高等院校
信息技术应用型
规划教材

C#语言程序设计

李　莹　田林琳　主编
吴晓艳　杨　玥　王　岩　田　丹　副主编

清华大学出版社

高等院校
信息技术应用型
规划教材

C#语言程序设计

李　莹　田林琳　主　编
吴晓艳　杨　玥　王　岩　田　丹　副主编
郝雪燕　王羚伊　参　编

清华大学出版社
北京

内 容 简 介

本书根据应用型人才的培养目标和“应用为本，学以致用”的教学理念，精选必需的教学内容，分别在 DOS 和 Windows 视窗两种运行环境下介绍 C＃程序设计基础知识与 Windows 应用开发技术，并以一个完整的基于三层架构的实例——图书借阅管理系统详细介绍实际项目的开发过程。全书共分 8 章，主要内容包括 C＃语言概述、.NET 框架和 Visual Studio. NET 开发工具概述，C＃语法与结构化程序设计基础，面向对象程序设计基础，基于三层架构的图书馆借阅管理系统基础设计，异常处理，ADO. NET 数据库应用程序设计，图书馆借阅管理系统的窗体设计与功能实现。

本书既可作为应用型本科计算机相关专业的专业教材，也可以作为非计算机专业学生及计算机爱好者学习 C＃语言的入门书籍。

图书在版编目(CIP)数据

C＃语言程序设计/李莹，田林琳主编. —北京：清华大学出版社，2018
(高等院校信息技术应用型规划教材)
ISBN 978-7-302-49025-8

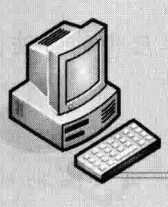
Ⅰ. ①C…　Ⅱ. ①李…　②田…　Ⅲ. ①C 语言－程序设计－高等学校－教材　Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 294526 号

责任编辑：孟毅新
封面设计：傅瑞学
责任校对：刘　静
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，010-62770175-4278
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：13.75　　**字　　数**：315 千字
版　　次：2018 年 2 月第 1 版　　**印　　次**：2018 年 2 月第 1 次印刷
印　　数：1～2000
定　　价：39.00 元

产品编号：076007-01



前　言

开发 Windows 应用软件的程序员都希望又快又好地开发出满足用户需求的软件产品，当然这除了要依靠程序员的能力和勤奋以外，还要有好用的软件开发平台，正所谓“工欲善其事，必先利其器”。自 2002 年微软推出 C# 语言和.NET 平台以来，经过十几年的发展，现在已经有越来越多的程序员开始利用 C# 语言和.NET 平台来开发各种应用软件。

作为一个软件开发平台，.NET 框架提供了一个庞大的类库，该类库以面向对象的方式封装了各种 Windows API 函数，通过它程序员可以高效地开发各种应用软件，从而摆脱了“编程语言＋Win32 API 函数”的低效软件开发模式。在.NET 框架类库中，有两种非常重要的技术，那就是 ADO.NET 和 ASP.NET，前者是数据访问平台，后者是 Web 开发平台，它们为开发数据库程序和 Web 应用程序提供了强有力的支持。另外，利用.NET 类库开发的程序将被编译成 MSIL(微软中间语言)代码，并需要在.NET 框架中的托管平台 CLR(公共语言运行库)上运行，CLR 将为其提供安全保障和垃圾回收等功能。

C# 语言是一种优雅的编程语言，它汲取了目前几种主流编程语言如 C++、Java 和 Visual Basic 的精华，拥有语法简洁、面向对象、类型安全和垃圾回收等现代语言的诸多特征，从而成为.NET 平台下的最佳编程利器。

本书是一本既详细讲解 C# 语法，又介绍如何利用 C# 开发三层架构应用项目的教材。本书使用 Visual Studio.NET 开发 Windows 应用程序，使读者掌握 Windows 窗体和控件的使用、自定义用户控件以及 Windows 应用程序的部署等。本书通过示范项目——图书借阅管理系统中的 Windows 的开发与管理，使读者经历一次 Windows 应用系统开发的全过程，并进行一次综合性训练，从而具备 Windows 应用程序开发的经验和基本能力。

本书包含了大量的示例性代码以验证书中介绍的知识，提升读者对 C# 语言的理解能力，并能编写真正的代码来解决实际的问题。

本书共 8 章，1 个附录。

第 1 章：介绍 C# 语言的开发环境和运行环境，以及 C# 应用程序的类型。

第 2 章：在 DOS 环境下通过示例介绍 C# 语言的数据类型、运算符和表达式。

第 3 章：在 Windows 视窗环境下通过示例介绍 C# 语言的流程控制语句。

第 4 章：综合使用 DOS 和 Windows 视窗环境介绍 C# 语言面向对象的编程技术。

第 5 章：介绍图书借阅管理系统的功能、数据库设计以及系统三层架构的搭建。

第 6 章：在 DOS 环境下通过示例介绍 C# 语言的异常处理。

第7章：通过Windows应用程序示例介绍利用ADO.NET开发数据库应用的方法。

第8章：介绍图书借阅管理系统的窗体设计与功能实现。

附录：C#应用系统开发实训。

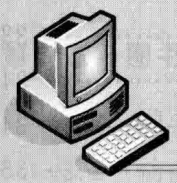
本教材的总学时为40～70学时，实验时数为15～30学时；C#应用系统开发实训可在课程结束后集中安排2～3周进行。

本书李莹、田林琳担任主编，吴晓艳、杨玥、王岩、田丹担任副主编。参加编写的还有郝雪燕、王羚伊。

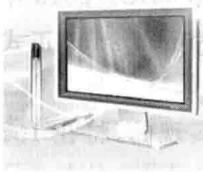
由于编者水平有限，书中难免有不足之处，敬请广大读者批评、指正。编者的E-mail是liying0000@sohu.com。

编　者

2018年1月



目 录



第1章 概 述

1.1 C#语言概述

C#(读作C sharp)语言是从C语言和C++语言发展而来的,它是一种简单的、功能强大的、类型安全的和面向对象的高级程序设计语言。C#凭借在许多方面的创新,在保持C语言风格的表现力和雅致特征的同时,实现了应用程序的快速开发。

Visual C#是Microsoft对C#语言的实现。Microsoft Visual Studio通过功能齐全的代码编辑器、编译器、项目模板、设计器、代码向导、功能强大且易用的调试器以及其他工具,实现了对Visual C#的支持。通过.NET框架类库,可以访问许多操作系统服务和其他有用的精心设计的类,这些类可显著加快开发周期。

1.1.1 Microsoft Visual Studio简介

Microsoft Visual Studio(简称VS)是一个基本完整的开发工具集,它包括了整个软件生命周期中所需要的大部分工具,如UML工具、代码管控工具、集成开发环境(IDE)等。VS的目标代码适用于微软支持的所有平台,包括Microsoft Windows、Windows Mobile、Windows CE、.NET框架、.NET Compact框架和Microsoft Silverlight及Windows Phone。

1997年,微软发布了Visual Studio 97,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,微软发布了Visual Studio 6.0。2002年,随着.NET口号的提出与Windows XP/Office XP的发布,微软发布了Visual Studio .NET(内部版本号为7.0)。与此同时,微软引入了建立在.NET框架上(版本1.0)的托管代码机制以及一门新的语言C#。2003年,微软对Visual Studio 2002进行了部分修订,以Visual Studio 2003的名义发布(内部版本号为7.1)。Visio作为使用统一建模语言(UML)架构应用程序框架的程序被引入,同时被引入的还包括移动设备支持和企业模版,.NET框架也升级到了1.1。2005年,微软发布了Visual Studio 2005,仍然还是面向.NET框架的(版本2.0)。2007年

11月，微软发布了 Visual Studio 2008。2010 年 4 月，微软发布了 Visual Studio 2010 以及. NET Framework 4.0。2012 年 9 月，微软在西雅图发布 Visual Studio 2012。2013 年 11 月，微软发布 Visual Studio 2013。2015 年 7 月，微软发布了 Visual Studio 2015 正式版和 C# 6.0 版本。

本书的所有示例都是在 Microsoft Visual Studio 2013 开发环境中实现的，将在 1.2 节中介绍如何安装 Microsoft Visual Studio 2013 开发环境。

1.1.2 C#运行环境——.NET 框架

.NET 框架是用于代码编译和执行的集成托管环境，它管理着应用程序运行的方方面面，包括程序首次运行的编译、为程序分配内存以存储数据和指令、对应用程序授予或拒绝相应的权限、启动并管理应用程序执行，并且管理剩余内存的再分配。. NET 框架的结构如图 1.1 所示。

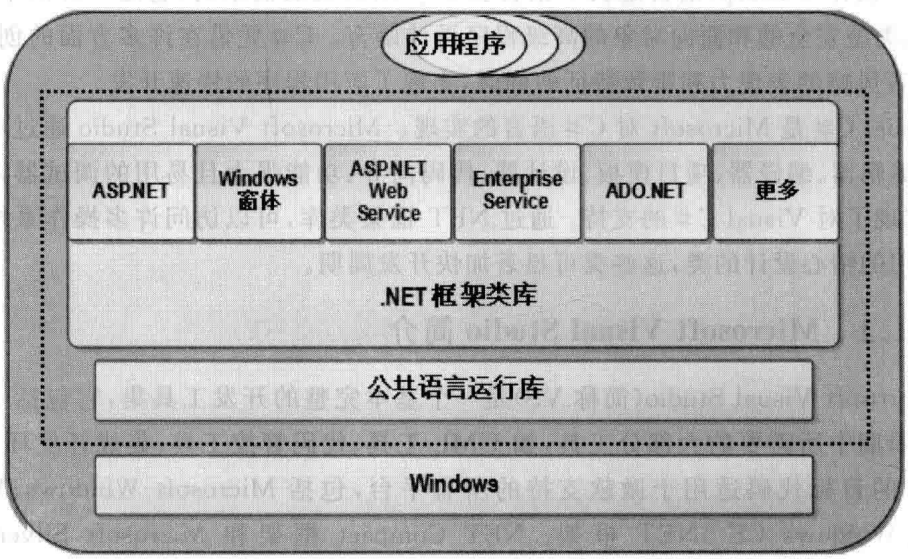


图 1.1 .NET 框架的结构

由图 1.1 可以看出，. NET 框架主要由两个组件组成：公共语言运行库(CLR)和 .NET 框架类库。

1. 公共语言运行库

CLR 可视为管理代码执行的环境，介于操作系统和应用程序之间，提供了代码编译、内存分配、线程管理以及垃圾回收之类的核心服务，主要体现在以下几方面。

(1) 管理代码的执行。各类. NET 应用程序的代码被编译为中间语言，在程序执行时，公共语言运行库将中间语言编译为机器指令，负责加载所需的元数据类型、组件及其他各种资源，并在执行过程中提供安全性机制、错误处理、垃圾回收等。

(2) 提供通用类型系统。包括值类型和引用类型两部分，这些类型为组件的资源控制、版本管理及组件间的交互提供关键信息。

(3) 提供系统服务。.NET 组件和应用程序使用公共语言运行库提供的统一接口，简化开发难度，且能够在不同的平台上移植。

2. .NET 框架类库

.NET 框架类库提供一整套通用功能的标准代码，可以供开发人员使用。类库虽然是用 C# 编写的，但是使用任何.NET 语言编写的应用程序都可以使用类库中的代码，如 C#、VB.NET、C++ 等。

.NET 框架类库的内容组织为命名空间树。命名空间是执行相关功能的类型的逻辑组织单位，每个命名空间还可以包含其他命名空间。图 1.2 给出了.NET 框架类库的一小部分。

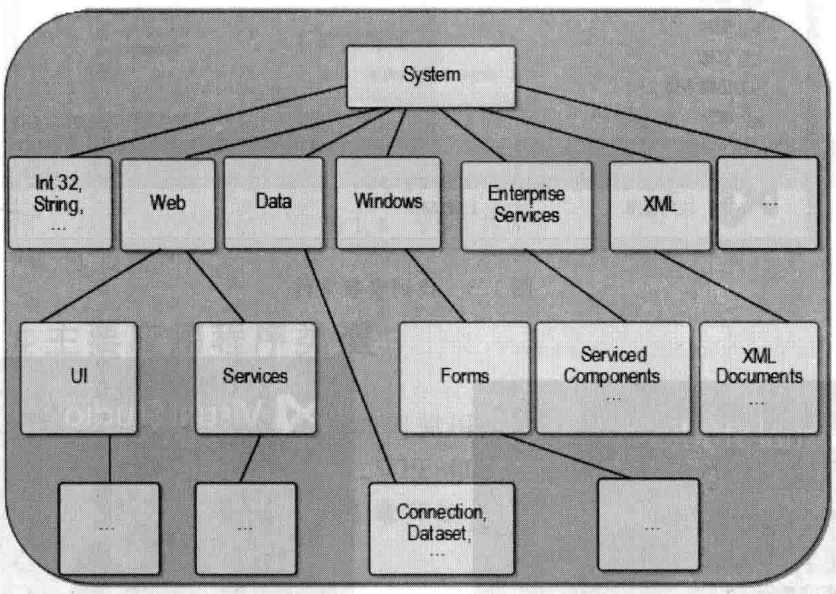


图 1.2 .NET 框架中的部分类库

类库中包含了很多命名空间，有些内容在后面的章节中有介绍，有些是后续课程的内容，感兴趣的读者可以参考与.NET 框架相关的书籍。

1.2 安装 Microsoft Visual Studio 2013

下面介绍 Microsoft Visual Studio 2013 的安装过程，该方法仅供个人学习之用。

(1) 在 VS 安装光盘中找到文件 vs_ultimate.exe，双击该可执行文件，如图 1.3 所示。

(2) 选择安装路径，这个软件占用空间非常大，最好不要装在系统盘下，会拖慢系统速度。选中“我同意许可条款和隐私策略”复选框，如图 1.4 所示。

(3) 等待创建系统还原点，如图 1.5 所示，系统将在计算机上安装组件。

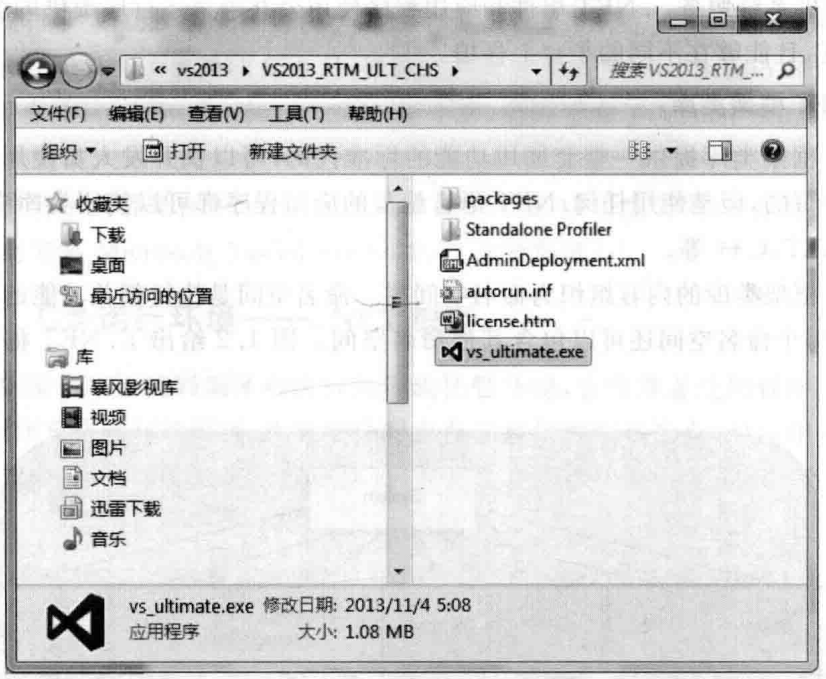


图 1.3 找到安装文件

图 1.4 安装程序界面(1)

图 1.5 安装程序界面(2)

(4) 完成安装,选中“启动”选项,进入启动页面,显示未激活,只有 30 天的试用期。如需长期使用,请在官方网站 https://www.visualstudio.com/products/how-to-buy-vs 购买正版软件。

(5) 选择“开始”→“程序”→Visual Studio 2013→Visual Studio 2013 命令,出现程序主界面,如图 1.6 所示。

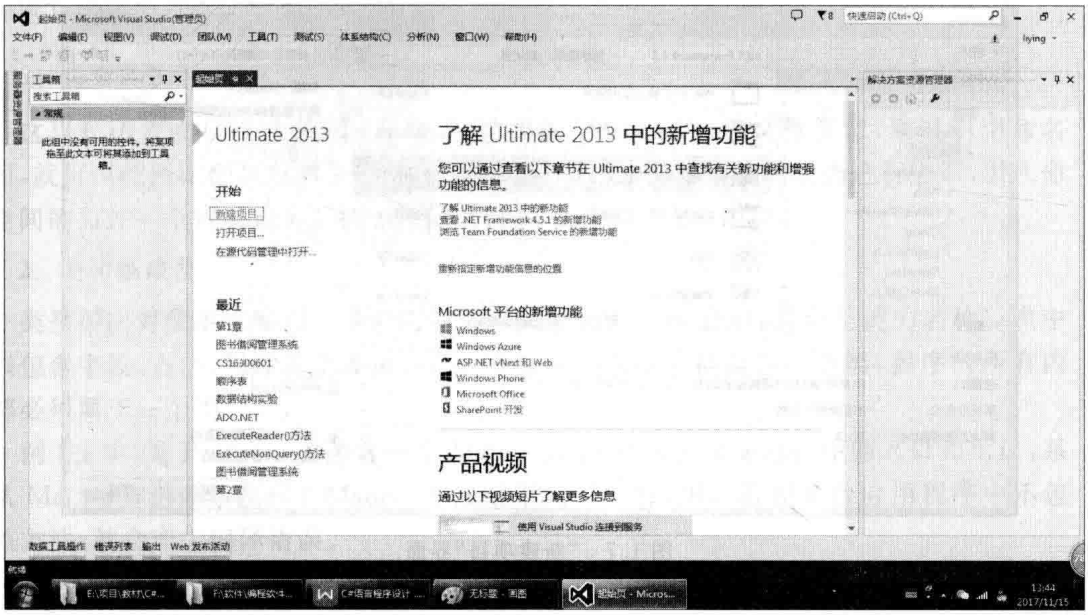


图 1.6 VS 主界面

1.3 C# 主要应用程序类型

本节从最基本的例子开始，介绍如何使用 Visual Studio 2013 创建控制台应用程序、Windows 应用程序、Web 应用程序、WPF 和 Silverlight 应用程序等。本书第 2 章中实例使用的是控制台应用程序，从第 3 章开始，大部分实例都是用 Windows 应用程序开发的。

1.3.1 控制台应用程序

控制台应用程序比较简单，一般是初学者在实践过程中需要的一个展现平台，也就是一个命令窗口。通过一些简单的程序，将一些数组、字符串等打印在控制台(一个黑色的命令窗口)上。

【例 1-1】 在控制台窗口中输出“使用 C# 语言开发第一个控制台应用程序”。

启动 Visual Studio 2013，选择“文件”→“新建”→“项目”→“新建项目”命令，弹出如图 1.7 所示的界面。

选择“模板”→Visual C#→“控制台应用程序”选项，然后为所创建的解决方案和项目命名并选择所要存放的路径。“名称”和“位置”这两个文本框虽然都有默认的名字，最好还是改成自定义的。特别是“位置”，系统默认的文件夹是 VS 系统文件夹，文件夹层数多，不便记忆，创建的项目文件不便查找，所以最好自己定义。

单击“确定”按钮，系统将建立一个控制台应用程序项目 1-1，并进入 VS 系统，系统将自动创建一个 C# 文件 Program.cs，在其中输入以下程序代码。

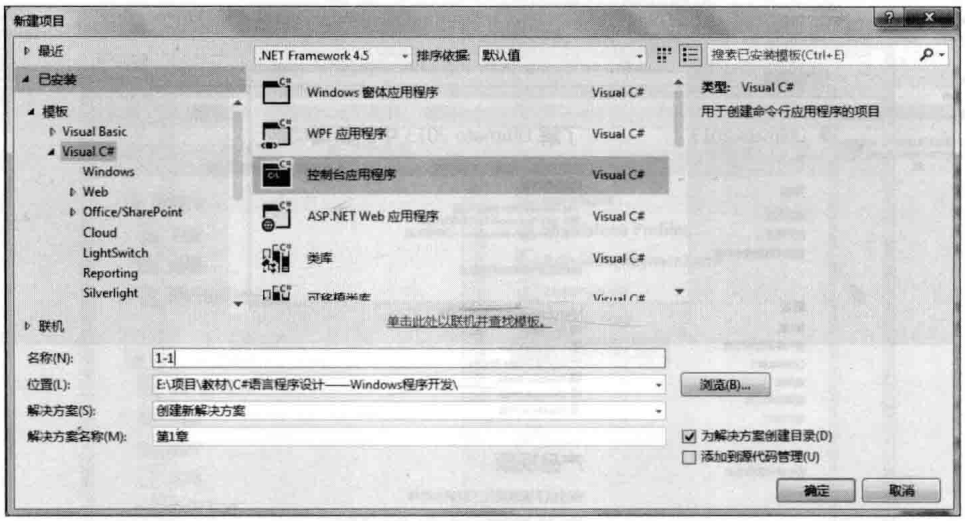


图 1.7 “新建项目”界面

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

namespace _1_1
{
    class Program
    {
        static void Main(string[] args)
        {
            //在控制台窗体输出信息
            Console.WriteLine("使用 C#语言开发第一个控制台应用程序");
        }
    }
}
```

图 1.8 例 1-1 运行结果

按 Ctrl+F5 键运行程序，或者选择“调试”→“开始执行(不调试)”命令，运行结果如图 1.8 所示。

注意：C# 程序代码是区分大小写的。

由例 1-1 可以看出，C# 程序主要由以下几部分组成。

1. 命名空间

代码的最前面是以 using 关键字开始的命名空间导入语句，使用 using 关键字用于导

入.NET 框架类库中的资源。

```
using 命名空间;
```

这与 C 语言或 C++ 中使用 #include 之类的语句来导入源文件类似，通过引用命名空间，就可以使用该命名空间下的类了。命名空间可以使元素按照功能进行组织，引入命名空间的另外一个目的是为了防止相同名称的不同标识发生冲突。

2. 类和类成员

类是面向对象语言编程中最常用的基本元素，用 class 表示，其后是类的名称。类中可以包含字段、方法、属性等类成员。C# 的每一个程序包括至少一个类，程序的所有内容都必须属于一个类。

例 1-1 中，类 Program 中包含一个 Main()方法，该方法是应用程序的入口和出口，系统从 Main()方法开始执行。Main()方法结束，程序就结束。若控制台应用程序中不包含该方法，就会产生编译错误。

3. 注释

注释也是程序的一个重要组成部分，适当地在程序中加入注释可以增强程序的可读性，以方便维护人员的维护。即便是对于程序员本人，注释对调试程序和编写程序也能起到很好的帮助作用。被标注为注释的内容在编译时会被忽略，所以不会对程序产生任何影响。

C# 中的注释有两种形式：单行注释和多行注释。使用//表示为单行注释，作用范围是直到该行结束。使用/*...*/表示多行注释，位于/*和*/之间的内容均为注释内容，例如：

```
static void Main(string[] args)
{
    /*第一个程序
      在控制台窗体输出欢迎信息*/
        Console.WriteLine("欢迎来到 C#世界");
}
```

注意：多行注释不能嵌套使用。

4. 输入/输出

例 1-1 中输出的功能是通过 System 命名空间中的 Console(控制台)类中的 WriteLine()方法完成的。Console 类表示控制台应用程序的标准输入流、输出流，控制台应用程序经常需要进行输入/输出操作。Console 类中的常用方法如下。

(1) Console.Write()：将指定字符串的值写入标准输出流。

(2) Console.WriteLine()：使用指定格式信息将指定的对象数组(后跟当前行终止符)的文本表示形式写入标准输出流。

(3) Console.Read()：从标准输入流读取一个字符。

(4) Console.ReadKey()：获取用户按下的字符或功能键。