

高等院校规划教材
计算机基础教育系列

C#程序设计(C#2.0版)

实验指导及习题解答

刘慧宁 李清华 刘 蕾 等编著



机械工业出版社
CHINA MACHINE PRESS



高等院校规划教材·计算机基础教育系列

C#程序设计 (C# 2.0 版)

实验指导及习题解答

刘慧宁 李清华 刘蕾 等编著



机械工业出版社

本书是与《C#程序设计(C#2.0版)》配套的实验指导及习题解答辅导教材。本书共16章。其中,前15章为基础部分,每章针对一个知识点,设计了若干实验和习题,并给出了解答,通过这些实验和习题,可以加深读者对相关知识点的理解和掌握;第16章为综合实验,解答这些实验需要综合运用C#语言的多个知识点,通过练习可以增强读者使用C#语言解决实际问题的能力。

本书在编排上完全独立,内容自成一体,可以与任何C#相关程序设计教材配套使用,也适合有一定语言基础的读者单独使用,以巩固自己所学的知识。

图书在版编目(CIP)数据

C#程序设计(C#2.0版)实验指导及习题解答 / 刘慧宁等编著. —北京:机械工业出版社, 2008.10

(高等院校规划教材·计算机基础教育系列)

ISBN 978-7-111-25175-0

I. C… II. 刘… III. C语言—程序设计—高等学校—教学参考资料
IV. TP312

中国版本图书馆CIP数据核字(2008)第146468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:唐德凯

责任印制:邓博

北京双青印刷厂印刷

2009年1月第1版·第1次印刷

184mm×260mm·14.75印张·362千字

0001—4000册

标准书号:ISBN 978-7-111-25175-0

定价:25.00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) 88379753 88379739

封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前言

计算机编程是一门实践性很强的学科，纸上谈兵永远无法真正学会它。因此，实践是读者掌握计算机编程的最重要环节。

为了指导读者的编程实践，笔者编写了本书。本书是与《C#程序设计（C# 2.0 版）》配套的实验指导及习题解答辅导教材。

本书共 16 章。其中，前 15 章为基础部分，每章针对一个知识点，设计了若干实验和习题，并给出了解答，通过这些实验和习题，可以加深读者对相关知识的理解和掌握；第 16 章为综合实验，解答这些实验需要综合运用 C# 语言的知识点，通过练习可以增强读者使用 C# 语言解决实际问题的能力。

必须指出，编程是一项创造性的活动，编程题的解答从来不是唯一的。因此，笔者不希望读者被书中给出的答案所束缚，应该独立思考，举一反三，尽量自己完成之后再参考答案。当然，读者也可以将解答当做例题阅读，因为，读程序同样是程序员需要掌握的重要技能。

本书所有的程序都在 Microsoft .NET Framework SDK v2.0 环境下调试运行通过。

参加本书编写工作的还有刘晓杰、梅向伟、王云清、向礼图、刘兰兰、程周财、刘光华等。

本书错误及不足之处，敬请读者及专业人士不吝赐教，我们将非常感激并及时更正。我们的 E-mail 是 liuhnbbox@yahoo.com.cn。

作者

目 录

前言

第 1 章 概述	1
1.1 基本实验	1
1.1.1 安装.NET Framework SDK	1
1.1.2 编辑、编译和运行 C#程序	2
1.1.3 使用.NET Framework SDK 文档	3
1.2 选做实验	5
1.3 习题及解答	5
第 2 章 C#语言编程基础	7
2.1 基本实验	7
2.1.1 求矩形的周长和面积	7
2.1.2 判断某年是否闰年	8
2.1.3 求三个整数的最大值	10
2.1.4 日期描述	11
2.2 选做实验	12
2.3 习题及解答	13
第 3 章 控制语句	19
3.1 基本实验	19
3.1.1 整数排序	19
3.1.2 简单计算器	20
3.1.3 求指定范围内的所有合数	23
3.2 选做实验	24
3.3 习题及解答	24
第 4 章 方法	31
4.1 基本实验	31
4.1.1 求两个正整数的最大公约数和最小公倍数	31
4.1.2 给三个整数排序并求其和及平均值	32
4.1.3 求 $n!$ 的值	34
4.2 选做实验	35
4.3 习题及解答	35
第 5 章 类	49
5.1 基本实验	49
5.1.1 图书类	49
5.1.2 DLL 文件	52
5.2 选做实验	54

5.3	习题及解答	54
第 6 章	继承与多态	65
6.1	基本实验	65
6.1.1	简单几何图形描述	65
6.1.2	求三角形的面积	77
6.2	选做实验	78
6.3	习题及解答	78
第 7 章	数组	92
7.1	基本实验	92
7.1.1	矩阵转置	92
7.1.2	求两个整数集合的并集和交集	93
7.2	选做实验	96
7.3	习题及解答	97
第 8 章	异常处理	104
8.1	基本实验	104
8.1.1	默认异常处理	104
8.1.2	完善简单计算器程序	105
8.2	选做实验	108
8.3	习题及解答	109
第 9 章	索引器	112
9.1	基本实验	112
9.1.1	英汉电子词典	112
9.1.2	多边形描述	114
9.2	选做实验	117
9.3	习题及解答	117
第 10 章	运算符重载	121
10.1	基本实验	121
10.1.1	复数运算	121
10.1.2	结构和类中的运算符	125
10.2	选做实验	126
10.3	习题及解答	126
第 11 章	委托与事件	135
11.1	基本实验	135
11.1.1	消防监控系统	135
11.1.2	Windows 编程	138
11.2	选做实验	140
11.3	习题及解答	140
第 12 章	字符与字符串	144
12.1	基本实验	144

12.1.1	身份证信息提取	144
12.1.2	两个字符串交叉合并	146
12.2	选做实验	148
12.3	习题及解答	148
第 13 章	集合	155
13.1	基本实验	155
13.1.1	医务室模拟	155
13.1.2	通信录管理	158
13.2	选做实验	161
13.3	习题及解答	161
第 14 章	泛型	168
14.1	基本实验	168
14.1.1	泛型链表	168
14.1.2	算术表达式中括号是否配对	173
14.2	选做实验	173
14.3	习题及解答	173
第 15 章	文件和流	180
15.1	基本实验	180
15.1.1	打印年历	180
15.1.2	二进制文件读写	183
15.2	选做实验	184
15.3	习题及解答	184
第 16 章	综合实验	197
16.1	工资管理系统	197
16.2	四则运算	214
16.3	猜数游戏	219

第1章 概 述

1.1 基本实验

1.1.1 安装.NET Framework SDK

实验题目:

下载并安装.NET Framework SDK。

实验目的:

- 1) 学习从网络下载.NET Framework SDK。
- 2) 熟悉.NET Framework SDK 的安装过程。

实验步骤:

- 1) 下载 Microsoft .NET Framework 2.0 版可再发行组件包。
- 2) 下载 Microsoft .NET Framework 2.0 软件开发工具包。
- 3) 安装 Microsoft .NET Framework 2.0 版可再发行组件包。
- 4) 安装 Microsoft .NET Framework 2.0 软件开发工具包。

参考答案:

要获得.NET Framework SDK,应连上微软公司的 MSDN 网站(<http://msdn.microsoft.com>),其中的信息和超链接会引导读者下载相应的 SDK。本书下载的版本是基于 x86 处理器的.NET Framework SDK 2.0 (更新的版本已经推出)。由于安装这个 SDK 之前,必须首先安装相应的可再发行组件包。因此,读者也需要下载它,即对应于 x86 处理器,读者应下载下述两个文件:

- (1) Microsoft .NET Framework 2.0 版可再发行组件包 (x86)

Microsoft .NET Framework 2.0 版可再发行组件包将安装运行针对.NET Framework 2.0 版开发的应用程序时所需的.NET Framework 运行库及相关文件。文件名为 dotnetfx.exe。

- (2) Microsoft .NET Framework 2.0 软件开发工具包 (SDK) (x86)

Microsoft .NET Framework 软件开发工具包 (SDK) 2.0 版包括了开发人员在 x86 平台上编写、生成、测试和部署 .NET Framework 应用程序所需的工具、文档和示例。文件名为 setup.exe。

由于.NET Framework SDK 2.0 目前只能安装在 Windows 2000 Service Pack 3、Windows Server 2003、Windows XP Service Pack 2、Windows Vista 或更高版本的操作系统中,因此,安装前,必须确保当前计算机的操作系统满足这一条件;否则,就应为当前计算机改装操作系统或将.NET Framework SDK 2.0 安装到其他满足条件的计算机中。

如果当前计算机满足条件,就可以在其中安装.NET Framework SDK。安装时,必须先安装可再发行组件包(dotnetfx.exe),然后再安装 SDK 软件(setup.exe)。

下载并安装 SDK 后，读者就可以单击“开始”，依次指向“所有程序”（或“程序”）和“Microsoft .NET Framework SDK v2.0”，然后单击“SDK 命令提示”，进入 SDK 命令提示窗口，编译并运行 C#程序。

1.1.2 编辑、编译和运行 C#程序

实验题目：

编辑、编译和运行习题 6 解答给出的 C#应用程序。

实验目的：

- 1) 了解 C#应用程序的基本结构。
- 2) 熟悉 C#应用程序的编译和运行过程。

实验步骤：

- 1) 录入习题 6 解答给出的程序源代码并保存。
- 2) 编译该应用程序。
- 3) 运行该应用程序。

参考答案：

使用编辑程序（如 Windows Notepad 等），录入习题 6 解答给出的源代码，并以普通文本格式保存。保存时，文件名取 `WelcomeToCS.cs`。因为较旧版本的记事本不能自动识别 `.cs` 扩展名，所以当读者以文本文档类型保存时，C#源程序文件的文件名可能需要用双引号括起来，比如：“`WelcomeToCS.cs`”（新版本的记事本已经不再需要）。

建立源程序文件 `WelcomeToCS.cs` 后，就需要使用编译程序 `csc.exe` 编译它。为了编译这个程序，应进入 SDK 命令提示窗口，并在保存文件 `WelcomeToCS.cs` 的目录下输入命令：

`csc WelcomeToCS.cs`

如果.NET Framework SDK 安装正确、程序源代码没有错误且保存的文件名正确，上述编译过程不会报告任何错误，否则会输出出错信息（即出现了编译错误）。如果编译时输出了出错信息，读者在继续下一步之前应改正错误，并重新编译。比如，故意漏掉语句 `Console.WriteLine("Welcome to C#!");` 最后的分号，编译就会输出如图 1-1 所示的出错信息。

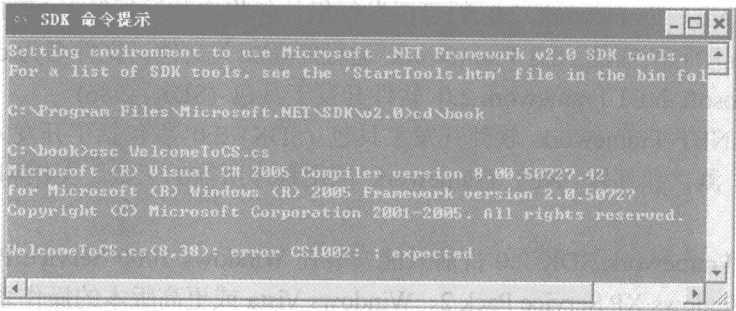


图 1-1 编译错误

图 1-1 中输出的出错信息指出程序源代码的第 8 行第 38 个字符，期待一个分号，但没有找到，因此，程序语法错误，无法编译通过。这时，就应该改正错误，并重新编译。

如果源程序文件 `WelcomeToCS.cs` 被编译通过，就会在同一目录下生成一个名为 `WelcomeToCS.exe` 的文件。这可以通过在同一目录下运行如下命令来验证：

```
dir WelcomeToCS.exe
```

应该注意的是，文件 `WelcomeToCS.exe` 并不是真正的机器码，而是一种中间码，该码遵循 MSIL 语法，因此，`WelcomeToCS.exe` 必须在支持 MSIL 的平台上运行。`.NET` 平台支持 MSIL，所以可以运行它。运行时，`.NET CLR` 中的 JIT 编译器负责将 MSIL 翻译成机器代码以供操作系统执行。具体命令如下：

```
WelcomeToCS
```

运行后，将在屏幕上输出：

```
Welcome to C#!
```

1.1.3 使用.NET Framework SDK 文档

实验题目：

查阅 .NET Framework SDK 文档，了解 `Console.WriteLine` 方法的使用。

实验目的：

- 1) 熟悉 .NET Framework SDK 文档。
- 2) 了解 `Console.WriteLine` 方法的使用。

实验步骤：

- 1) 启动 .NET Framework SDK 文档。
- 2) 在 .NET Framework SDK 文档中查阅 `Console.WriteLine` 方法。

参考答案：

要启动 .NET Framework SDK 文档，读者应该单击“开始”，依次指向“所有程序”（或“程序”）和“Microsoft .NET Framework SDK v2.0”，然后单击“文档”。.NET Framework SDK v2.0 文档启动后的界面如图 1-2 所示。

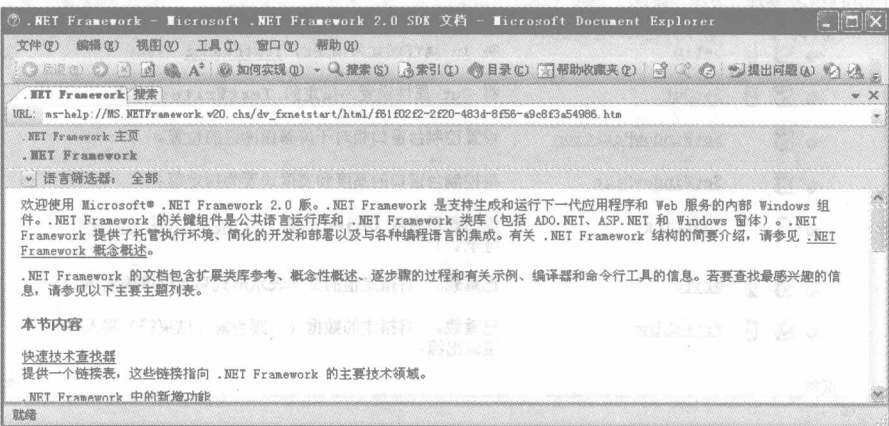


图 1-2 .NET Framework SDK v2.0 文档启动后的界面

单击其中的“搜索”选项卡，搜索 Console 类。搜索结果如图 1-3 所示。

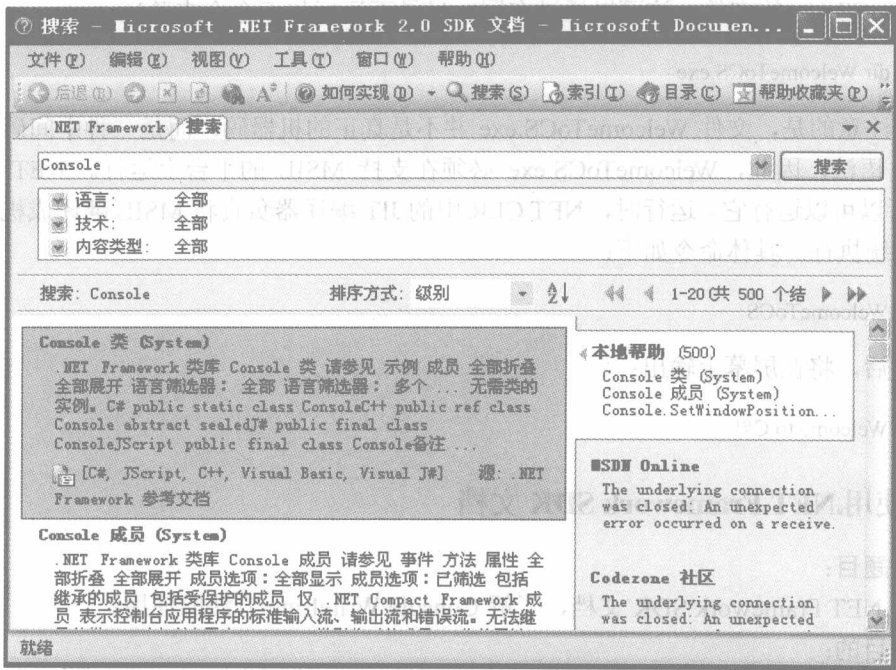


图 1-3 搜索 Console 类

单击其中的“Console 成员 (System)”，结果如图 1-4 所示。

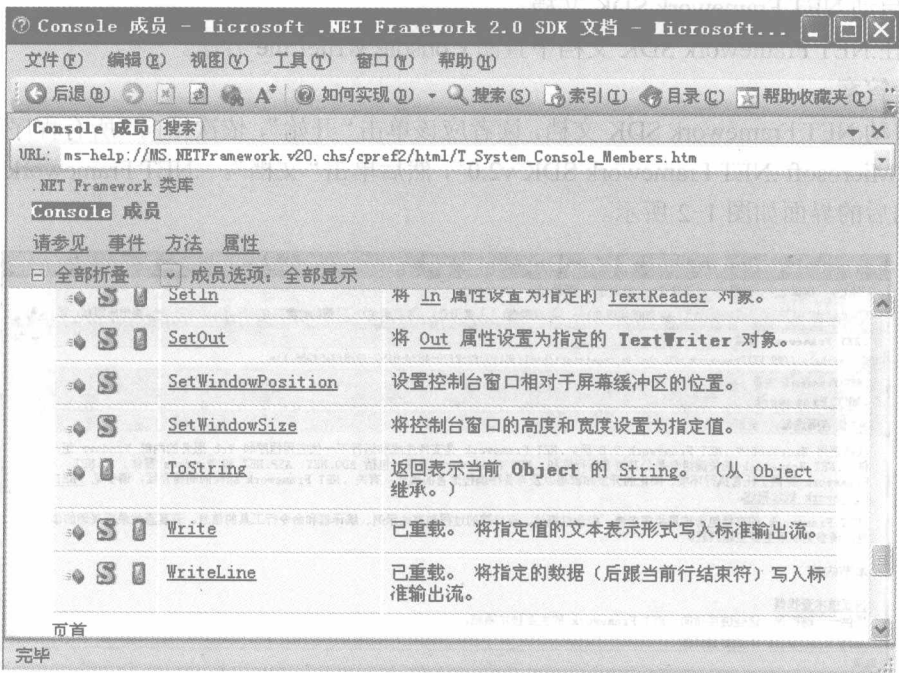


图 1-4 Console 类的成员

单击其中的“WriteLine”，结果如图 1-5 所示。

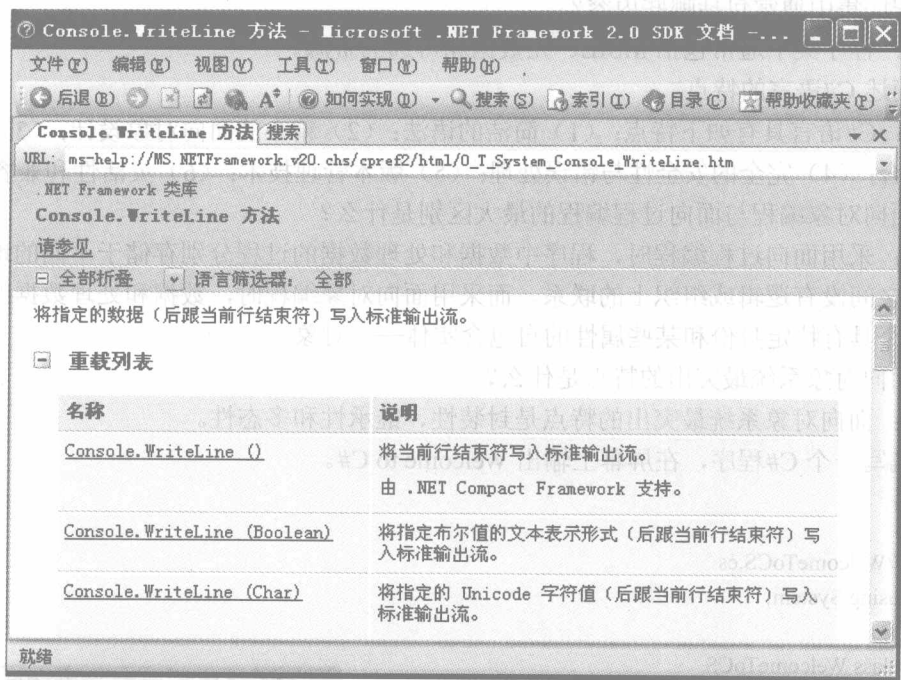


图 1-5 WriteLine 方法

单击相应的 WriteLine 方法，就可以查看相应方法的意义及使用方式。

1.2 选做实验

1. 编译如下程序，编译过程是否输出出错信息？如果编译时输出了出错信息，分析出错信息，并改正错误，然后重新编译。

```
class WelcomeToCS
{
    static void Main()
    {
        Console.WriteLine("Welcome to C#!");
    }
}
```

2. 查阅 .NET Framework SDK 文档，了解 Console.Write 方法的使用。
3. 使用 Console.Write 方法编写一个程序，编辑、编译并运行该程序。

1.3 习题及解答

1. .NET 框架包括哪两部分？

解答：.NET 框架包含公共语言运行库和框架类库两部分。

2. 程序集中通常包括哪些内容？

解答：程序集中通常包括 MSIL、元数据和可选的资源。

3. 简述 C#语言的特点。

解答：C#语言具有如下特点：（1）简洁的语法；（2）精心的面向对象设计；（3）与 Web 的紧密结合；（4）完全的安全性及错误处理；（5）版本管理技术；（6）灵活性和兼容性。

4. 面向对象编程与面向过程编程的最大区别是什么？

解答：采用面向过程编程时，程序中数据和处理数据的过程分别存储于不同的地方，数据和过程之间没有逻辑或组织上的联系。而采用面向对象编程时，数据和处理数据的过程被抽象成一个具有特定身份和某些属性的自包含实体——对象。

5. 面向对象系统最突出的特点是什么？

解答：面向对象系统最突出的特点是封装性、继承性和多态性。

6. 编写一个 C#程序，在屏幕上输出 Welcome to C#。

解答：

```
//WelcomeToCS.cs
using System;

class WelcomeToCS
{
    static void Main()
    {
        Console.WriteLine("Welcome to C#!");
    }
}
```



图 1-6 习题 6 的运行结果

运行结果如图 1-6 所示。

7. 注释的作用是什么？C#语言有哪两种常用的注释形式？

解答：编写程序时，在程序的源代码中加上必要和清晰的注释，可以大幅提高程序的可读性，便于事后维护。

C#语言中，常用的注释有两种：

（1）单行注释。

以符号“//”开始，本行其后的所有字符都是注释信息。

（2）多行注释。

“/*”表示注释的开始，“*/”表示注释的结束。在“/*”与“*/”之间的字符均被视为注释信息。

第2章 C#语言编程基础

2.1 基本实验

2.1.1 求矩形的周长和面积

实验题目:

已知两个矩形的长和宽,编程求它们的面积和周长。假设,矩形1的长和宽分别为20和50;矩形2的长和宽分别为3.6和4.5。

实验目的:

- 1) 掌握 double 型变量的声明和使用方法。
- 2) 熟悉基本算术运算符的运算规则。
- 3) 掌握表达式的使用及运算符的优先级概念。
- 4) 了解 C#语言的控制台输出方法。

实验步骤:

1) 声明 double 型变量 length、width, 分别用于表示矩形的长和宽, 并将 length 初始化为20, width 初始化为50。

2) 声明 double 型变量 area 用于表示矩形的面积, 并将它初始化为 length 和 width 的积(即矩形1的面积)。

3) 声明 double 型变量 perimeter 用于表示矩形的周长, 并将它初始化为 length 和 width 的和与2的积(即矩形1的周长)。

4) 分别输出矩形1的面积和周长。

5) 将矩形2的长和宽分别赋值给变量 length 和 width。

6) 分别求矩形2的面积和周长并相应赋值给变量 area 和 perimeter。

7) 分别输出矩形2的面积和周长。

参考答案:

```
//Test2_1.cs
using System;
```

```
class Test2_1
{
```

```
    static void Main()
```

```
    {
```

```
        double length=20,width=50;
```

```
        double area=length*width;
```

```
        double perimeter=(length+width)*2;
```

```

Console.WriteLine("矩形 1 长:{0} 宽:{1}",length,width);
Console.WriteLine("      面积:{0} 周长:{1}",area,perimeter);
Console.WriteLine();

length=3.6;
width=4.5;
area=length*width;
perimeter=(length+width)*2;

Console.WriteLine("矩形 2 长:{0} 宽:{1}",length,width);
Console.WriteLine("      面积:{0} 周长:{1}",area,perimeter);
}

```

运行结果如图 2-1 所示。

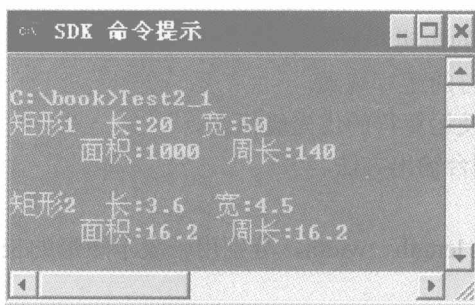


图 2-1 实验 2.1.1 的运行结果

2.1.2 判断某年是否闰年

实验题目:

编写一个程序,判断用户从键盘输入的年份是否是闰年。如果一个年份能被 4 整除,并且如果该年份是 100 的倍数,它不能被 400 整除,那么该年份就是闰年。

实验目的:

- 1) 掌握 int 型、bool 型和 string 型变量的声明和使用方法。
- 2) 了解 C#语言的控制台输入方法。
- 3) 熟悉基本算术运算符的运算规则。
- 4) 熟悉关系运算符的运算规则。
- 5) 熟悉条件逻辑运算符的运算规则。
- 6) 熟悉条件运算符的运算规则。
- 7) 理解运算符的优先级概念。

实验步骤:

- 1) 声明 int 型变量 year, 用于表示年份。
- 2) 声明 bool 型变量 isLeap, 用于表示是否闰年。

- 3) 声明 `string` 型变量 `str`, 用于表示一串字符。
- 4) 向屏幕输出提示信息, 提示用户输入年份。
- 5) 从键盘读入年份字符串赋值给变量 `str`。
- 6) 将变量 `str` 中的数据转换成 `int` 型数据并赋值给变量 `year`。
- 7) 通过混合取余运算、关系运算和条件逻辑运算判断 `year` 是否闰年, 并将结果赋值给变量 `isLeap`。
- 8) 使用条件运算符, 当 `isLeap` 为 `true` 时, 给变量 `str` 赋字符串“是”; 否则, 赋“不是”。
- 9) 输出结果, 告诉用户相应年份是否闰年。

参考答案:

```
//Test2_2.cs
using System;

class Test2_2
{
    static void Main()
    {
        int year;
        bool isLeap;
        string str;

        Console.WriteLine("请输入年份: ");
        str=Console.ReadLine();
        year=Int32.Parse(str);

        isLeap=(year%4==0&&year%100!=0)||(year%400==0); //判断是否闰年
        str=isLeap?"是":"不是";

        Console.WriteLine("{0}年{1}闰年",year,str);
    }
}
```

运行结果如图 2-2 所示。

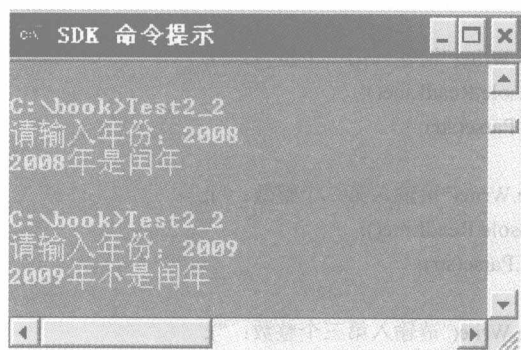


图 2-2 实验 2.1.2 的运行结果