、菜单与工具栏、对话框等在内的界面设计, 以及 Windows 窗体应用程序进阶, 包括文件操作和数据库技术等编程技术。

书中所有程序全部运行通过, 并免费提供所有源程序代码, 以及模仿练习、拓展训练和自我测试练习等源程序代码。

本书可以作为高职高专院校计算机信息管理、软件技术等相关专业“C# 面向对象程序设计”课程的教材, 也可以作为应用型本科相关专业“C# 面向对象程序设计”课程的教材。

图书在版编目 (CIP) 数据

C# 程序设计案例教程/王明福主编. —2 版. —北京: 科学出版社, 2016
(高等职业教育“十二五”规划教材·计算机类核心课程教改项目成果系列教材)

ISBN 978-7-03-047171-0

I. ①C… II. ①王… III. ①C 语言-程序设计-高等职业教育-教材
IV. ①TP312

中国版本图书馆 CIP 数据核字 (2016) 第 019422 号

责任编辑: 孙露露/责任校对: 王万红
责任印制: 吕春珉/封面设计: 耕者设计工作室

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

三河市骏杰印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2009 年 3 月第 一 版 开本: 787×1092 1/16

2016 年 3 月第 二 版 印张: 18

2016 年 3 月第一次印刷 字数: 437 000

定价: 39.00 元

(如有印装质量问题, 我社负责调换〈骏杰〉)

销售部电话 010-62136230 编辑部电话 010-62138978-2010

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

面向对象设计技术已成为当今流行的软件设计技术。C#是在面向对象的大潮流中诞生的宠儿，同时由于它的广泛运用又极大地推动了面向对象技术的发展。

本书以面向对象的基本思想、方法为主要内容，以微软 Visual Studio 2010 作为开发平台，兼顾面向过程与面向对象程序设计的适度分离和高度融合的原则。本书前 4 章是面向过程的 C# 语言基础和语法规则的学习；第 5~8 章是面向对象基本特征和基本技术的学习，主线突出 C# 面向对象的抽象、封装、继承、多态和动态联编五大特征的知识讲授，主要内容包括 C# 语言基础、流程控制、方法、数组与结构，以及类与对象、继承与多态、接口、委托、事件、泛型和文件等；第 9~12 章是 C# 窗体应用程序开发，通过开发简单的计算器程序、记事本和学生信息管理系统，简单介绍了 Windows 窗体程序框架、对话框和常用控件、菜单和工具栏、数据库技术等 Windows 程序的开发方法。

本版是在第一版的基础上，根据大量的教学反馈意见和职业教育的发展需要，吸收行业发展的新知识、新技术和新方法，以适应培养技能型人才的新要求而编写的。结合在国家示范性院校、省示范性专业建设过程中所取得的课程改革成果，从如下三个方面进行了修订。

1. 编写思想和组织形式

本书的编写理念是：以就业为导向、以学生为主体，着眼于学生职业生涯发展，注重职业素养的培养。采用“项目驱动+知识学习+情境应用+自测练习”的四位一体教学模式组织教学内容。前 8 章安排“模仿练习”和“拓展训练”两个层次的实训环节，用于模仿、验证概念、语法规则及其应用，以适应自主学习、合作学习和个性化教学。第 9~12 章选择综合案例“简单计算器、记事本和学生信息管理系统”程序，分解提炼项目的功能模块和程序，按照 C# 面向对象知识结构分配到各个子项目中，伴随系统的设计、开发、优化到最后完善，使学生在项目实施的过程中掌握 C# 面向对象设计技术，在职业情境中实现知识构建。

2. 内容的新增和取舍

为突出“代码重用性”在实际项目开发中的重要意义，新增加了“泛型方法”、“泛型类”和“异常处理”等方面的内容；同时，还将“运算符重载”纳入了本书，以完善方法重载多态性的特征。另外，新增了一章“目录与文件操作（第 8 章）”，本书的 C# 面向对象程序设计知识系统更为完整，以满足开发实际项目的需要。

删除了第一版中的多媒体技术、图形编程、多线程和网络编程等知识的开发案例内容。其目的是突出 C# 面向对象的五大特征，使教材内容与“面向对象程序设计”知识内容归属划分相一致。

3. 遵循“趣味、实用、易学”的特点，更换和新增了部分案例和项目

部分案例、项目来自企业和近几届全国蓝桥杯软件大赛的变形考题，充分反映产业升

级、技术进步和职业岗位变化的要求，从而使本书内容体现新知识、新技术和新方法。

本书可以作为高职高专院校计算机信息管理、软件技术等相关专业“C#面向对象程序设计”课程的教材，也可以作为应用型本科相关专业学习C#面向对象程序设计的教材。

尽管我们在本书的编写方面做了很多努力，但由于作者水平所限，不当之处在所难免，恳请各位读者批评指正，并将意见和建议及时反馈给我们，以便下次修订时改进（联系邮箱：360603935@qq.com）。

前言	
第 1 章 绪论	1
1.1 C#简介	1
1.1.1 Visual C#.NET 简介	1
1.1.2 Visual C#.NET 的特点	2
1.1.3 C#与其他语言的关系	2
1.2 Visual Studio 2010 开发环境	3
1.2.1 起始页窗口	3
1.2.2 菜单栏和工具栏	4
1.2.3 工具箱	5
1.2.4 “类视图”面板	6
1.2.5 代码编辑器/窗体设计器	6
1.2.6 “属性”面板	7
1.2.7 定制开发环境	7
1.3 简单的 C#程序	8
1.3.1 创建 C#程序项目	8
1.3.2 编写 C#程序代码	9
1.3.3 编译、连接与运行	10
1.3.4 C#程序结构分析	10
1.4 基本输入/输出	14
1.4.1 Console.WriteLine()方法	14
1.4.2 Console.Write()方法	17
1.4.3 Console.ReadLine()方法	17
1.4.4 Console.Read()方法	19
自我测试练习	19
第 2 章 数据类型、运算符和表达式	21
2.1 关键字和标识符	21
2.1.1 关键字	21
2.1.2 标识符	22
2.1.3 中文标识符	23
2.1.4 标识符的命名约定	24
2.2 常量与变量	24
2.2.1 常量	24
2.2.2 变量	27
2.3 基本数据类型	28
2.3.1 值类型	28
2.3.2 引用类型	29
2.3.3 类型转换	31
2.4 运算符与表达式	36
2.4.1 算术运算符与算术表达式	36
2.4.2 关系运算符与关系表达式	39
2.4.3 逻辑运算符与逻辑表达式	40
2.4.4 位运算符与位运算	41
2.4.5 条件运算符	45
2.4.6 赋值运算符与赋值表达式	45
2.4.7 运算符的优先级与结合顺序	47
2.5 情境应用——案例拓展	48
2.5.1 案例：解方程	
$ax^2+bx+c=0$ ($a \neq 0$)	48
2.5.2 案例：逻辑推理与判断	49
自我测试练习	50
第 3 章 结构化程序设计	52
3.1 顺序结构	52
3.1.1 简单赋值语句	52
3.1.2 复合赋值语句	52
3.2 选择结构	53
3.2.1 if 语句	53
3.2.2 if 语句的嵌套	56
3.2.3 switch 语句	58
3.3 循环语句	59
3.3.1 while 语句	59
3.3.2 do-while 语句	60
3.3.3 for 语句	61
3.3.4 foreach 语句	63
3.3.5 循环的嵌套	64
3.4 break、continue 和 goto 语句	66
3.4.1 break 语句与 continue 语句	66
3.4.2 goto 语句和标号语句	68

3.5 情境应用——案例拓展	69	5.2.5 方法的重载	109
3.5.1 案例: 爱因斯坦阶梯问题	69	5.2.6 重载构造函数	112
3.5.2 案例: 趣味古典数学问题	70	5.2.7 Main 方法	113
自我测试练习	71	5.3 属性	113
第4章 数组和枚举	74	5.3.1 属性的定义	113
4.1 一维数组	74	5.3.2 属性的读写控制	115
4.1.1 一维数组的定义	74	5.4 静态成员与实例成员	116
4.1.2 一维数组的引用	75	5.4.1 静态数据成员	116
4.1.3 一维数组初始化	76	5.4.2 静态方法	117
4.2 多维数组	78	5.4.3 静态构造函数	118
4.2.1 二维数组的定义	78	5.5 结构	119
4.2.2 二维数组的引用	78	5.5.1 结构的声明	119
4.2.3 二维数组初始化	79	5.5.2 结构成员的访问	120
4.3 不规则数组	82	5.5.3 结构与类的区别	121
4.3.1 不规则数组的定义	82	5.6 索引器	122
4.3.2 不规则数组的引用	82	5.6.1 索引器的定义	123
4.3.3 不规则数组的初始化	83	5.6.2 类中的索引器	123
4.4 综合应用举例	84	5.6.3 结构中的索引器	124
4.4.1 数组与 System.Array	84	5.7 情境应用——案例拓展	125
4.4.2 foreach 语句的应用元素	85	5.7.1 案例: 静态成员的应用	
4.4.3 数组元素的清空	86	实例	125
4.4.4 数组的查找	86	5.7.2 案例: 复杂索引器的应用	
4.4.5 数组的排序	87	实例	127
4.5 枚举	88	自我测试练习	129
4.5.1 枚举类型的定义	88	第6章 面向对象编程进阶	131
4.5.2 枚举变量的定义	89	6.1 类的继承	131
4.5.3 引用枚举	89	6.1.1 继承概述	131
4.6 情境应用——案例拓展	90	6.1.2 继承的实现	132
4.6.1 案例: 冒泡排序	90	6.1.3 构造函数与析构函数	134
4.6.2 案例: 不同进制数的转换	92	6.1.4 成员的继承、添加和隐藏	135
自我测试练习	93	6.1.5 关键字 base 和 this	136
第5章 面向对象编程基础	95	6.1.6 访问控制	138
5.1 类和对象	95	6.2 类的多态	141
5.1.1 类的声明	95	6.2.1 虚方法	141
5.1.2 对象的声明	96	6.2.2 抽象类	142
5.1.3 类成员的访问控制	98	6.3 运算符重载	143
5.2 方法	100	6.3.1 运算符重载的定义	144
5.2.1 方法的声明	100	6.3.2 双目运算符重载为类成员	
5.2.2 方法的参数	101	方法	144
5.2.3 构造函数	106	6.3.3 单目运算符重载为类成员	
5.2.4 析构函数	108	方法	147

6.4 接口	148	8.1.4 目录信息类 DirectoryInfo	191
6.4.1 接口的声明	149	8.2 文件操作	192
6.4.2 接口的实现	150	8.2.1 文件编码	192
6.4.3 接口与多重继承	151	8.2.2 文件流类 FileStream	193
6.5 委托	153	8.2.3 流写入类 StreamWriter	195
6.5.1 委托概述	153	8.2.4 流读取类 StreamReader	197
6.5.2 多路委托	155	8.2.5 二进制流写入类 BinaryWriter	198
6.6 事件	157	8.2.6 二进制流读取类 BinaryReader	199
6.6.1 事件概述	157	8.3 情境应用——案例拓展	200
6.6.2 定义事件	158	8.3.1 案例：文件的加密	200
6.6.3 预定事件	158	8.3.2 案例：C#源文件的编译预 处理	203
6.6.4 引发事件	158	自我测试练习	205
6.7 情境应用——案例拓展	159	第9章 Windows 窗体程序	206
6.7.1 案例：抽象类的应用实例	159	9.1 MyCalculator 程序	206
6.7.2 案例：利用委托进行四则 运算	161	9.2 预备知识	207
自我测试练习	162	9.2.1 创建 Windows 窗体应用 程序	207
第7章 泛型和异常处理	165	9.2.2 认识和使用窗体设计器	208
7.1 泛型简介	165	9.2.3 认识和使用属性面板	209
7.2 泛型方法	166	9.2.4 认识设计器生成的代码	211
7.2.1 泛型方法的定义	166	9.2.5 分部类	212
7.2.2 泛型方法的调用	168	9.3 MyCalculate 程序的开发	213
7.2.3 泛型方法的重载	169	9.3.1 创建程序项目	213
7.3 泛型类	171	9.3.2 界面可视化设计	213
7.3.1 泛型类的定义	171	9.3.3 编写事件代码	215
7.3.2 泛型参数的约束	172	9.3.4 程序运行测试	216
7.3.3 泛型类的重载	174	9.4 窗体与常用控件	217
7.3.4 泛型类的继承	174	9.4.1 Windows 窗体的属性、事件 和方法	217
7.4 异常处理	176	9.4.2 控件中一些常用的属性和 事件	218
7.4.1 异常的概念	176	9.4.3 常用控件简介	218
7.4.2 常见的异常类	176	自我测试练习	220
7.4.3 异常处理	178	第10章 菜单与工具栏设计	221
7.5 情境应用——案例拓展	173	10.1 我的记事本	221
7.5.1 案例：冒泡排序泛型类	173	10.2 创建“我的记事本”程序	222
7.5.2 案例：最大值泛型类	184	10.2.1 创建项目	222
自我测试练习	186	10.2.2 设计窗体	223
第8章 目录与文件操作	188		
8.1 目录和文件管理	188		
8.1.1 文件类 File	188		
8.1.2 目录类 Directory	189		
8.1.3 文件信息类 FileInfo	190		

10.3 菜单设计	224	11.6.1 创建“查找/替换”对话框 ..	249
10.3.1 添加主菜单	224	11.6.2 修改 Form2 和 Form1 类 ..	249
10.3.2 修改主菜单属性	225	11.6.3 实现“查找/替换”功能 ..	250
10.3.3 为菜单项分配快捷键	226	11.7 实现打印功能	251
10.3.4 处理主菜单事件	226	11.7.1 认识 PrintDocument 控件 ..	251
10.4 工具栏设计	227	11.7.2 实现打印功能	253
10.4.1 创建项目资源	228	11.7.3 实现页面设置功能	253
10.4.2 添加工具栏, 导入资源 ..	228	11.7.4 实现打印预览功能	254
10.4.3 设计工具栏	229	11.7.5 运行测试	255
10.4.4 工具栏事件处理	230	自我测试练习	255
10.5 状态栏设计	231	第 12 章 数据库编程	256
10.5.1 状态栏界面设计	231	12.1 学生信息管理系统	256
10.5.2 处理状态栏显示	232	12.1.1 问题描述	256
10.6 剪贴板功能	233	12.1.2 解决方案	257
自我测试练习	234	12.2 ADO.NET 编程基础	257
第 11 章 对话框及其应用	235	12.2.1 ADO.NET 简介	257
11.1 我的记事本(续)	235	12.2.2 基本 SQL 语句	258
11.1.1 问题描述	235	12.2.3 数据库连接	259
11.1.2 解决方案	235	12.2.4 数据库的操作命令	260
11.2 标准对话框	236	12.3 创建数据库	263
11.2.1 标准对话框及使用方法 ..	236	12.3.1 建立 Access 数据库	264
11.2.2 认识保存文件对话框 (SaveFileDialog)	237	12.3.2 创建 SQL Server 数据库 ..	266
11.2.3 认识打开文件对话框 (OpenFileDialog)	239	12.4 创建应用程序、访问数据库 ..	267
11.2.4 认识字体对话框 (FontDialog)	240	12.4.1 数据库应用程序开发步骤 ..	267
11.2.5 颜色对话框 (ColorDialog)	242	12.4.2 创建应用程序项目	268
11.3 自定义对话框	242	12.4.3 连接并访问数据库	269
11.3.1 创建自定义对话框	242	12.5 数据库记录的编辑	270
11.3.2 窗体间的数据交换	243	12.5.1 创建“添加记录”窗体 ..	270
11.4 文件保存和加载	244	12.5.2 设计“添加记录”窗体 ..	271
11.4.1 文件保存	244	12.5.3 修改 RecordAdd 类	271
11.4.2 文件打开	245	12.5.4 实现记录添加功能	273
11.5 实现查找功能	246	12.5.5 程序运行与测试	274
11.5.1 创建“查找”对话框	246	12.6 数据库记录的查询	274
11.5.2 设计“查找”对话框	247	12.6.1 创建“记录查询”窗体 ..	274
11.5.3 修改 MySearch 类	247	12.6.2 修改 RecordINQ 类, 实现 数据交换	275
11.5.4 实现查找功能	248	12.6.3 实现记录查询功能	276
11.6 实现查找/替换功能	249	12.6.4 程序运行与测试	277
		12.7 ListView 控件	278
		自我测试练习	279
		参考文献	280

学习目标

- 了解 C# 程序设计语言发展过程及其特点
- 掌握 C# 程序的基本结构和开发工具
- 掌握 C# 程序的基本输入/输出语句
- 了解本课程的学习目标和技能要求

C#是为 .NET 平台应用开发而设计的一种现代编程语言，它继承了 C++、Java 等语言的优点，因此，问世后很快成为 Windows 应用开发语言中的宠儿。C#语言及其.NET 开发环境，被认为是近年来最重要的程序设计与开发新技术。

本书通过知识讲解、模仿练习、案例拓展和综合案例相结合的方式介绍学习内容，读者学完本书全部内容后，就可以掌握 C#面向对象程序设计的基本技能，并能开发简单的 Windows 应用程序。

1.1 C# 简介

1.1.1 Visual C#.NET 简介

过去的几十年，计算机语言一直在加速发展。流行的说法是，20 世纪 80 年代学 C 语言，90 年代则是属于 C++ 的黄金时代。那么，21 世纪的理想语言是什么呢？

C#是微软公司在 2000 年 7 月发布的一种全新的简单、安全、面向对象的程序设计语言，它充分吸收了过去几十年中计算机科学发展的经验教训，体现了当前最新的程序设计技术的功能和精华，从 C#中可以看到 C++、Visual Basic、Delphi、Java 等语言的优点。而且，C#于 2001 年 12 月获得了国际标准组织 ECMA 批准的 C#标准 ECMA-334，并于 2003 年 4 月被国际标准化组织 ISO 采纳为 ISO/IEC 23270 标准，这一切都为 C#发展推广奠定了坚实的基础。

通常，我们对于 C#和 Visual C#.NET 可以不加区分，但严格地说，两者是有区别的。C#只是一门语言或者说是一个标准，它是专门为微软的 .NET 平台设计的。作为 Visual Studio.NET 套件中的语言之一（该套件还包括 Visual Basic.NET、Visual C++ .NET 和 J#.NET 语言），C#充当了微软推行 .NET 战略的拳头产品。但是，难保今后不会出现其他使用 C#语言的开发工

具（就像有 Visual C++ 和 C++ Builder 一样）。Visual C#.NET 则是指“C#语言+.NET 框架”。由于目前 C# 必须和 .NET 平台一同使用，因此，C# 语言的学习与以往的语言学习有显著不同，对以往的编程工具可以先学习语言语法，再学习编程环境（例如，先学习 C++ 再使用 Visual C++）；而对 C# 的学习一开始就要进入到 Visual Studio.NET 中进行编程学习。

C# 语言是建立在 .NET Framework 环境之上的，.NET Framework 是一个类库。其为 C# 语言开发的应用程序提供了强大的类库支持，但是，它 also 支持 Visual Basic.NET 和 C++ 的托管方式。C# 语言是 .NET Framework 平台首选的开发语言。也可以这样说，C# 语言就是为 .NET Framework 平台而产生的语言。

注意

C# 语言用于生成面向 .NET 环境的代码，但是它不是 .NET 环境的一部分，初学者很容易把 .NET 和 C# 语言混为一谈。C++ 运行在 .NET Framework 之上的方式称为 C++ 的托管方式。由公共语言运行库环境 .NET Framework 执行，而不是直接由操作系统执行的代码就是托管代码。

1.1.2 Visual C#.NET 的特点

Visual C#.NET 以 Visual Studio.NET 为开发工具，包括交互式开发环境、可视化设计器、编译器和调试器。Visual C#.NET 的特征主要体现在以下两个方面。

1. 语言的变化

C# 是在 C、C++ 的基础上改进而来的，作为一种全新的语言，它继承了 C、C++ 的强大功能，同时，吸收了 Visual Basic 语言简单易用的特点。虽然从整体来说，它基本继承了 C 语言的语法风格，但还是有明显的区别和改进，具体的语言变化细节将在本书相关地方体现。

2. .NET 框架支持

Visual C#.NET 完全集成了 .NET 框架。.NET 框架封装了传统的 Windows API，为用户提供了全新的编程接口，并吸收了微软 20 世纪 90 年代中后期发展的各种新技术（COM+ 组件、ASP 技术、XML 支持等），为程序提供了对语言互操作性、垃圾回收、增强的安全性和改进支持。

其中，.NET 框架的支持是 C# 运行和功能实现的基础，离开了 .NET 框架，C# 就是无源之水、无本之木。因此，对于 C# 的学习，大部分精力要放在对 .NET 框架的熟练掌握和应用方面。

1.1.3 C# 与其他语言的关系

1. 与 C、C++ 的关系

就像名字上的相似性一样，C# 是从 C、C++ 语言演变改进而来的，存在着血缘关系。C# 基本上继承了 C 语言的语法风格，同时，又从 C++ 那里继承了面向对象特性。但是，不能够简单地将 C# 看成 C++ 在 .NET 框架上的翻版。毕竟，它们之间的不同点也是很明显的，

主要体现在以下三点。

第一，C#的对象模型已经面向 Internet 进行了重新设计，使用的是.NET 框架的类库，与 C++的对象模型结构完全不一样。因此，在编程中没有太大的相识感。

第二，C#语言不再提供对指针类型的支持，使得程序不能随便访问内存地址空间，从而使程序更加稳定可靠。用惯了指针的程序员需要一定的适应过程。

第三，在面向对象技术方面，C#不再支持多重继承，避免了以往类层次结构中由于多重继承带来的可怕后果。相应的功能可以通过对接口的多重继承来实现。

.NET 框架的支持，为 C#提供了一个强大的、易用的、逻辑结构一致的程序设计环境。同时，公共语言运行时（Common Language Runtime）为 C#程序提供了一个托管的运行环境，使程序比以往更加稳定、安全。C#被设计用来满足新的在线环境功能的开发需要，解决新环境中的程序设计问题，而不是用来替代 C++，两者在未来一段时期将共存。

2. 与 Java 的关系

从整体上来说，C#与 Java 极其相似，甚至超过了与 C、C++的相似程度，不过，两者还是有区别的。例如，Java 通过虚拟机来实现平台的可移植性，而 C#则是首先被编译成一种中间语言（类似 Java 的字节码），然后，在执行时由公共语言运行中的即时编译器编译本机代码交由 CPU 处理。而且，Java 虚拟机只能执行 Java 程序，而即时编译器能够编译任何由.NET 框架支持的语言（如 C#、Visual Basic、J#）编写的程序。

鉴于 C#与 Java 的相似性，学习过 Java 的人员对 C#掌握起来不会感到太难，反过来也一样。由于两者支持平台（或者说运行环境）的不同，相互之间不会取代，而会长期共存。

1.2 Visual Studio 2010 开发环境

Microsoft Visual Studio 2005/2008/2010 等都可以作为 C#程序的开发平台。本书采用 Microsoft Visual Studio 2010 集成开发环境作为 C#程序的开发平台。

注意

本书的所有源程序代码都在 Visual Studio 2010 集成开发环境中测试通过，但也同样适合 Microsoft Visual Studio 2005/2008 等集成开发环境。读者可以自主选择不同版本的开发环境。

1.2.1 起始页窗口

单击“开始→程序→Microsoft Visual Studio 2010→Microsoft Visual Studio 2010”菜单项，启动 Microsoft Visual Studio 2010 开发环境，进入起始页。可以看到，整个起始页窗口主要包含 7 个区域：菜单栏、工具栏、工具箱、解决方案资源管理器、输出窗口、状态栏和中间的起始页，如图 1-1 所示。

下面分别简单介绍其中的几个主要区域，以帮助读者快速了解 Visual Studio 2010 开发

环境的使用。



图 1-1 Visual Studio 2010 起始页窗口

1.2.2 菜单栏和工具栏

1. 菜单栏

菜单栏中包括了 Visual Studio 2010 的大多数功能，菜单项目众多。Visual Studio 2010 的菜单随着不同的项目、不同的文件会进行动态变化。此处就不对所有的菜单进行一一介绍了，而只对常用的“文件”菜单、“编辑”菜单和“视图”菜单进行简单的介绍，以方便读者尽快熟悉 Visual Studio 2010 常用菜单的使用。

“文件”菜单提供了对 Visual Studio 2010 中文件操作的各种功能，其主要菜单项的功能如表 1-1 所示。

表 1-1 “文件”菜单项功能

菜 单 项	功 能
新建	新建项目、网站、文件等
打开	打开项目、网站、文件等
关闭	关闭当前页面
关闭解决方案	关闭当前解决方案
保存选定项	保存当前打开的项目或文件
将选定项另存为	将项目另存为其他项目或文件
全部保存	将所有未保存的文件保存
最近的文件	最近打开的文件
最近使用的项目和解决方案	列表最近打开的项目和解决方案
退出	退出 Visual Studio 2010

“编辑”菜单提供了 Visual Studio 2010 中大多数常见的文本编辑操作，其菜单项的功能

如表 1-2 所示。

表 1-2 “编辑”菜单项功能

菜 单 项	功 能
撤销	撤销上次的操作
重复	重复上次的操作
撤销上次全局操作	撤销上次全局操作
重复上次全局操作	重复上次全局操作
剪切	剪切选中内容到剪贴板
复制	复制选中内容到剪贴板
粘贴	粘贴剪贴板中的内容

“视图”菜单中各菜单项提供的功能比较简单，主要是对各种窗口的显示和隐藏的控制，此处就不一一列出其功能了。感兴趣的读者可以进行尝试，以获得直观的认识。

2. 工具栏

工具栏提供了最常用的功能按钮。对工具栏的熟悉可以大大节省工作时间，提高工作效率。同菜单一样，Visual Studio 2010 的工具栏也是动态变化的。随着工作的不同，工具栏也不尽相同。工具栏的内容还可以根据个人的使用习惯进行自定义，以方便不同开发人员的使用。图 1-2 给出的是位于菜单栏下的第一个工具栏。

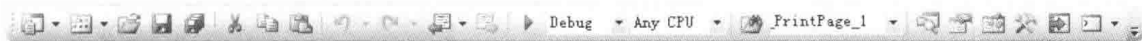


图 1-2 工具栏

这个工具栏提供了本节中提到的“文件”菜单、“编辑”菜单和“视图”菜单中的部分功能。另外，它还提供了部分编译选项功能。具体功能可以参见菜单栏中的介绍，此处就不一一列举了。感兴趣的读者可以将鼠标光标悬停于工具栏中相应的按钮上，观察 Visual Studio 2010 给出的提示，提示中一般会给出简单的帮助，可以协助读者快速熟悉相应功能。

1.2.3 工具箱

工具箱是 Visual Studio 2010 的重要工具，它提供了进行 Windows 窗体应用程序开发所必需的控件。通过工具箱，开发人员可以方便地进行可视化窗体设计，从而简化了程序设计，提高了工作效率。

图 1-3 所示的是工作箱的外观。如果看不到工具箱，可以从“视图”菜单中选择“工具箱”菜单项。展开工具箱中的“所有 Windows 窗体”列表，可以看到如图 1-4 所示的效果。

由于“所有 Windows 窗体”列表中的内容太多，故此处仅展示了部分内容。在图 1-4 中可以发现几乎任何程序中都会用到的 Button（按钮），还有常用的 CheckBox（复选框）等许多控件。在工具箱的其他列表中还有许多其他控件，读者可以自行查看。工具箱的使用将在以后的学习中逐渐地向读者介绍，此处只需要有一个直观的认识。

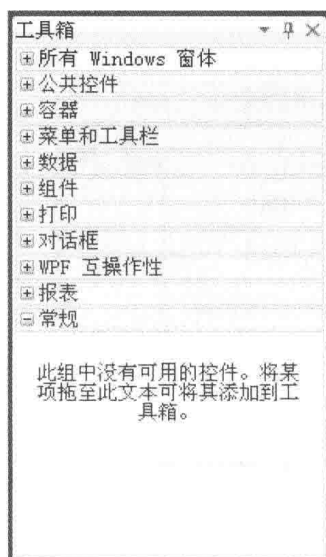


图 1-3 工具箱图示 1

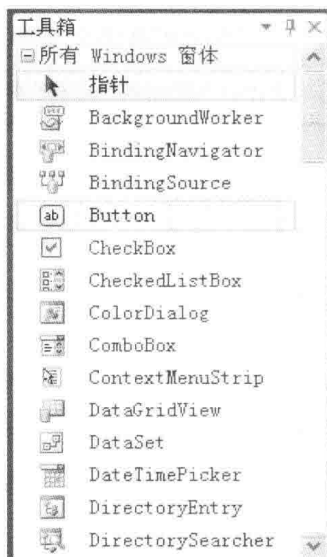


图 1-4 工具箱图示 2

1.2.4 “类视图”面板

“类视图”面板是一个非常方便的工具，该面板提供了观察类结构的非常直观的功能。通过“类视图”面板可以对类的内部构造进行方便的查看。图 1-5 即为使用“类视图”面板查看类结构的示意图。

读者还可以对“类视图”面板进行自定义设置。设置方法为右击“类视图”面板，之后根据提示进行相应的操作。由于设置方法比较简单，此处就不进行介绍了。

1.2.5 代码编辑器/窗体设计器

代码编辑器是 Visual Studio 中开发人员需要面对和消耗时间最多的一个工具。该工具提供了强大的代码编辑功能，当打开或新建一个项目时，就会在“起始页”的位置打开代码编辑器和窗体设计器。如图 1-6 所示，窗体设计器用来设计程序的用户界面，初始显示

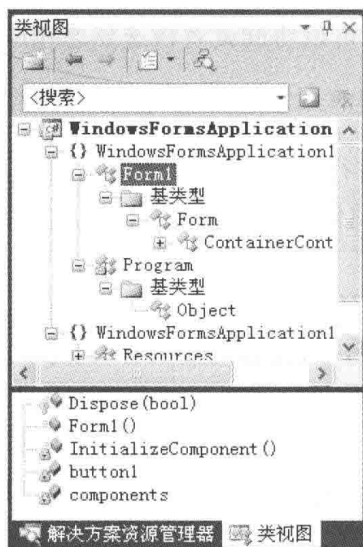


图 1-5 “类视图”面板

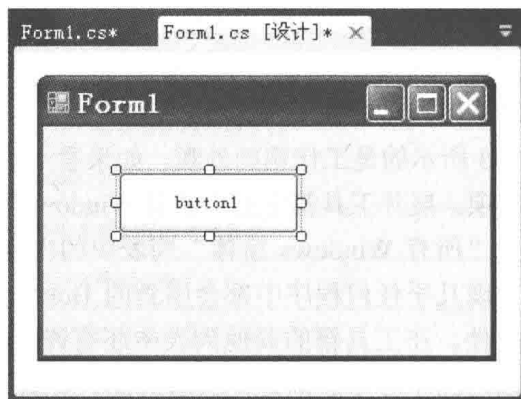


图 1-6 代码编辑器/窗体设计器

一个空的窗体代码编辑器用来编辑项目的源代码。可以通过选择“视图”菜单中的“代码”和“设计器”菜单项在代码编辑器和窗体设计器之间切换，也可以单击解决方案资源管理器窗口上部的“查看代码”和“视图设计器”按钮进行切换。

1.2.6 “属性”面板

“属性”面板是 Visual Studio 2010 中另一个重要的工具。该面板为 Windows 窗体应用程序的开发提供了简单的属性修改方式。对窗体应用程序开发中的各个控件的属性修改都可以由“属性”面板来完成。“属性”面板不仅提供了属性的修改功能，还提供了事件的管理功能。“属性”面板可以管理控件的事件，方便编程时对事件的处理。

“属性”面板同时采用了两种方式管理属性和方法，即按分类方式和按字母顺序方式，读者可以根据自己的习惯采取不同的方式。面板的下方还有简单的帮助内容，方便开发人员对控件的属性和方法进行的操作和修改。图 1-7 是按分类方式列出窗体属性的“属性”面板。

在“布局”列表中可以看到窗体的位置 (Location)、大小 (Size) 等属性，通过“属性”面板可以对其进行方便的设置。

1.2.7 定制开发环境

除了前面介绍的 Visual Studio 默认窗口布局，用户还可以根据自己的需要自由地定制环境布局。最常用的方法是选择“工具”菜单中的“自定义”和“选项”两个命令。

1) 选择“自定义”命令时，会打开“自定义”对话框，如图 1-8 所示。其中有两个选项卡：“工具栏”和“命令”。它们的使用与 Word 中类似，这里就不介绍了。



图 1-7 “属性”面板

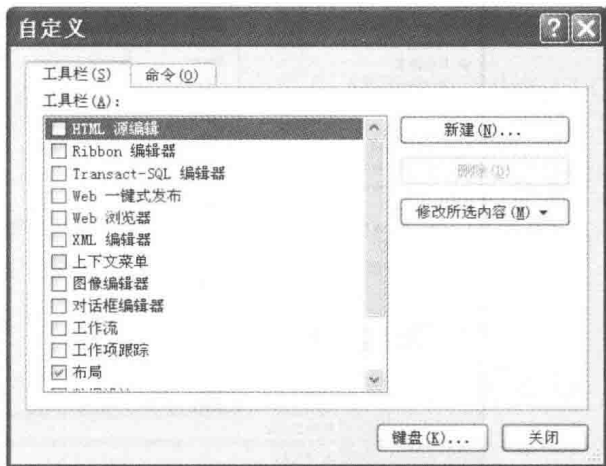


图 1-8 “自定义”对话框

2) 选择“选项”命令时，会打开“选项”对话框，如图 1-9 所示，这里是设置继承开发环境的重要场所。

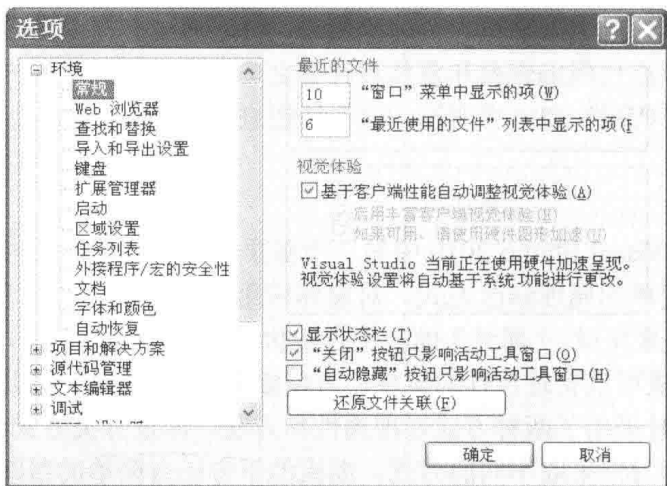


图 1-9 “选项”对话框

1.3 简单的 C#程序

下面来编写本书中的第一个 C#程序，了解 C#程序的结构和上机开发步骤。

1.3.1 创建 C#程序项目

下面将使用 Visual Studio 2010 提供的项目模板来创建一个控制台应用程序 (Console Application)。这个程序将在窗口中显示“欢迎使用 C#”字符串。操作步骤如下。

1) 选择“文件→新建→项目”命令，打开“新建项目”对话框，如图 1-10 所示。

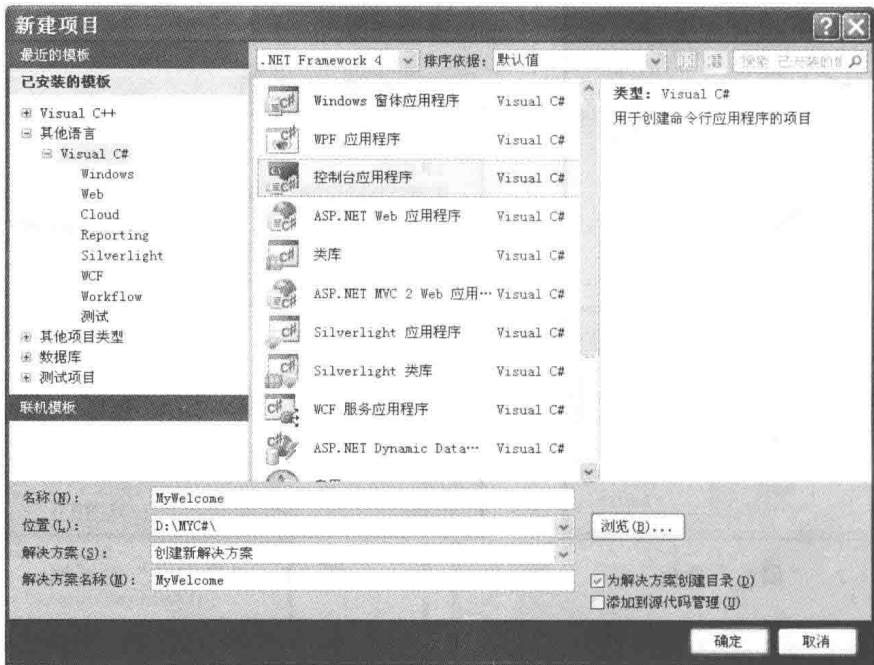


图 1-10 “新建项目”对话框