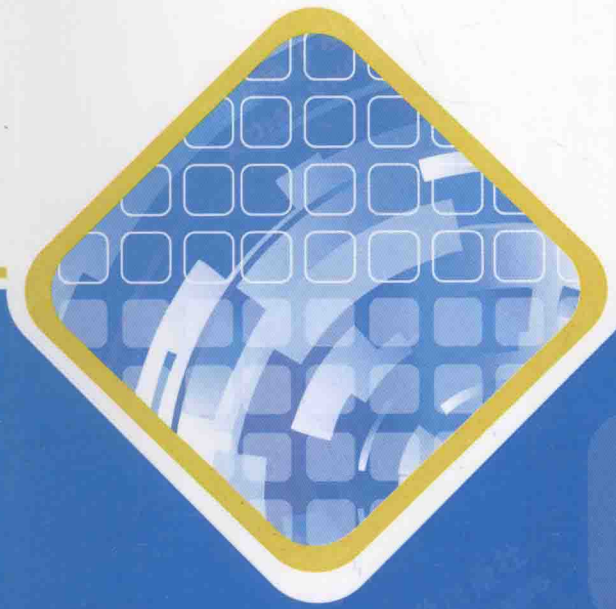
高职高专计算机专业“十二五”规划教材

Visual C# .NET 案例教程

主　编　翁健红
副主编　刘志成　林东升



西安电子科技大学出版社
http://www.xduph.com

高职高专计算机专业“十二五”规划教材

Visual C# .NET 案例教程

主　编　翁健红

副主编　刘志成　林东升

参　编　冯向科　宁云智　刘荣胜

杨茜玲　刘红梅

西安电子科技大学出版社

内 容 简 介

本书详细介绍了如何使用 C#进行 Windows 应用系统的开发，开发环境为 VS2005，数据库采用 SQL Server2000/2005。

本书共分 12 章，主要内容包括 C#语言基础、C#面向对象程序设计、窗体控件使用、ADO .NET 数据库编程、报表制作、GDI 绘图、文件操作、网络通信与多线程等。所有内容围绕项目进行组织，其中，C#语言基础知识围绕学生成绩管理系统项目，Windows 数据库应用系统开发围绕图书管理系统项目，网络通信与多线程围绕聊天室项目，GDI 绘图及文件操作等围绕画图程序项目。

本书内容丰富，结构清晰，叙述深入浅出，适合作为高职高专院校计算机类专业的教材，也可作为计算机培训的教材及自学者的参考书。

图书在版编目(CIP)数据

Visual C# .NET 案例教程 / 翁健红主编. —西安：西安电子科技大学出版社，2012. 1

高职高专计算机专业“十二五”规划教材

ISBN 978 - 7 - 5606 - 2715 - 1

Ⅰ. ① V… Ⅱ. ① 翁… Ⅲ. ① C 语言—程序设计—高等职业教育—教材 Ⅳ. ① TP312

中国版本图书馆 CIP 数据核字(2011)第 266295 号

责任编辑　胡华霖　陈　婷

出版发行　西安电子科技大学出版社(西安市太白南路 2 号)

电　　话　(029)88242885　88201467　　邮　　编　710071

网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com

经　　销　新华书店

印刷单位　陕西天意印务有限责任公司

版　　次　2012 年 1 月第 1 版　2012 年 1 月第 1 次印刷

开　　本　787 毫米×1092 毫米　1/16　　印　张　18

字　　数　426 千字

印　　数　1～3000 册

定　　价　28.00 元

ISBN 978 - 7 - 5606 - 2715 - 1/TP • 1315

XDUP 3007001 - 1

前　言

Visual C# .NET 是一种功能强大、使用简单的语言。基于 C# .NET 既可以进行传统的 C/S 模式的 Windows 应用程序开发，也可以进行基于 B/S 模式的 WEB 应用程序开发。本书重点讲解用 C#进行 C/S 模式的 Windows 应用程序开发，不过本书大部分知识对 B/S 模式的 WEB 程序开发也极有意义。本书共 12 章，各章的主要内容如下：

第 1 章介绍 C#语言基础。

第 2 章介绍数据类型与运算符。

第 3 章介绍程序控制语句。

第 4 章介绍一维数组与二维数组的使用。

第 5 章介绍方法的使用。

第 6 章介绍 C#的面向对象设计，包括类、对象、继承、多态等。

第 7 章使用用户登录案例介绍 ADO .NET 编程及 Windows 编程基础。

第 8 章使用图书管理系统主窗体案例介绍菜单、MDI 窗体、工具栏控件等。

第 9 章使用图书维护案例介绍 DataSet 对象、报表制作等。

第 10 章使用画图程序项目介绍 GDI 绘图、通用对话框、文件操作等。

第 11 章使用聊天室项目介绍网络通信与多线程。

第 12 章通过完整的图书管理系统项目展示对 C#知识的综合运用及安装程序的制作。

在本书的编写过程中，编者力求体现出职业教育的性质、任务和培养目标，坚持以就业为导向、以能力为本位的创作原则。本书所有内容围绕项目进行组织，其中 C# 语言基础知识围绕学生成绩管理系统项目，Windows 数据库应用系统开发围绕图书管理系统项目，网络通信与多线程围绕聊天室项目，GDI 绘图及文件操作等围绕画图程序项目。大部分内容采用案例驱动的教学方法，首先展示案例的运行结果并提示案例的学习目标，然后讲述相关的知识点，最后讲述案例的实现过程。

本书由翁健红主编，参与编写的有刘志成、林东升、冯向科、宁云智、刘荣胜、杨茜玲、刘红梅，全书由翁健红统稿。本书含课件、源代码、习题答案等教学资料，如需要可与作者联系。由于时间仓促和作者水平有限，书中不足与疏漏之处在所难免，敬请广大读者批评指正。

编　者

2011 年 10 月

E-mail：davewjh@163.com

目　录

第 1 章　程序开发基础

1.1　C#语言简介

C# 是用于创建运行在 Microsoft .NET 公共语言运行库上的应用程序的语言之一，它从 C 语言和 C++ 语言演化而来，是 Microsoft 专门为使用 .NET 平台而创建的，并且具备了其他语言的许多优点。

经过短短几年的发展，C# 已经成为 Windows 平台上软件开发的主流语言之一。C#如此受欢迎，主要是因为两个方面：首先，C#是专门为 .NET 而量身定制的，因而是 .NET 开发的最佳语言，.NET 框架提供的由一百多万行代码构建的类库也基本上由 C# 实现；其次，C# 是一种基于面向对象设计方法的现代语言，如果抛开一切非技术方面的因素，C# 无疑是编程语言和企业级开发语言的集大成者。其特点为面向对象、类型安全、组件技术、自动内存管理、跨平台异常处理、版本控制、代码安全管理。

1.2　Visual Studio .NET IDE 集成开发环境

Visual Studio .NET(VS .NET)是微软针对 .NET 平台提供的集成的开发环境，它包含一套完整的开发工具，可以开发 Windows 应用程序、ASP .NET Web 应用程序、XML Web services 和移动应用程序。

软件人员最关心的是开发效率，VS .NET 的人性化界面和众多工具将成倍地提高开发效率。Visual Studio .NET 是 .NET 平台下最为强大的开发工具，无论是软件服务商，还是企业应用程序的部署与发布，Visual Studio .NET 都可以提供近乎完美的解决方案。

Visual Studio .NET 提供了包括设计编码、编译调试、数据库连接操作等基本功能和基于开放架构的服务器组件开发平台、企业开发工具和应用程序重新发布工具以及性能测评报告等高级功能，Visual Studio .NET 2005 集成开发环境的操作界面如图 1-1 所示。

Microsoft 公司提供了 4 种 Visual Studio .NET 语言：C++ .NET、C#、Visual J# 和 VisualBasic .NET。无论选择何种语言来创建、测试和部署应用程序，其集成开发环境都是相同的。实际上，Visual Studio.NET 允许用户先使用一种语言创建一部分应用程序，然后使用另一种语言创建应用程序的其他部分。

每一种 Visual Studio .NET 语言都提供相近的功能，选择使用哪一种语言不但取决于该语言的特性，还取决于开发者的喜好，实际上具体选择何种语言是无关紧要的。

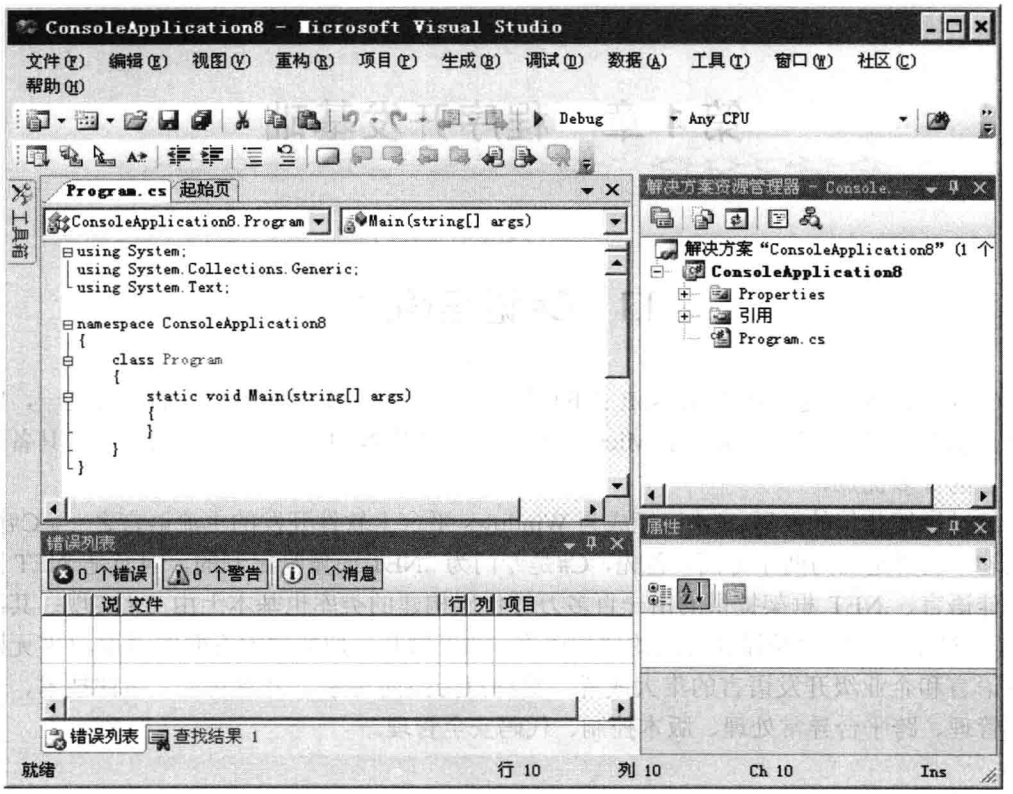


图 1-1 Visual Studio .NET 2005 集成开发环境

1.3 第一个 C#程序

【例 1-1】 编写、编译和运行一个简单的 C# 应用程序，该程序在屏幕上输出“你好!”，运行效果如图 1-2 所示。

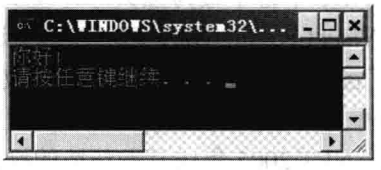


图 1-2 例 1-1 运行效果

(1) 打开 Visual Studio .NET 开发环境 IDE。

(2) 在菜单中，单击“文件”→“新建”→“项目”，弹出“新建项目”对话框。

(3) 在左边“项目类型”列表框中选择“Visual C#”下的 Windows；在右边“模板”列表框中选择“控制台应用程序”选项；在“名称”文本框中输入项目名，本处可输入“firstApp”；在下面的“位置”下拉列表框中输入该项目的保存路径。开发者可视具体情况输入或利用

右边的“浏览”按钮来选择项目的保存路径，本处选择“C:\zz”，如图 1-3 所示。

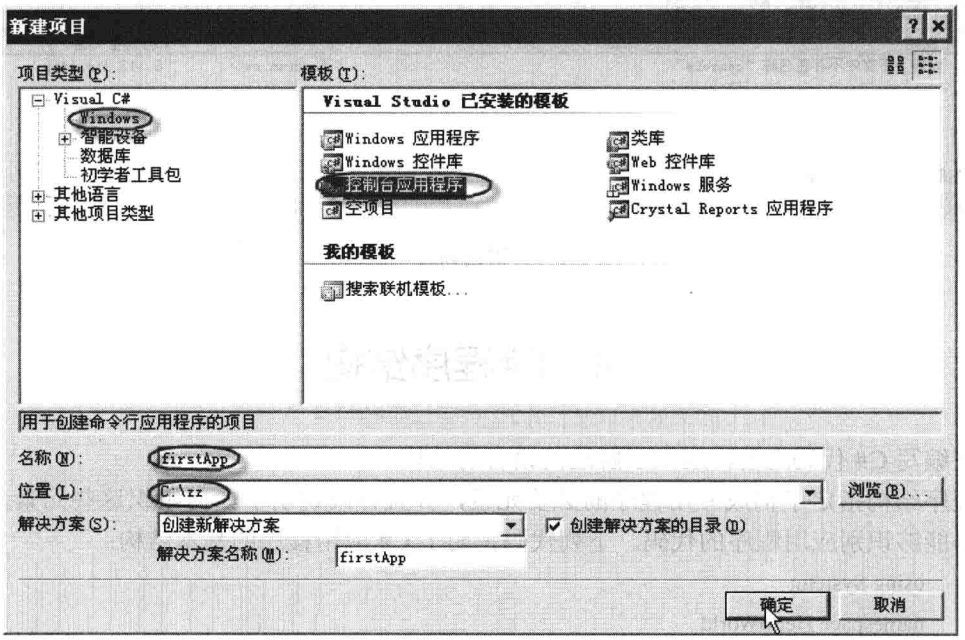


图 1-3　“新建项目”对话框

(4) 单击“确定”按钮，系统自动产生了一个名为 Program.cs 的文件，并直接进入了编辑器窗口，系统自动生成一个代码框架。

(5) 为编写一个简单的 C# 程序，可先删除 Program.cs 文件已有的内容，输入如下代码：

```
using System;
namespace HelloWorld
{
    class HelloWorldApp
    {
        static void Main(string[] args)
        {
            Console.WriteLine("你好！ ");   //输出你好！
        }
    }
}
```

(6) 编辑好 C# 代码后，单击“保存”按钮，保存此 Program.cs 文件。

(7) 在菜单中选择“调试”→“开始执行(不调试)”命令，系统编译和运行该项目，并自动打开一个输出窗口。

提示：

如果有误，在 IDE 下方可以查看错误信息，如图 1-4 所示，在错误报告信息框中的某条错误信息上双击，光标会跳到编辑窗口中与之对应的那条出错语句上，以便进行修改。

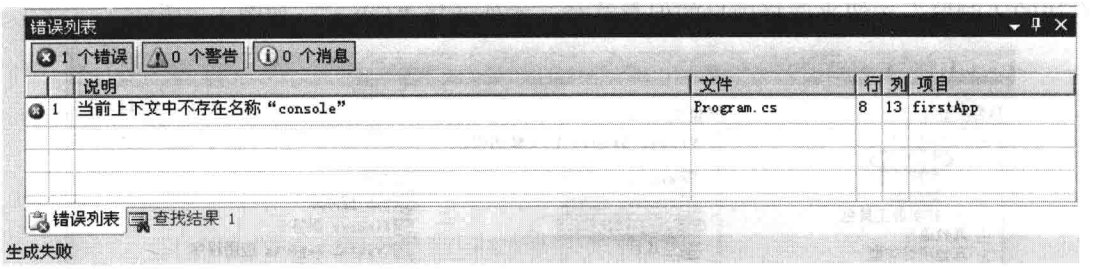


图 1-4 “错误列表”窗口

1.4 C# 程序结构

在编写 C# 代码之前，应该了解这门语言的结构。

程序结构指定了构成应用程序的必要元素，并且详细说明了如何组织这些元素，以使编译器能够识别应用程序的代码。下列代码说明了 C# 应用程序的基本结构：

```
using System;
namespace HelloWorld
{
    class HelloWorldApp
    {
        static void Main(string[] args)
        {
            Console.WriteLine("你好！ ");   //输出你好！
        }
    }
}
```

以下是对上述代码中的元素和组织原则的简要描述：

(1) using 关键字。using 关键字可以引用 Microsoft .NET 框架类库中的现有资源。通常，在程序文件的开头使用这个关键字。通过多次使用该关键字，程序可以引用多种资源。

(2) System 命名空间。System 命名空间提供了对构建应用程序所需的所有系统功能的访问。程序中用到的类 Console 在命名空间 System 中定义。

(3) 类。在 C# 或其他任何面向对象语言的编程过程中，都需要编写类，并用类来创建对象。例如，class HelloWorldApp 语句定义了一个名为 HelloWorldApp 的类。

(4) Main 方法。Main 方法用来描述类的行为。上面示例中的 static void Main 是一个全局方法，指编译器从该处开始执行应用程序，是程序运行的入口。需要注意的是，每个 C# 应用程序都必须在组成程序的某一个类中包含 Main 方法。

(5) 语句。语句就是在 C# 应用程序中执行操作的指令。语句之间用分号分隔，编译器通过分号来区分它们。在 C# 中，可以在一行中包含多条语句，也可以将一条语句拆分到多行中。尽管把一条长语句拆分为几行可能会提高可读性，但仍然推荐每行仅写一条语句。

(6) 大括号。大括号“{”和“}”用于在应用程序中标识某个代码块的开始和结束，从而可以用来对语句进行分组。每个左括号必须要有与之对应匹配的右括号。大括号可以嵌套，以表示应用程序中的不同层次。

在上面的示例中，class HelloWorldApp 之后的大括号限定了 HelloWorldApp 类的范围。在 Main 之后的大括号限定了 Main 方法中的语句的范围。

(7) 区分大小写。C# 语言区分大小写，也就是说，编译器区分大写字符和小写字符。例如在应用程序中，code、Code 和 CODE 是不同的变量名，相互不能替代。

(8) 注释。注释在运行时不会被执行，而只用于说明程序，便于程序员理解。在双斜杠(//)之后可书写不跨行的注释。

若注释很长，并跨越多行，那么就可以用另外一种方法，即用斜杠加星号(/*)来表示注释的开始，用星号加斜杠(*/)来代表注释的结束，如：

```
/*
这是一个
多行注释的示例
  */
```

1.5 习　　题

1. 解释 C# 程序的基本结构。
2. 编写程序，在屏幕上输出“欢迎学习 C# 程序设计!”。

第 2 章　数据类型与运算符

2.1 变　　量

变量是计算机内存中被命名的数据存储单元，其中存储的值是可以改变的。在程序中通过变量名来引用其中存储的信息。变量名实际上是一个符号地址，在程序进行编译时由系统给每一个变量分配一个真正的内存地址。在程序中通过变量取值即通过变量名找到相应的内存地址，再从其中读取数据或存入数据。

从用户的角度看，变量是用来描述一条信息的名称的，在变量中可以存储各种类型的信息，比如某人的年龄、工资等。

2.1.1 变量名

和其他高级语言一样，用来标识变量名、常量名、对象名、过程名等的有效字符序列称为标识符，简单地说，标识符就是一个名字。C# 对变量命名的最低要求是它应该能标识并区别变量类型、变量作用范围、常量、对象以及过程，并且它应当便于理解和使用。

在 C# 中，变量名必须遵守以下规则：

- 首字符必须是字母，或下划线(但不推荐)。
- 第 2 个字符开始可以是任意字母、数字和下划线，但不能包含空格、标点符号、运算符等其他符号。
- 变量名不能与 C# 中的类库名相同。
- C# 区分大小写，所以 year、Year 和 YEAR 代表了 3 个不同的变量。

变量名不能与 C# 关键字名相同，关键字是 C# 语言的特殊标识符，用户最好不要使用，否则容易降低程序的清晰度。如果一定要用 C# 语言中的关键字作为标识符，应使用“@”字符作为前缀。以下是一些 C# 变量名：

Person(合法)　yearly-cost(合法)　name(合法)　_Debug(合法)　Max_Size(合法)　float (非法，是一个 C# 关键字)　5day(非法，第 1 个字符是数字)

2.1.2 变量声明

程序中所有用到的变量都必须在程序中定义，即遵守“先定义后使用”，或“先声明后使用”的原则。

定义(声明)变量的作用是：定义的变量名和数据类型将告诉编译器要为变量分配多少内存空间，以及变量中要存储什么类型的值。因为针对不同类型的数据，计算机将会为其分

配不同大小的存储空间。有了变量名和变量类型，一个变量的定义才是完整的。定义变量的一般格式：

[访问修饰符] 数据类型 变量名 [= 初始值];

其中：

- []内的内容可有可无，可根据需要添加。
- “修饰符”是指访问修饰符。
- “数据类型”是指变量的类型。
- “变量名”是给变量取的名称。

例如：

```
int number;              //声明一个整型变量
bool open;               //声明一个布尔型变量
decimal bankBlance;  //声明一个十进制变量
```

可以一次声明多个变量，变量名之间用逗号分隔，例如：

```
sbyte a , b;             //声明两个有符号字节型变量
int a=6,b=12,c=24 ;
```

2.1.3　给变量赋值

变量的值表示变量的名称所指向的内存空间中所存储的内容。变量必须先定义后使用，使用之前必须赋值。变量赋值就是将数据保存到变量中的过程。变量在定义时既可以赋值也可以不赋值，在定义时直接对变量赋值是良好的编程习惯。如果在定义时没有赋值，可以在程序代码中使用赋值语句直接对变量进行赋值。

在 C# 中，给一个变量赋值语句的格式如下：

变量名=表达式;

这里的表达式可以是数学表达式，在计算机语言中，单个常数或变量，也可算作一个表达式，由单个常数或变量构成的表达式的值，就是这个常量或变量本身。

变量赋值语句的功能是：计算表达式的值，然后将这个值赋予变量。

举例：

```
int age = 26 ;    //定义变量 age，并在定义时直接对变量赋值，变量的值为 26
int age ;         //定义变量 age，但在定义时没有对变量赋值，变量没有值
age = 26 ;        //使用赋值语句直接对变量 age 进行赋值，变量的值为 26
```

在程序中，可以给一个变量多次赋值，变量的当前值等于最后一次给该变量所赋的值。例如：

```
int number;
number = 32;    //为变量赋值 32
int num=5;      //定义一个 int 型变量 num，并将其赋予初始值 5
```

【例 2-1】 变量使用。

```
using System;
class Program
{
```

```
    static void Main(string[] args)
    {
        int num1, num2;                    //声明两个整型变量 num1、num2
        num1 = 10;                         //把 10 赋给 num1
        num2 = num1 + 1;                   //把变量 num1 的值加 1 再赋给变量 num2
        Console.WriteLine( num2);          //输出 num2 变量的值
    }
}
```

运行结果如图 2-1 所示。

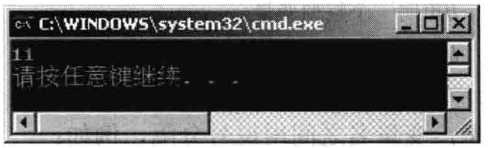


图 2-1　例 2-1 运行结果

【例 2-2】 已知圆的半径 Radius = 2.5，计算圆的面积(PI = 3.14159)。

```
using System;
class Program
{
    static void Main(string[] args)
    {
        double Radius = 2.5, Area;              //声明 double 类型变量 Radius 与 Area
        Area = Math.PI * Radius * Radius;
        Console.WriteLine("面积={0}", Area);
    }
}
```

运行结果如图 2-2 所示。

图 2-2　例 2-2 运行结果

程序说明：

Console.WriteLine 语句中{0}为占位符，输出时 Area 的值出现在{0}的位置。

2.2 常　　量

同变量不同，常量(Constant)是指在程序的运行过程中值不能改变的量。常量与变量都是数据的存储符号，区别在于存放在变量中的值是可变的，而存放在常量中的值是不变的。

(1) 整型常量。整型常量即整数，整型常量有以下三种形式：

十进制形式，即通常意义上的整数，如 123，48910 等。

八进制形式，输入八进制整型常量，需要在数字前面加“0”，如 0123、038 等。

十六进制形式，输入十六进制整型常量，需要在数字前面加“0x”或“0X”，如 0x123、0X48910 等。

(2) 实型常量。实型常量即带小数的数值，实型常量有以下两种表示形式：

小数形式，即人们通常的书写形式，如 0.123、12.3、.123 等。

指数形式，也叫科学记数，由底数加大写的 E 或小写的 e 再加指数组成，例如，123e5 或 123E5。

(3) 字符常量。字符常量是用单引号 ' ' 括起来的一个字符。如 's'、'x'、'¥' 等都是字符常量。

在表示一个字符常量时单引号内的有效字符数必须且只能是一个，并且不能是单引号或反斜杠(\)。

在 C# 中，有些字符具有特殊的含义，这时不能直接放在单引号中作为字符常量，而需要使用转义符来表示。转义符由反斜杠(\)加字符组成，如 '\n' 代表一个“换行”符。表 2-1 列出一些常见转义符。

表 2-1　转　义　符

转义符	字符名
\'	单引号
\"	双引号
\\	反斜杠
\n	新行
\r	回车
\t	水平 tab
\v	垂直 tab

【例 2-3】 转义符的使用。

```
using System;
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("This is a \t test\n");
        Console.WriteLine("\'Hello\'");
    }
}
```

运行结果如图 2-3 所示。

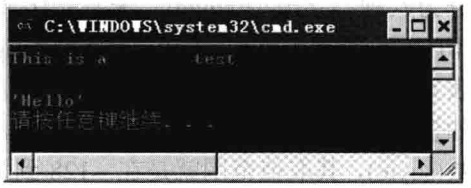


图 2-3　例 2-3 运行结果

(4) 字符串常量。字符串常量是由一对双引号界定的字符序列，例如：

"欢迎使用 C#！"

"I am a student."

需要注意的是，即使是由双引号界定的一个字符，也是字符串常量，不能当做字符常量看待，例如，'A' 与 "A"，前者是字符常量，后者是字符串常量。

(5) 布尔常量。布尔常量即布尔值本身，如前所述，布尔值 true(真)和 false(假)是 C# 的两个关键字。

2.3 数 据 类 型

程序处理的对象是各种各样的数据，因此，必须让计算机了解需要处理什么样的数据，以及采用哪种方式进行处理、按什么格式进行保存等。比如，在编码程序中需要处理单个字符、在定购票系统中需要打印货币金额、在科学运算中需要不同精度的小数，这些都是不同的数据类型。

C# 将数据分为不同的类型，分别表示不同范围、不同精度、不同用途的数据。将数据分类是因为不同类型的数据在计算机内占用的内存空间大小和运算速度不同。为了有效利用计算机的内存资源并达到最佳的程序运行效果，需要根据不同形式数据的大小和特征来选择最合适表示它们的数据类型。

C# 语言的数据类型包括了由整数类型、浮点数类型、小数类型、字符型、布尔型组成的简单类型和由结构、枚举等组成的构造类型。

2.3.1 整数类型

整数类型是指那些没有小数部分的数字，包括整数常量和整数变量。

整数常量即整数类型的常数，一般包括以下两种形式：

- 十进制数：348、−56、0 等。
- 十六进制数：这类数据以“0x”(其中“0”是数字 0)开头，如 0x61，相当于十进制数据 97。

整数类型又分有符号整数与无符号整数。有符号整数可以带正负号，无符号整数无需带正负号，默认为正数。

顾名思义，整数类型的变量的值为整数。数学上的整数可以从负无穷大到正无穷大，但是由于计算机的存储单元是有限的，所以计算机语言提供的整数类型的值总是在一定的范围之内。根据变量在内存中所占的二进制位数不同和是否有符号位，C# 语言中整数类型分为八种：字节型(sbyte)、无符号字节型(byte)、短整型(short)、无符号短整型(ushort)、整型(int)、无符号整型(uint)、长整型(long)、无符号长整型(ulong)。所占的二进制位数不同，表示的数值的取值范围也不同，所占的二进制位数越多，表示的数值的取值范围越大。比如说 8 位整数，它可以表示 2^8 个数值，即 256 个不同的数值，如果用来表示有符号 8 位整数(sbyte)，其取值范围就是在 −128 到 127 之间，而如果用来表示无符号 8 位整数(byte)，其取值范围就是在 0 到 255 之间。各整数类型及其取值范围见表 2-2。