高等学校计算机基础教育教材精选

C#程序设计实践教程

刘丽 张淑芬 朱俊东 编著

清华大学出版社

高等学校计算机基础教育教材精选

C#程序设计实践教程

刘丽 张淑芬 朱俊东 编著

清华大学出版社
北京

内 容 简 介

本书是《C＃程序设计教程》(ISBN：978-7-302-34927-3)配套实验教程。全书共12章，内容包括Visual Studio 2010集成开发环境的认识、C＃程序设计基础、流程控制与算法、面向对象程序设计、Windows应用程序的开发、文件操作、数据库编程、图形与图像、网络编程等。

本书与教程相互呼应，根据教材内容合理设计实验，全书共32个实验，2个综合实例。每个实验都包括实验目的、实验内容、实践提高等内容，帮助读者扎实地掌握教材对应的内容，一步一步了解并掌握程序设计的思想，进而实现独立编程。其中实践提高部分给出解题思路，具体实现留给读者自行完成，以培养其独立设计程序的能力。综合实例的设置有利于学生形成程序设计的整体观，了解小型软件的开发流程，也可以作为课程设计的参考。

图书在版编目(CIP)数据

C＃程序设计实践教程/刘丽，张淑芬，朱俊东编著. --北京：清华大学出版社，2014
高等学校计算机基础教育教材精选
ISBN 978-7-302-35960-9

Ⅰ. ①C… Ⅱ. ①刘… ②张… ③朱… 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(2014)第066040号

责任编辑：张　玥
封面设计：常雪影
责任校对：焦丽丽
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954

印 刷 者：北京季蜂印刷有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：13.5　　**字　　数**：314千字
版　　次：2014年7月第1版　　**印　　次**：2014年7月第1次印刷
印　　数：1～2000
定　　价：29.00元

产品编号：057177-01

前言

本书是《C#程序设计教程》(ISBN：978-7-302-34927-3)配套实验教程，以 Visual Studio 2010 作为实验开发环境。本书实验内容丰富，涵盖了 Visual C#所有重要内容的实验操作。

本书共 12 章，第 1 章为 Visual Studio 2010 集成开发环境的使用，介绍常用开发环境，为后续章节的学习打下基础。第 2 章为 C#程序设计基础，介绍常量、变量、运算符、表达式等 C#中的基础元素。第 3 章为流程控制与算法，介绍顺序结构、选择结构、循环结构等结构化程序设计的内容。第 4 章为面向对象程序设计基础，介绍类的创建及静态类的应用。第 5 章为面向对象的高级程序设计，介绍类的继承、多态和接口的设计。第 6 章为 Windows 编程基础，介绍常用基本控件的使用。第 7 章为 Windows 窗体的高级功能，介绍菜单、工具栏、状态栏、对话框及 ActiveX 控件的使用。第 8 章为文件操作，介绍常用文件操作类的使用及文本文件和二进制文件的读写。第 9 章为数据库编程，介绍 C#对数据库操作的方法。第 10 章为图形与图像，介绍 GDI＋的概念，使读者掌握基本图形的绘制与填充。第 11 章为网络编程，介绍常用网络编程方法。第 12 章为综合实例，精心设计了两个综合实例，使读者掌握复杂程序的设计思想与方法。

本书的编写特色如下：

(1) 知识结构设计合理，内容由浅入深，适合初学者掌握程序设计的思想。

(2) 注重独立思考能力的培养，每个实验案例都包括问题分析、操作步骤。问题分析给读者一些简单的提示，如果通过问题分析即可解决问题，读者可以不看操作步骤，自行完成实验，从而提高分析问题和解决问题的能力。

(3) 强调编程能力的培养，每个实验都附有实践提高环节，该环节的习题只给出提示，不给出具体步骤，读者可以通过解决这些习题掌握实验内容，从而提高编程能力。

(4) 强调综合能力的培养，本书第 12 章精选了两个综合案例，为读者介绍复杂程序的设计过程，使其初步具有软件开发的概念。

本书可作为高等院校计算机及其相关专业的本科教学用书，也可用作非计算机专业公共课基础教材，亦可作为计算机爱好者的自学参考书。

本书由河北联合大学刘丽、张淑芬和朱俊东编写，融入了编者多年的教学和项目开发经验。刘丽编写第 2、3、8、10 章，张淑芬编写第 4～7、9、11、12 章，朱俊东编写第 1 章，全书由刘丽统稿。

由于编者时间仓促，水平有限，书中难免存在疏漏和不足，敬请读者批评指正。

编　者

2014 年 2 月

目录

第1章 Visual Studio 2010 集成开发环境的使用

实验 1.1 建立一个简单的控制台程序

实验目的

- 熟悉 Visual Studio 2010 集成开发环境。
- 了解 C# 程序的组成。
- 了解控制台应用程序的创建流程。

实验内容

【案例 1-1】 编写一个简单的控制台应用程序。

1. 问题分析

本案例要创建一个简单的 C# 控制台程序，该程序将在屏幕上输出一行文字，即“这是我的第一个 C# 控制台程序”。

程序中用到的主要是控制台的 WriteLine()方法。

2. 操作步骤

(1) 启动 Visual Studio 2010，选择“文件”→“新建”→“项目”选项，弹出如图 1-1 所示的对话框，在对话框右侧的项目类型列表中选择“控制台应用程序”，在对话框下方输入控制台应用程序的名称和位置，如本例的项目名称为 ex1_1，位置为 D:\C#\1，单击“确定”按钮，即可进入代码编辑窗口。

(2) 在 Main()方法中输入如下代码。

```
Console.WriteLine("这是我的第一个 C#控制台程序");
Console.ReadLine();
```

整个程序的结构如图 1-2 所示。

(3) 执行程序，单击“调试”菜单的“启动调试”命令，或直接按下 F5 键执行程序，程序运行结果如图 1-3 所示。

图 1-1 “新建项目”窗口

图 1-2 案例 1-1 的程序结构

图 1-3 案例 1-1 的运行结果

说明

① 如果第一次启动 Visual Studio 2010，将出现一个对话框，提示用户设置默认的开

发环境，如图 1-4 所示，在此对话框的“选择默认环境设置”列表框中选择“Visual C# 开发设置”选项，之后单击“启动 Visual Studio”按钮即可。

图 1-4 “选择默认环境设置”对话框

② 程序中的第 1 行～第 4 行是由 using 引导的语句，它们是使得程序能够顺利编译所需要的常用服务，如果删除，程序将不能编译。其实，对于本案例这样简单的控制台程序来说，只需要用到 Console 类的相关方法，而 Console 类是包含在 System 命名空间下的。所以，如果删除 using System 代码行下面的 3 行语句，本程序也不会受到影响。

③ 本程序的命名空间(namespace)为 ex1_1，其实这一命名空间是在“新建项目”对话框中输入的项目的名称。

④ Main()方法为程序的入口点，也就是说，Main()方法是应用程序执行时调用的第一个方法，而且程序以 Main()方法的结束作为整个程序执行结束的标志。

⑤ WriteLine 方法用于将字符串显示在屏幕上，具体使用方法可参见第 3 章。

⑥ //引导的部分为注释内容，该部分编译系统不会执行。

⑦ 分号(;)是语句结束的标志，如果缺少会报错。

【案例 1-2】 输入程序代码，并分析观察程序的运行结果。

1. 问题分析

本案例将利用控制台的 WriteLine()方法输出一个问题，并给出 4 个选项(A、B、C、D)。通过本程序可进一步熟悉 WriteLine()方法的使用。

2. 操作步骤

(1) 新建一个控制台应用程序，过程仿照案例 1-1。为该控制台程序的项目命名为 ex1_2。

(2) 创建好项目后，在 Main()方法中输入如下代码。

```
Console.WriteLine("你知道 VS2010 下启动调试的快捷键是哪个吗?");
Console.WriteLine("A.F2\nB.F3\nC.F4\nD.F5\n");
Console.ReadLine();
```

添加完上述代码后，该程序的整个结构如图 1-5 所示。

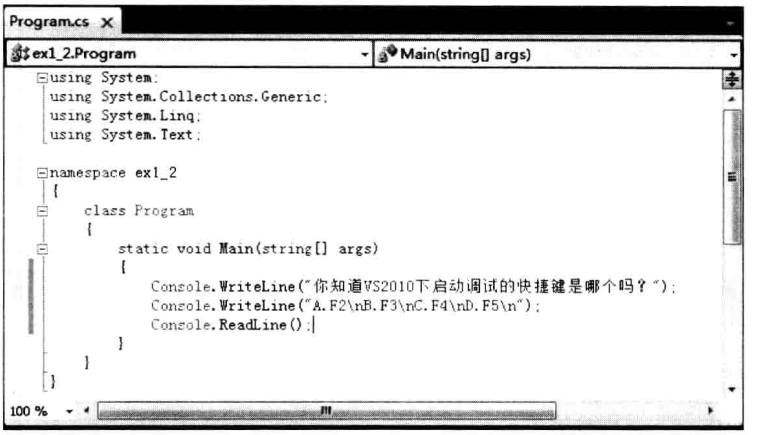

图 1-5　案例 1-2 的程序结构

程序分析

程序中的主要代码是调用控制台(Console)的 WriteLine()方法。第一个 WriteLine()方法中双引号内的内容将原样输出，输出完毕后光标将换到下一行；第二个 WriteLine()方法将输出 4 个选项。请注意观察：每个选项后有一个\n，\n 是转义字符，起到换行的作用。程序运行结果如图 1-6 所示。

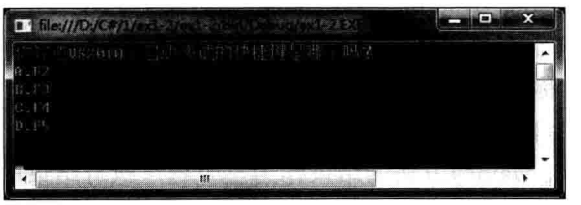

图 1-6　案例 1-2 的运行结果

实践提高

设计一个控制台应用程序，并在输出窗口输出如下内容。

```
****************************************
寒雨连江夜入吴，平明送客楚山孤。
洛阳亲友如相问，一片冰心在玉壶。
                    ----王昌龄
****************************************
```

程序运行结果如图 1-7 所示。

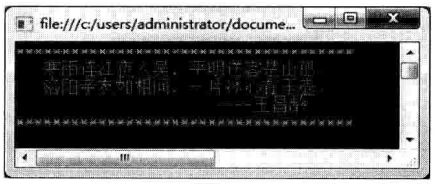

图 1-7　运行结果

提示　本案例主要涉及的内容为输出，可以使用控制台的 WriteLine()方法，也可以使用 Write()方法。

实验 1.2　第一个 Windows 窗体应用程序

实验目的

- 熟悉开发 Windows 窗体应用程序的环境。
- 熟悉 C# 中创建 Windows 窗体应用程序的流程。
- 能够创建简单的窗体，并添加常用的控件。

实验内容

【案例 1-3】 创建一个如图 1-8 所示的窗体应用程序。

1. 问题分析

本案例将创建一个简单的 Windows 窗体应用程序，该程序要求在一个窗体中包含一个 Label(标签)控件和一个 Button(按钮)控件，单击按钮时，窗体上将显示文字“有趣的窗体应用程序!”。通过本例的学习，读者将对 Windows 窗体应用程序的创建过程有一个简单了解。

2. 操作步骤

(1) 新建一个 Windows 窗体应用程序。选择“文件”→“新建”→“项目”选项，弹出如图 1-1 所示的“新建项目”对话框，在项目类型列表框中选择“Windows 窗体应用程序”，确定项目的存储位置为 D:\C#\1，确定项目名称为 ex1_3，单击“确定”按钮。

(2) 在“工具箱”的“公共控件”区域选择 Button 控件，并拖动到窗体的合适位置，在“工具箱”的“公共控件”区域选择 Label 控件，拖动到窗体的合适位置。

(3) 设置控件属性。选择按钮 button1，在“属性”窗口中找到 Text 属性，设置其值为“显示”；选择标签 label1，设置其 Text 属性为空(即原来其值为 label1，直接删除即可)，如图 1-8 所示。

(4) 双击按钮 button1，进入代码编辑窗口，在 button1 的 Click 事件中输入如下

代码。

```
label1.Text="有趣的窗体应用程序!";
```

执行程序，运行结果如图 1-9 所示。

图 1-8　案例 1-3 的设计界面

图 1-9　案例 1-3 的运行结果

整个程序的结构如图 1-10 所示。

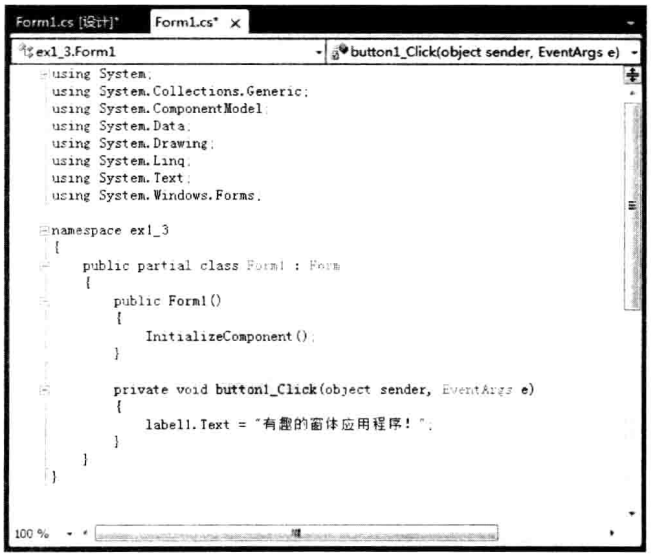

图 1-10　案例 1-3 的程序结构

说明

① 如果“工具箱”窗口、“属性”窗口等没有出现，可以通过“视图”菜单将其显示出来。

② Windows 窗体应用程序的结构大体与控制台程序相似。

③ 创建 Windows 窗体应用程序的一般流程为添加控件→为控件或窗体设置属性→编写事件代码。

【案例 1-4】 创建一个如图 1-11 所示的 Windows 窗体应用程序。

1. 问题分析

本案例要创建的 Windows 窗体应用程序可以实现以下计算功能：在第 1 个文本框和第 2 个文本框中分别输入两个整数，单击“计算”按钮，两个整数之和将出现在第 3 个文本框中。

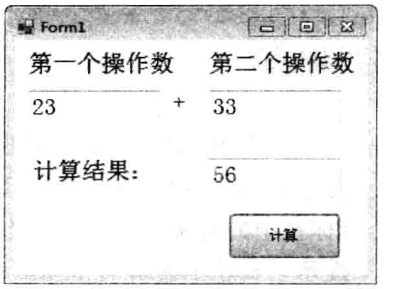

图 1-11　案例 1-4 的运行结果

2. 操作步骤

(1) 创建一个 Windows 窗体应用程序，将其命名为 ex2_2。

(2) 在窗体中添加 3 个 TextBox(文本框)控件、4 个标签控件、1 个按钮控件，按图 1-11 所示设置控件的位置，按表 1-1 所示设置控件的属性。

表 1-1　案例 1-4 各控件的属性设置

控件名	属性名	属性值	控件名	属性名	属性值
label1	Text	第一个操作数	label4	Text	计算结果：
label2	Text	第二个操作数	button1	Text	计算
label3	Text	+			

(3) 编写代码。双击按钮 button1，进入代码编辑状态，并在 button1 的 Click 事件中输入如下代码。

```
//定义一个整型变量,存储计算结果
int x;
//计算两个文本框中的数值之和
x=Convert.ToInt32(textBox2.Text)+Convert.ToInt32(textBox1.Text);
//将 x 转换为字符串类型后再显示在第 3 个文本框中
textBox3.Text=Convert.ToString(x);
```

(4) 执行程序，运行结果如图 1-11 所示。

说明

① 程序中共 3 行代码，以//引导的为注释。

② 第 1 条语句告知编译系统：定义一个整型变量 x，x 的作用是存储计算结果。

③ 第 2 条语句实现的功能是将两个文本框中输入的数据相加，并将结果存储于变量 x 中。程序中用到了 Convert 类，这是一个静态类，提供了很多静态方法，实现数据类型之间的转换。本例中用到的是 Convert 的 ToInt32()方法，功能是将括号中的字符串转换为 32 位的基本整型。之所以使用 Convert 类，是因为文本框的 Text 属性默认为字符串型，必须将其转换为整型才可以进行数学运算。

④ 第 3 条语句的功能是将整型变量 x 的值转换为字符串，并显示在 textBox3 中。

实践提高

设计一个 Windows 窗体应用程序，要求如下。

程序初始运行时，显示如图 1-12 所示的界面；单击“显示版本”按钮，将显示一个标签，该标签中的文字为程序的版本信息，如图 1-13 所示。

提示 程序主要用到了标签的 Visible 属性。该属性设置为 False 时，标签处于不可见状态；要想显示，须将其设置为 True。

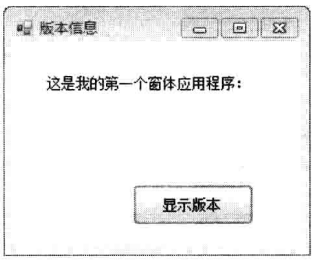

图 1-12 初始运行结果

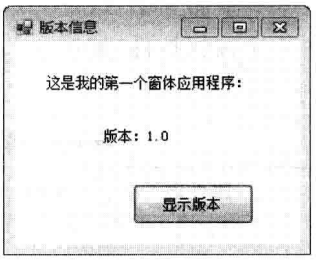

图 1-13 单击“显示版本”按钮后的运行结果

第2章 C#程序设计基础

实验 2.1 认识数据类型、运算符和表达式

实验目的

- 掌握常量与变量的概念。
- 掌握变量的声明及初始化方法。
- 熟练掌握各种基本数据类型，理解不同数据类型的转换。
- 掌握 C# 提供的常用运算符。
- 掌握表达式的运算规则，利用运算符和表达式解决简单的数学问题。

实验内容

【案例 2-1】 创建控制台应用程序，输入代码，分析程序运行结果。

1. 问题分析

C# 提供了多种整型数据类型，由于不同的整型数据类型的范围不同，适用的场合也不同。本案例要求掌握各种整型变量的定义，并利用控制台程序输出各种整数类型的边界值，以了解各种整数类型适用的场合。

2. 操作步骤

(1) 新建控制台应用程序 ex2_1，并在 Main()方法中输入如下代码。

```
//以下程序段的功能是输出 sbyte 型的最大值和最小值
sbyte smax=sbyte.MaxValue;
sbyte smin=sbyte.MinValue;
Console.WriteLine("sbyte 类型的最大值为:{0}", smax);
Console.WriteLine("sbyte 类型的最小值为:{0}", smin);
Console.WriteLine();
//以下程序段的功能是输出 byte 型的最大值和最小值
byte bmax=byte.MaxValue;
```

```
byte bmin=byte.MinValue;
Console.WriteLine("byte类型的最大值为:{0}", bmax);
Console.WriteLine("byte类型的最小值为:{0}", bmin);
Console.WriteLine();
//以下程序段的功能是输出 short 型的最大值和最小值
short shmax=short.MaxValue;
short shmin=short.MinValue;
Console.WriteLine("short类型的最大值为:{0}", shmax);
Console.WriteLine("short类型的最小值为:{0}", shmin);
Console.WriteLine();
//以下程序段的功能是输出 ushort 型的最大值和最小值
ushort ushmax=ushort.MaxValue;
ushort ushmin=ushort.MinValue;
Console.WriteLine("ushort类型的最大值为:{0}", ushmax);
Console.WriteLine("ushort类型的最小值为:{0}", ushmin);
Console.WriteLine();
//以下程序段的功能是输出 int 型的最大值和最小值
int imax=int.MaxValue;
int imin=int.MinValue;
Console.WriteLine("int类型的最大值为:{0}", imax);
Console.WriteLine("int类型的最小值为:{0}", imin);
Console.WriteLine();
//以下程序段的功能是输出 uint 型的最大值和最小值
uint uimax=UInt32.MaxValue;
uint uimin=UInt32.MinValue;
Console.WriteLine("uint类型的最大值为:{0}", uimax);
Console.WriteLine("uint类型的最小值为:{0}", uimin);
Console.WriteLine();
//以下程序段的功能是输出 long 型的最大值和最小值
long lmax=Int64.MaxValue;
long lmin=Int64.MinValue;
Console.WriteLine("long类型的最大值为:{0}", lmax);
Console.WriteLine("long类型的最小值为:{0}", lmin);
Console.WriteLine();
//以下程序段的功能是输出 ulong 型的最大值和最小值
ulong ulmax=UInt64.MaxValue;
ulong ulmin=UInt64.MinValue;
Console.WriteLine("ulong类型的最大值为:{0}", ulmax);
Console.WriteLine("ulong类型的最小值为:{0}", ulmin);
Console.WriteLine();
Console.ReadLine();
```

(2) 运行程序,查看程序的运行结果,如图 2-1 所示。

图 2-1　案例 2-1 的运行结果

说明

① 该程序的代码虽然比较多，但大多都是重复性的，以第一段代码为例：第 1 行语句和第 2 行语句分别定义了两个 sbyte 型变量，并将其初始化为 sbyte 类型的最大值和 sbyte 类型的最小值。

② MaxValue 和 MinValue 是 sbyte 类的属性，其他类型如 byte、short、ushort、Int32、UInt32、Int64、UInt64 也都具有这两个属性。

③ 后面的两个 WriteLine()方法将输出相应类型的最大值和最小值。

【案例 2-2】 输入程序代码，分析输出结果。

1. 问题分析

本案例要求学会定义各种实数类型的变量，并了解实数类型的边界值。

2. 操作步骤

(1) 新建一个控制台应用程序，在 Main()方法中输入如下代码。

```
//显示 float 类型的最大值与最小值
float a=Single.MaxValue;
float b=Single.MinValue;
Console.WriteLine("float 数据类型的最大值为：{0}",a);
Console.WriteLine("float 数据类型的最小值为：{0}",b);
Console.WriteLine();
//显示 double 类型的最大值与最小值
double c=Double.MaxValue;
double d=Double.MinValue;
Console.WriteLine("double 数据类型的最大值为：{0}", c );
Console.WriteLine("double 数据类型的最小值为：{0}", d);
Console.WriteLine();
//显示 decimal 类型的最大值与最小值
decimal e=Decimal.MaxValue;
decimal f=Decimal.MinValue;
```