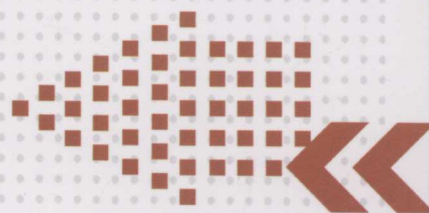
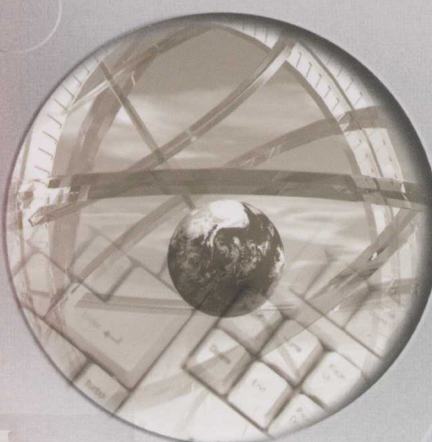




普通高等教育“十二五”计算机类规划教材

C#程序设计基础

◎ 伍星 熊壮 编著



 机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”计算机类规划教材

C# 程序设计基础

伍 星 熊 壮 编著

机 械 工 业 出 版 社

本书针对初次学习程序设计语言的读者,以 C# 语言为载体,以微软 Visual Studio 2010 为开发环境,通过讨论 C# 程序设计的一般过程和方法,重点介绍程序设计的基本思想和实现方法。本书通过数据组织、控制结构、文件处理等程序设计基础知识的讨论,向读者介绍使用 C# 语言进行程序设计的基本方法,通过类的设计、对象定义、重载、派生等知识的讨论向读者介绍面向对象程序设计的基本思想,通过对 GUI 程序设计、图形和图像处理、数据库访问及 Web 程序开发等知识的讨论,使读者能够循序渐进地掌握使用 C# 语言开发各类常见应用程序的基本技能。

本书覆盖了 C# 语言的应用基础,内容深入浅出、语言流畅、例题丰富,可作为非计算机专业第一门程序设计语言课程的教材,也可作为计算机类专业程序设计基础课程的教材,对于程序设计爱好者也是极佳的入门教材或参考书。

为方便教师教学,本书配有免费教学课件,欢迎选用本书作为教材的教师登录 www.cmpedu.com 注册下载或发邮件到 llm7785@sina.com 索取。

图书在版编目 (CIP) 数据

C# 程序设计基础/伍星,熊壮编著. —北京:机械工业出版社,2012.7
普通高等教育“十二五”计算机类规划教材
ISBN 978-7-111-39125-8

I. ①C… II. ①伍… ②熊… 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 152709 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:刘丽敏 责任编辑:刘丽敏 马 超

版式设计:霍永明 责任校对:于新华

封面设计:张 静 责任印制:乔 宇

三河市宏达印刷有限公司印刷

2012 年 8 月第 1 版第 1 次印刷

184mm×260mm·16.25 印张·401 千字

0001—3000 册

标准书号:ISBN 978-7-111-39125-8

定价:32.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

教材网:<http://www.cmpedu.com>

销售一部:(010) 68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010) 88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010) 88379203

封面无防伪标均为盗版

前言

本书编写的主要目的有两个：一是引导没有任何程序设计经历、初学程序设计语言的人进入到程序设计的“广阔天地”；二是从实用的角度出发，通过 C# 语言中关键基础知识的学习和讨论，使读者掌握使用 C# 语言开发实际应用程序的基本技能。

C# 是微软 .NET 平台下的主要程序语言，内容丰富。为了使读者通过本书的学习掌握 C# 语言的关键基础知识，本书将 C# 程序设计基础的内容分为两个相辅相成的阶段。第一阶段主要介绍 C# 应用中最基础的知识，包含的主要内容：控制台应用程序、Windows 窗体程序以及 Web 程序开发的基本步骤和过程，C# 程序中的数据描述和组织、表达式运算和程序中最常使用的数据输入/输出方法，使用分支、循环等流程控制语句描述复杂问题处理过程的方法，使用数组组织相关数据及数组常见的应用，面向对象的基本思想、面向对象方法的定义和调用、数据的作用域规则、类的派生及常用系统定义类的使用方法，Windows 程序设计的基本方法，以及常用控件和组件的使用方法，文件处理的基本步骤和文件中数据的读/写方法。通过第一阶段的学习，读者可以掌握使用 C# 语言开发应用程序最基本的知识和技能。第二阶段讨论最常见的 C# 语言应用问题，包含的主要内容：Graphics 对象概念和绘图基础知识，以及图形、图像的处理基础，关系数据库基本概念、SQL 语句的基本使用规则、数据库数据访问，Web 程序设计的步骤、常用 Web 控件的使用方法及网页之间数据的传递方法。通过第二阶段的学习，读者可以掌握 C# 语言在数据库访问、图形/图像处理以及 Web 应用开发等方面的基本技能，结合相关资料即可以开发相关方面的应用软件。

本书选用 Microsoft Visual Studio 2010 作为教学环境，书中的所有教学示例都在 Visual Studio 2010 集成开发环境中调试通过。基于本书的教学课件、教学示例及习题的参考答案均可以在机械工业出版社网站上下载，也可通过电子邮件向编著者直接索取。

本书由伍星和熊壮共同编著，各章节编写分工如下：伍星编写第 1 章、第 7~10 章，熊壮编写第 2~6 章。全书由两位作者共同协商，进行内容调整、修改，统一定稿。

限于编著者水平，书中错误和不妥之处在所难免，恳请读者不吝指教。

联系地址：重庆大学计算机学院

E-Mail: wuxing@cqu.edu.cn, xiongz@cqu.edu.cn

编者

目 录

前 言	
第1章 C# 及 Visual Studio 2010 开发环境简介	1
1.1 .NET 平台简介	1
1.2 Visual Studio 2010 平台简介	1
1.2.1 Visual Studio 平台的发展过程	1
1.2.2 Visual Studio 2010 中的组件	2
1.2.3 Visual Studio 2010 版本特点	2
1.3 C# 开发应用程序	3
1.3.1 Visual Studio 2010 环境介绍	3
1.3.2 控制台应用程序开发	5
1.3.3 Windows 窗体应用程序开发	8
1.3.4 Web 程序开发	10
习题	14
第2章 C# 程序设计基础	16
2.1 C# 基础语法	16
2.1.1 C# 的字符集	16
2.1.2 C# 的词汇集	16
2.2 C# 的基本数据类型	17
2.2.1 C# 中的常量与变量	18
2.2.2 整型数据类型	18
2.2.3 实型数据类型	19
2.2.4 字符类型	20
2.2.5 布尔类型	22
2.3 基本运算符	22
2.3.1 赋值运算符	23
2.3.2 算术运算符	24
2.3.3 复合赋值运算符	25
2.3.4 自增/自减运算符	26
2.3.5 sizeof 运算符	27
2.4 C# 中的数据类型转换	28
2.4.1 数据的隐式类型转换	28
2.4.2 数据的显式类型转换	29
2.5 数据的基本输入/输出方法	30
2.5.1 控制台应用程序中数据输入/输出方法	30
2.5.2 Windows 窗体应用程序中数据常用的输入/输出方法	33
习题	35
第3章 流程控制	36
3.1 关系运算和逻辑运算	36
3.1.1 关系运算	36
3.1.2 逻辑运算	37
3.2 选择结构	39
3.2.1 if 语句与程序的单分支结构	39
3.2.2 if - else 语句与程序的双分支结构	42
3.2.3 条件运算符与条件表达式	43
3.2.4 if 语句的嵌套与程序的多分支结构	44
3.2.5 switch 语句与程序的多分支结构	47
3.3 循环结构	48
3.3.1 while 型循环结构	49
3.3.2 do - while 型循环结构	50
3.3.3 for 型循环结构	51
3.3.4 循环的嵌套	54
3.4 C# 的其他简单控制结构	55
3.4.1 break 语句	55
3.4.2 continue 语句	56
3.5 C# 控制结构的简单应用	57
3.5.1 穷举思想及其程序实现	57
3.5.2 迭代方法及其程序实现	59
习题	62
第4章 面向对象程序设计基础	66
4.1 类的概念和对象的定义	66
4.1.1 字段	66

4.1.2 方法	67	5.2 数组的常用属性和方法	110
4.1.3 对象的定义和访问	67	5.2.1 数组的 Length 属性和 Rank 属性	110
4.2 方法调用过程中的参数传递	69	5.2.2 数组的 Clone 方法	112
4.2.1 值参数	69	5.2.3 数组的 CopyTo 方法	114
4.2.2 引用参数	70	5.2.4 数组的 Reverse 方法	115
4.3 方法的嵌套调用和递归调用	72	5.2.5 数组的 Sort 方法	118
4.3.1 方法的嵌套调用	73	5.2.6 数组的 BinarySearch 方法	119
4.3.2 方法的递归调用	74	5.3 数组参数和参数数组 (*)	121
4.4 作用域和方法的重载	75	5.3.1 数组参数	121
4.4.1 定义局部作用域	75	5.3.2 参数数组	122
4.4.2 定义类属作用域	76	习题	123
4.4.3 方法的重载	78	第 6 章 Windows 程序设计基础	127
4.5 static 关键字的使用	79	6.1 Windows 系统的消息机制	127
4.5.1 静态字段	79	6.1.1 Windows 系统的工作方式	127
4.5.2 静态方法	80	6.1.2 事件与消息	127
4.6 构造函数	81	6.2 Windows 窗体应用程序常用控件和组件	128
4.6.1 默认构造函数	81	6.2.1 按钮控件	128
4.6.2 构造函数的重载	82	6.2.2 文本控件	129
4.7 继承	84	6.2.3 选择控件和分组控件	134
4.7.1 基类和派生类	84	6.2.4 列表选择控件	136
4.7.2 调用基类构造函数	86	6.2.5 图片列表组件	141
4.7.3 在基类中使用 protected 关键字	86	6.2.6 定时器组件	142
4.8 常用系统定义类	88	6.3 菜单	144
4.8.1 数学类 (System. Math 类)	88	6.3.1 菜单的创建	144
4.8.2 字符串类 (System. String 类)	90	6.3.2 菜单事件处理	147
4.8.3 日期时间类 (System. DateTime 类)	91	6.3.3 快捷菜单	148
4.8.4 随机数类 (System. Random 类)	93	6.4 工具栏和状态栏	148
习题	96	6.4.1 工具栏控件	149
第 5 章 数组	99	6.4.2 状态栏控件	149
5.1 数组的定义和使用	99	6.5 对话框	150
5.1.1 数组的定义和初始化	99	6.5.1 打开文件对话框	150
5.1.2 数组元素值的引用	104	6.5.2 保存文件对话框	151
5.1.3 用 foreach 语句遍历数组	107	6.5.3 颜色对话框	152
5.1.4 数组元素值的随机生成	109	6.5.4 字体对话框	153
		习题	153



第7章 文件读写	156	习题	205
7.1 流文件简介	156	第9章 图形和图像处理	206
7.1.1 流文件概念	156	9.1 GDI+ 概述	206
7.1.2 流类型	156	9.1.1 Graphics 对象	206
7.2 文件操作	157	9.1.2 绘图基础知识	209
7.2.1 字节流的读/写	157	9.2 图形处理基础	213
7.2.2 字符流的读/写	161	9.2.1 图形的绘制	213
7.2.3 二进制流的读/写	164	9.2.2 图形绘制的应用举例	220
7.3 文件管理	166	9.3 图像处理基础	223
7.3.1 File 类	166	9.3.1 图像的存储和显示	224
7.3.2 FileInfo 类	168	9.3.2 彩色图像转换为灰度图像	225
7.4 目录管理	172	9.3.3 灰度图像转换为伪彩色图像	226
7.4.1 目录的创建与删除	173	习题	228
7.4.2 目录中文件和子目录信息的获取	175	第10章 数据库访问基础	230
7.4.3 复制指定目录下的文件	176	10.1 关系数据库	230
7.4.4 删除指定目录下的文件	177	10.1.1 关系数据库的基本概念	230
习题	178	10.1.2 SQL 基础	231
第8章 Web 程序开发基础	180	10.1.3 创建数据库和表	232
8.1 Web 页面结构分析	180	10.2 ADO.NET	234
8.1.1 前台页面文件分析	180	10.2.1 ADO.NET 基础	234
8.1.2 后台代码文件	182	10.2.2 DataReader 对象读取数据	237
8.2 Web 控件	186	10.2.3 DataSet 对象读取数据	244
8.2.1 通用控件	186	10.3 数据绑定	247
8.2.2 网页专用控件	191	10.3.1 数据控件数据绑定	248
8.3 不同页面间的信息传递	199	10.3.2 公共控件数据绑定	250
8.3.1 跨页传递	199	习题	251
8.3.2 查询字符串	202	参考文献	253

第1章 C# 及 Visual Studio 2010 开发环境简介

本章首先介绍 .NET 平台和 Visual Studio 2010 的相关知识, 然后通过 3 个简单的实例, 介绍运用 C# 语言编写控制台应用程序、Windows 窗体应用程序及 Web 程序的整个过程, 初步了解使用微软公司的快速应用开发工具 (Rapid Application Development, RAD) —— Visual Studio 2010 的基本方法。

1.1 .NET 平台简介

.NET 平台是微软公司的最新技术, 其设计目标是帮助软件开发人员轻松并高效地开发各种类型应用程序。本质上, .NET 平台包括一些革命性的新技术和一些对现有技术进行改进的技术, 具体包含如下内容:

- 1) .NET 支持的程序设计语言: .NET 中支持多种语言进行软件开发, 包括 C#、Visual Basic、C++ 等。
- 2) 通用语言运行时 (Common Language Runtime, CLR): 通用语言运行时是所有 .NET 语言的执行引擎, 它为各种应用提供了自动化的服务。
- 3) .NET Framework 类库: 该类库包含了数千个预置功能的类, 这些类可以作为构建各类应用的基石, 在编写程序时可以直接引用这些已经设计好的类。
- 4) ASP.NET: 所有在 .NET 中创建的 Web 应用程序, 都以 ASP.NET 作为引擎或运行平台。ASP.NET 还支持绝大多数 .NET 类库所支持的特性。
- 5) Visual Studio: Visual Studio 是一种可选的软件开发集成工具, 包含了大量工具和特性以提高软件的开发效率和支持软件的调试。Visual Studio 安装光盘中包含了完整的 .NET Framework。

.NET 中语言使用的编译过程与 C、C++ 等语言不同。C、C++ 语言等高级语言基本上都是通过编译器编译成可以执行的机器代码, 而 .NET 中编写好的程序首先编译成通用中间语言 (Intermediate Language, IL) 表示的中间代码, 通用语言运行时中仅支持中间语言 (IL) 代码, 然后进行第二次编译——将中间语言表示的代码转换为当前计算机平台的本地机器语言代码。

1.2 Visual Studio 2010 平台简介

1.2.1 Visual Studio 平台的发展过程

Visual Studio 是目前较流行的 Windows 平台应用程序开发环境, 目前已经开发到 10.0 版本, 也就是 Visual Studio 2010。正在开发的为 11.0 版本, 也就是 Windows 8 的“搭档”(预

览版本 Visual Studio 11)。

Visual Studio 97 是最早的 Visual Studio 版本, 包含有面向 Windows 开发使用的 Visual Basic 5.0、Visual C++ 5.0, 面向 Java 开发的 Visual J++ 和面向数据库开发的 Visual FoxPro, 还包含有创建 DHTML (Dynamic HTML) 所需要的 Visual InterDev。其中, Visual Basic 和 Visual FoxPro 使用单独的开发环境, 其他的开发语言使用统一的开发环境。1998 年, 微软公司发布了 Visual Studio 6.0。所有开发语言的开发环境均升至 6.0 版本。

2002 年, 随着 .NET 概念的提出与 Windows XP/Office XP 的发布, 微软公司发布了 Visual Studio .NET。在该版本中, 引入了建立在 .NET 框架上的托管代码机制以及一门新的语言——C# (读作 C Sharp)。C# 是一门建立在 C++ 和 Java 基础上的现代语言, 是编写 .NET 框架的语言。Visual Basic、Visual C++ 都被扩展为支持托管代码机制的开发环境, 且 Visual Basic .NET 更是从 Visual Basic “脱胎换骨”, 彻底支持面向对象的编程机制。同时, Visual J++ 也变为 Visual J#。

2003 年, 微软公司对 Visual Studio 2002 进行了部分修订, 以 Visual Studio 2003 的名义发布, .NET 框架升级到 1.1 版。

2005 年, 微软公司发布了 Visual Studio 2005。Visual Studio 2005 包含有众多版本, 分别面向不同的开发角色。

2007 年 11 月, 微软公司发布了 Visual Studio 2008 英文版。

2010 年 4 月, 微软公司发布了 Visual Studio 2010 及 .NET Framework 4.0, 并于同年 5 月发布了它们的中文版。

1.2.2 Visual Studio 2010 中的组件

Visual Studio 2010 中包含如下的组件:

- 1) Visual Basic .NET 2010。
- 2) Visual C++ .NET 2010。
- 3) Visual C# .NET 2010。
- 4) Visual F# .NET 2010。

1.2.3 Visual Studio 2010 版本特点

Visual Studio 2010 基于前面几个版本, 在如下几个方面得到了增强:

- 1) 支持 Windows Azure, 微软云计算架构进入重要阶段。
- 2) 助力移动与嵌入式装置开发。
- 3) 实践当前热门的 Agile/Scrum 开发方法, 强化团队竞争力。
- 4) 升级的软件测试功能及工具, 为软件质量严格把关。
- 5) 搭配 Windows 7、Silverlight 4 与 Office, 发挥多核并行运算能力, 创建美感与效能并重的新一代软件。
- 6) 支持最新 C++ 标准, 增强 IDE, 切实提高程序员开发效率。

1.3 C# 开发应用程序

1.3.1 Visual Studio 2010 环境介绍

Visual Studio 2010 安装后首次启动,显示如图 1-1 所示的界面,在该界面中选择所使用的默认程序设计语言,本书用于进行 C# 教学,因此选择“Visual C# 开发设置”,选择后单击“启动 Visual Studio”按钮,出现 Visual Studio 2010 正在加载用户设置的界面,如图 1-2 所示。

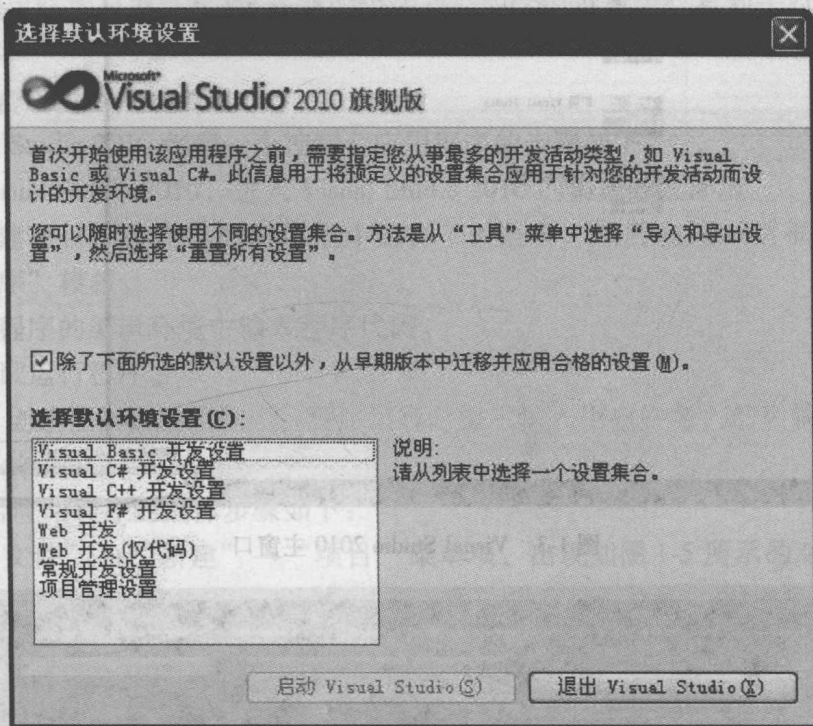


图 1-1 Visual Studio 2010 首次启动界面

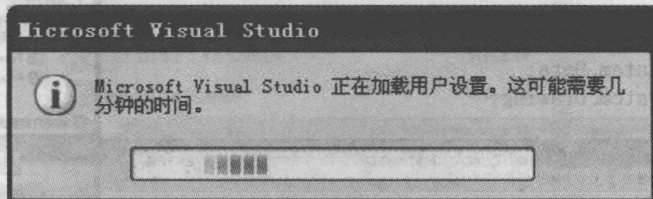


图 1-2 Visual Studio 2010 加载用户设置

Visual Studio 2010 加载用户设置后显示如图 1-3 所示的主窗口,在该窗口的右部可以选择新建项目或打开已经存在的项目;该窗口的中部提供了访问学习资料的方法;窗口的右部是解决方案资源管理器,目前尚未打开和新建项目,因此该部分目前为空。

新建或打开项目后,Visual Studio 2010 会显示如图 1-4 所示的窗口。该窗口中除了包含菜单、图标工具外,还有如下几部分:

1) 工具栏:隐藏在如图 1-4 所示的 Visual Studio 2010 窗口的最左边,其中“工具箱”

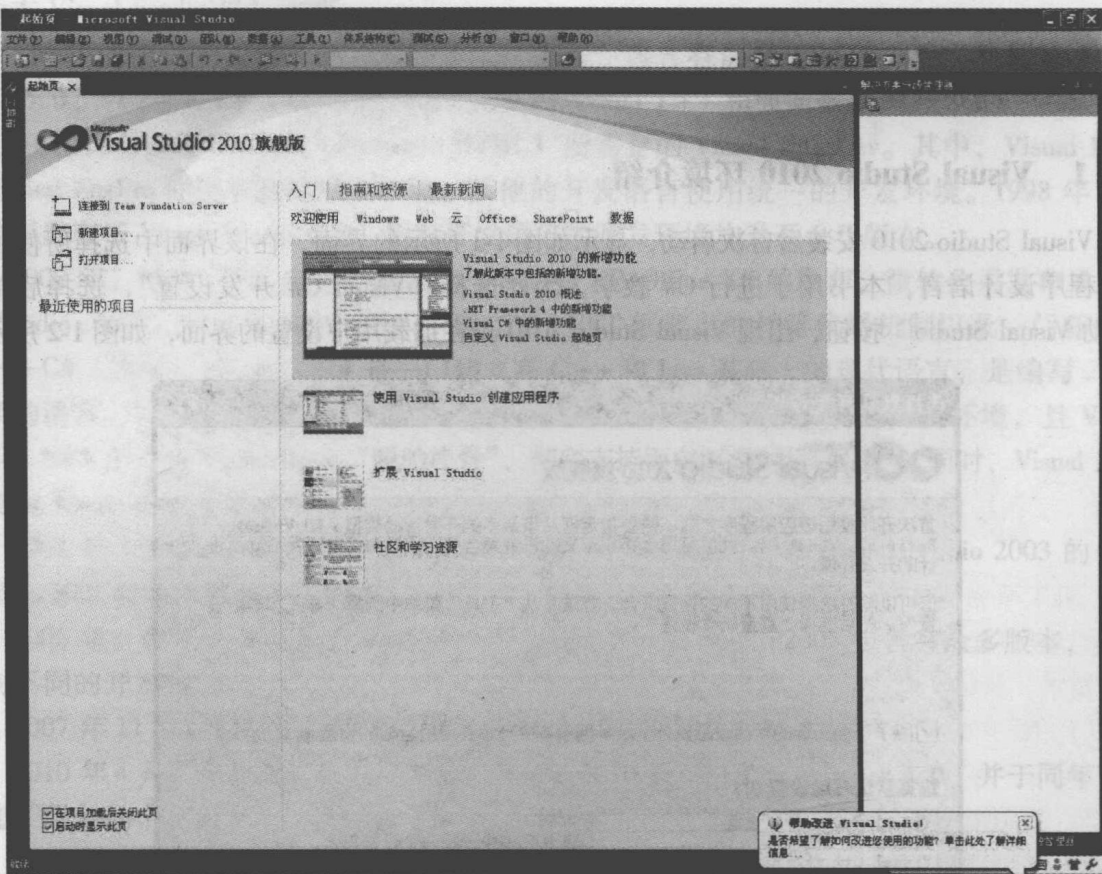


图 1-3 Visual Studio 2010 主窗口

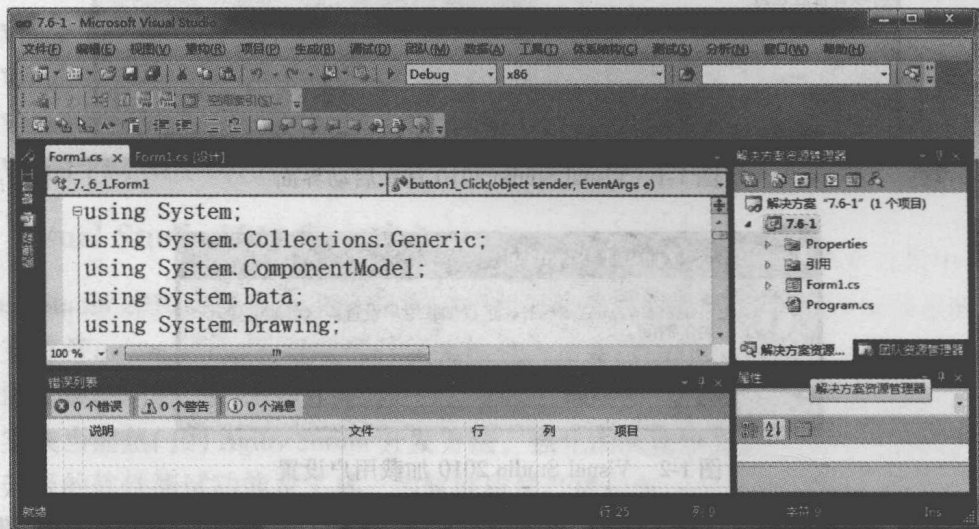


图 1-4 Visual Studio 2010 项目设计窗口

中提供了常用的控件。

2) 工作区：位于如图 1-4 所示的窗口的中间，可以进行代码的编写和窗体的设计，当在该区域有多个源程序文件打开时，采用标签页的方式进行显示。选中上方的标签页即可显示相应的源程序文件或窗体，单击标签页上的“×”，即可在工作区中关闭相应的源程序文

件或窗体。

3) 错误列表：位于如图 1-4 所示的窗口的左下部，作用是在程序编写和调试期间提示程序存在的语法错误和警告信息。

4) 解决方案资源管理器：用于管理当前项目所有的资源，如程序文件等。

5) 属性：用于现实和设置可视化控件的属性，另外还可添加控件的事件响应方法。

1.3.2 控制台应用程序开发

控制台应用程序是 C# 语言开发的应用程序类型之一，不涉及 Windows 操作系统的组成元素。因此，程序结构简单，在 C# 语言的语法部分的学习过程中，有利于对所讨论的问题进行简单的描述。

1. C# 控制台应用程序的创建

使用 Visual Studio 2010 创建一个控制台应用程序的步骤如下：

1) 启动 Visual Studio 2010，进入 Visual Studio 2010 的集成开发环境。

2) 选择新建项目，在弹出的对话框中选择 C# 语言，然后在该对话框的右部，选择“控制台应用程序”模板。

3) 在应用程序的编辑环境中输入程序代码。

4) 依次调试运行程序。

【例 1-1】 创建控制台应用程序，程序运行时提示输入用户姓名，然后输出欢迎用户开始学习 C# 语言的语句。

创建控制台应用程序的具体步骤如下：

1) 选择“文件”→“新建”→“项目”菜单项，出现如图 1-5 所示的对话框。

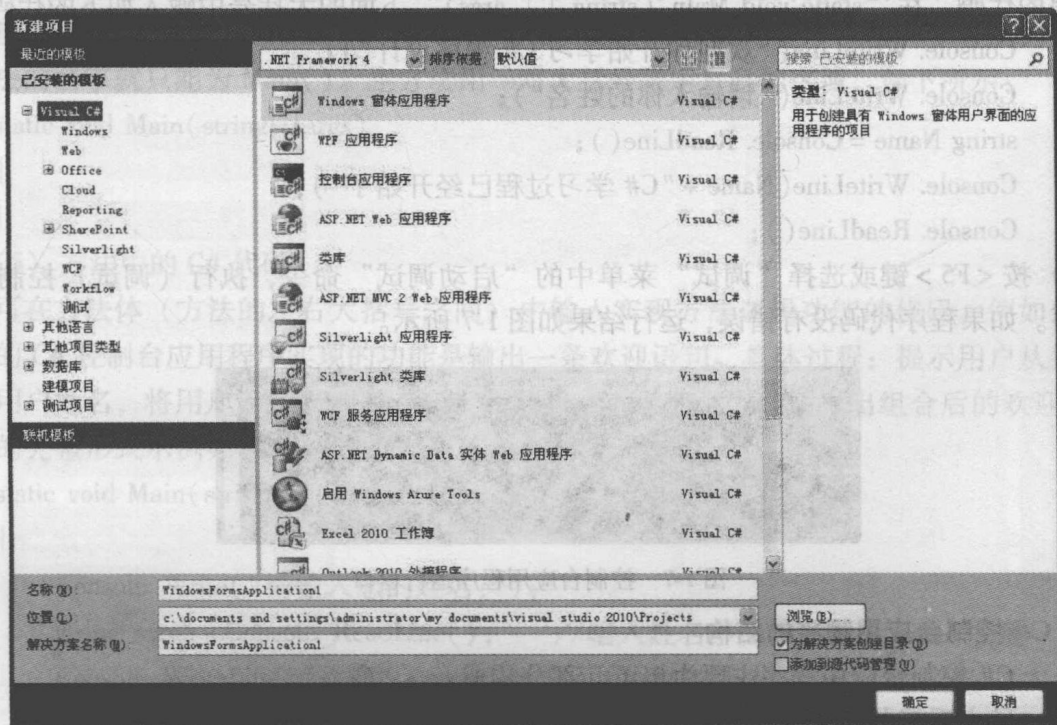


图 1-5 选择新建项目类型

2) 在该对话框左部的“已安装的模板”中选择“Visual C#”，然后在对话框的中间选择“控制台应用程序”。

3) 在该对话框下部的“名称”文本框中输入一个名称作为新建项目的名称，默认情况下 Visual Studio 2010 会提供一个默认名称 (ConsoleApplication1) 作为新建项目的名称。

4) 在该对话框下部的“位置”文本框中指定新建项目的存储位置，也可通过后面的“浏览”按钮指定任意的磁盘位置作为项目的存储位置。

5) 完成上述过程后，单击对话框右下部的“确定”按钮，出现如图 1-6 所示的窗口。

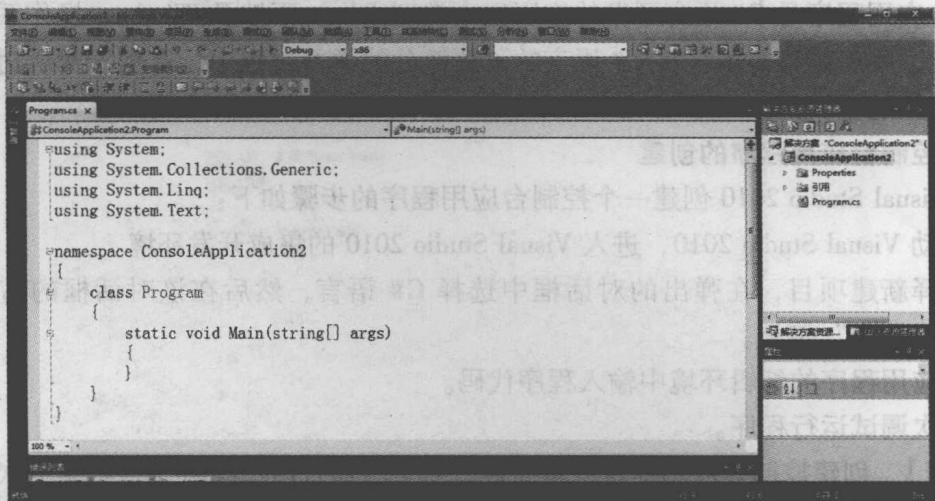


图 1-6 控制台应用程序开发窗口

6) Visual Studio 2010 已经创建了一个名为 Program.cs 的文件，里面已经包含了一些自动创建的代码。在“static void Main (string [] args)”下面的大括号中输入如下的代码：

```
Console.WriteLine("欢迎你开始学习 C# 程序设计");  
Console.WriteLine("请输入你的姓名");  
string Name = Console.ReadLine();  
Console.WriteLine(Name + "C# 学习过程已经开始了");  
Console.ReadLine();
```

7) 按 <F5> 键或选择“调试”菜单中的“启动调试”命令，执行 (调试) 控制台应用程序。如果程序代码没有错误，运行结果如图 1-7 所示。

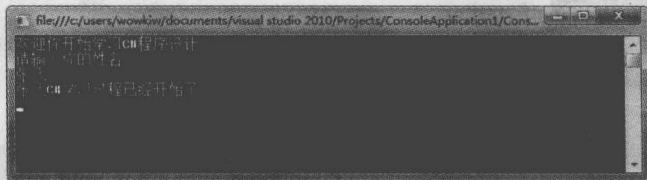


图 1-7 控制台应用程序运行窗口

2. C# 控制台应用程序的结构

一个 C# 控制台应用程序主要由以下几部分构成：

(1) 导入系统预定义元素部分

高级程序设计语言总是依赖许多系统预定义元素，为了在 C# 程序中能够使用这些预定

义元素，需要对这些元素进行导入。在上面创建的控制台应用程序中，使用下述代码段导入了对其他命名空间的引用：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

(2) 命名空间

使用关键字 `namespace` 和命名空间标识符（命名空间名）构建用户命名空间，空间的范围用一对大括号限定，如下所示：

```
namespace Hello //默认情况下命名空间名与解决方案名相同
```

```
{
}
```

(3) 类

类必须包含在某个命名空间中（如 `namespace Hello`），使用关键字 `class` 和类标识符（类名，默认为 `Program`）构建类，类的范围使用一对大括号限定，如下所示：

```
class Program
```

```
{
}
```

(4) 主方法

每个应用程序都有一个执行的入口用以指明程序执行的开始点。C# 应用程序中的入口点用主方法标识，主方法的名字为 `Main()`，后面的括号中即使没有参数也不能省略。一个 C# 应用程序必须有而且只能有一个 `Main()` 方法，如果一个应用程序仅由一个方法构成，这个方法的名字就只能为 `Main()`。主方法用一对大括号限定自己的区域，如下所示：

```
static void Main(string[] args)
```

```
{
}
```

(5) 方法中的 C# 代码

可在方法体（方法的左右大括号之间）中输入实现方法逻辑功能的代码。例如，下面所示的 C# 控制台应用程序实现的功能是输出一条欢迎语句。具体过程：提示用户从控制台输入用户姓名，将用户输入的姓名字符串添加到欢迎语句中，然后输出组合后的欢迎语句。方法的完整形式示例如下：

```
static void Main(string[] args)
```

```
{
```

```
    Console.WriteLine("请输入您的姓名:");
```

```
    string name = Console.ReadLine(); //输入姓名字符串赋值给 name 变量
```

```
    Console.WriteLine("欢迎" + name + "进入 C# 程序设计的广阔天地!");
```

```
    Console.ReadLine(); // * 使得程序执行不会自动退出调试环境 *
```

```
}
```


1.3.3 Windows 窗体应用程序开发

Windows 窗体应用程序通过窗体上的各种 GUI（图形用户界面）元素形成与用户交流的界面。本节介绍 Windows 窗体应用程序的创建过程，以及 Windows 窗体应用程序中最常用控件（窗体、文本框、标签和按钮等）的基本使用方法。

1. Windows 窗体应用程序的创建

使用 Visual Studio 2010 创建一个 Windows 窗体应用程序通常有以下 4 个步骤：

- 1) 设计用户界面。
- 2) 设置对象属性。
- 3) 编写对象事件过程代码。
- 4) 保存并运行程序（生成可执行代码）。

【例 1-2】 创建如图 1-8 所示的 Windows 窗体应用程序。程序运行时可在前面的两个文本框中分别输入数据，单击“相加”按钮计算出两个数的和，并将结果显示到第 3 个文本框中。

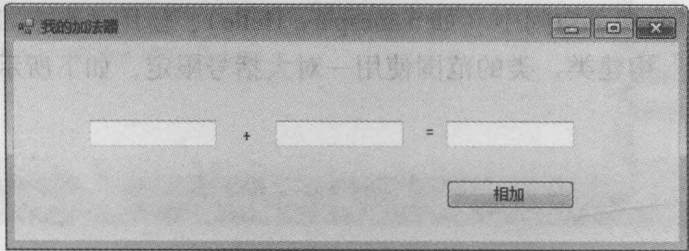


图 1-8 Windows 窗体应用程序示例

(1) 用户界面设计

启动 Visual Studio 2010，新建项目时选择“Windows 窗体应用程序”作为项目类型，确定后出现如图 1-9 所示的界面。

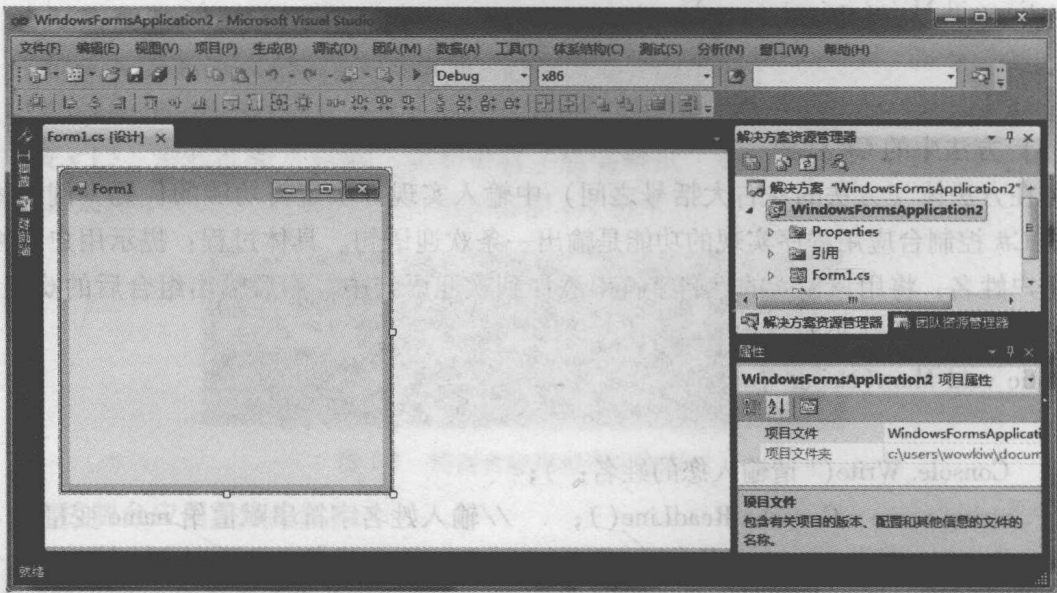


图 1-9 Windows 窗体应用程序开发环境

(2) 添加控件

从工具箱中选择文本框和按钮控件添加到窗体上，具体步骤如下：

- 1) 单击图 1-9 中的工具箱，将隐藏的工具箱显示出来，选择“公共控件”，展开显示相应的控件。
- 2) 双击工具箱中的“文本框”(TextBox) 控件图标，窗体上就会出现一个文本框，该文本框的默认名称为 textBox1，将该文本框拖放到窗体中适当的位置。继续添加两个文本框、两个标签和一个命令按钮到窗体上，添加后如图 1-10 所示。

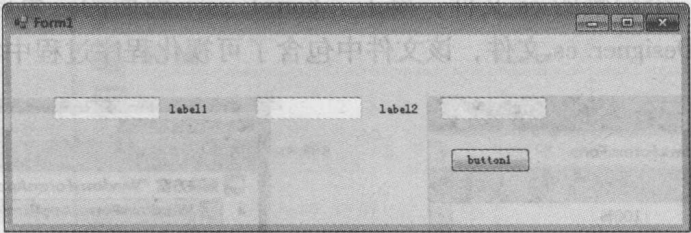


图 1-10 窗体设计界面

- 3) 完成上面两步操作后，然后按照表 1-1 修改相应控件的属性。修改属性的方法：首先选中需要修改属性的控件，然后在 Visual Studio 2010 开发环境的右下角属性界面（见图 1-11）中选择相应的属性，在该属性后面的文本框中进行修改。

表 1-1 窗体中控件的属性设置

控 件	属 性	设 置 值
Form1	Text	我的加法器
label1	Text	+
label2	Text	=
button1	Text	相加

- 4) 双击“相加”按钮，添加按钮响应鼠标单击事件的方法，并在方法中添加如下代码：

```
private void button1_Click( object sender, EventArgs e)
{
    int num1 = int. Parse( textBox1. Text);
    int num2 = int. Parse( textBox2. Text);
    int result = num1 + num2;
    textBox3. Text = result. ToString( );
}
```

- 5) 按 <F5> 键或选择“调试”菜单中的“启动调试”命令，执行（调试）Windows 应用窗体程序，在前两个文本框中分别输入整数，然后单击“相加”按钮，相加的结果显示在第 3 个文本框中。

2. Windows 窗体应用程序结构

Windows 窗体应用程序的总体结构与控制台应用程序的总体结构类似。Windows 窗体应用程序主要由下面几部分组成：

- 1) 导入其他系统预定义元素部分。
- 2) 命名空间。
- 3) 类。
- 4) 方法（主方法、事件响应处理过程）。

从解决方案资源管理器上看，本项目包含两个 cs 文件，如图 1-12 所示，一个是 Form1.cs，也就是前面的设计过程中主要编辑的对象，另一个是 Program.cs，其结构与控制台应用程序非常相似，但它的内容都是自动生成的，不需要程序员干预。其实，解决方案资源管理器中还包含一个隐藏的 cs 文件，单击“Form1.cs”前面的三角形图标，可见项目中还包含一个 Form1.Designer.cs 文件，该文件中包含了可视化程序过程中添加控件的代码。

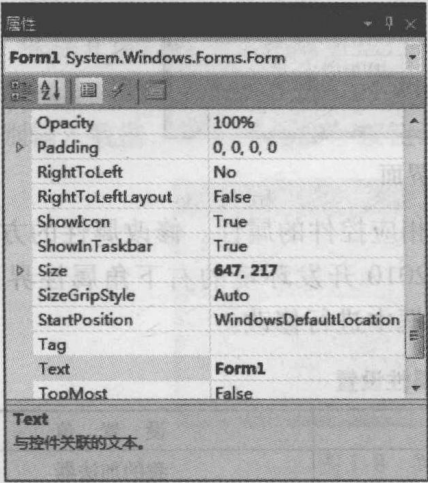


图 1-11 属性界面



图 1-12 解决方案资源管理器

1.3.4 Web 程序开发

在早期的 Web 程序开发过程中，程序员使用记事本之类的简单文本编辑软件创建 Web 页面。由于每一种编辑软件有不同的优缺点，因此在功能上存在较大的局限性。Visual Studio 的出现，尤其是基于 .NET 平台的 Visual Studio 版本的出现，极大地改善了 Web 程序的开发过程。首先，Visual Studio 是可扩展的，可以联合其他的文本编辑器进行工作，其次 Visual Studio 提供了快速开发 Web 程序的功能，可以大大节约开发所需的时间。

Visual Studio 中开发 Web 页面的过程如下：

- 1) 创建新的网站。
- 2) 新建 Web 页面。
- 3) 设计 Web 页面。
- 4) 添加 C# 代码。

【例 1-3】 创建一个 Web 网站，并新建一个 Web 网页，在该网页上提供两个文本框分别用于输入用户名和密码，再添加一个按钮，当单击该按钮后，核对输入文本框中的用户名和密码是否正确并输出相应的提示信息。

(1) 创建 Web 站点

启动 Visual Studio 2010 后选择“新建”，然后选择“网站”，在出现的界面中（见图 1-13）