普通高等教育“十一五”国家级规划教材

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C#应用程序设计教程（第2版）

C# Programming (2nd Edition)

耿肇英 周真真 耿燚 编著

- 用C#介绍面向对象程序设计概念
- 实例教学，案例短小精悍
- 重点介绍WPF控件

精品系列

人民邮电出版社
POSTS & TELECOM PRESS

普通高等教育“十一五”国家级规划教材

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C#应用程序设计教程（第2版）

C# Programming (2nd Edition)

耿肇英 周真真 耿燚 编著

精品系列

人 民 邮 电 出 版 社

北 京

图书在版编目（CIP）数据

C#应用程序设计教程 / 耿肇英，周真真，耿燚编著
-- 2版. -- 北京 : 人民邮电出版社，2010.11
21世纪高等学校计算机规划教材
ISBN 978-7-115-23527-5

Ⅰ. ①C… Ⅱ. ①耿… ②周… ③耿… 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(2010)第173370号

内 容 提 要

本书介绍 C#客户端应用程序设计技术，内容包括：C#语言基础、WPF 和 WinForm 应用程序开发、图形和图像编程、文件读/写和管理、多线程应用、Socket 编程、ADO.NET 数据库应用程序设计等。本书采用实例教学法，在讲清基本知识点的基础上，尽量使用短小精悍的实例加以说明，使内容容易理解。本书使用微软免费的速成版集成开发环境。本书所有例子在.NET FrameWork 3.5 下调试通过，大部分例子使用 WPF 控件。

本书可作为高等院校“面向对象 Windows 程序设计”教材，或作为学习使用 C#语言开发应用程序的培训班教材，也适合使用 C#语言开发项目的程序员参考。

普通高等教育“十一五”国家级规划教材

21 世纪高等学校计算机规划教材

C#应用程序设计教程（第 2 版）

◆ 编　　著　耿肇英　周真真　耿　燚

责任编辑　刘　博

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号

邮编　100061　　电子函件　315@ptpress.com.cn

网址　http://www.ptpress.com.cn

北京昌平百善印刷厂印刷

◆ 开本：787×1092　1/16

印张：20.25　　2010 年 11 月第 2 版

字数：528 千字　　2010 年 11 月北京第 1 次印刷

ISBN 978-7-115-23527-5

定价：34.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223

反盗版热线：(010)67171154

出版者的话

计算机应用能力已经成为社会各行业最重要的工作要求之一，而计算机教材质量的好坏会直接影响人才素质的培养。目前，计算机教材出版市场百花争艳，品种急剧增多，要从林林总总的教材中挑选一本适合课程设置要求、满足教学实际需要的教材，难度越来越大。

人民邮电出版社作为一家以计算机、通信、电子信息类图书与教材出版为主的科技教育类出版社，在计算机教材领域已经出版了多套计算机系列教材。在各套系列教材中涌现出了一批被广大一线授课教师选用、深受广大师生好评的优秀教材。老师们希望我社能有更多的优秀教材集中地呈现在老师和读者面前，为此我社组织了这套“21世纪高等学校计算机规划教材——精品系列”。

“21世纪高等学校计算机规划教材——精品系列”具有下列特点。

（1）前期调研充分，适合实际教学需要。本套教材主要面向普通本科院校的学生编写，在内容深度、系统结构、案例选择、编写方法等方面进行了深入细致的调研，目的是在教材编写之前充分了解实际教学的需要。

（2）编写目标明确，读者对象针对性强。每一本教材在编写之前都明确了该教材的读者对象和适用范围，即明确面向的读者是计算机专业、非计算机理工类专业还是文科类专业的学生，尽量符合目前普通高等教学计算机课程的教学计划、教学大纲以及发展趋势。

（3）精选作者，保证质量。本套教材的作者，既有来自院校的一线授课老师，也有来自IT企业、科研机构等单位的资深技术人员。通过他们的合作使老师丰富的实际教学经验与技术人员丰富的实践工程经验相融合，为广大师生编写出适合目前教学实际需求、满足学校新时期人才培养模式的高质量教材。

（4）一纲多本，适应面宽。在本套教材中，我们根据目前教学的实际情况，做到“一纲多本”，即根据院校已学课程和后续课程的不同开设情况，为同一科目提供不同类型的教材。

（5）突出能力培养，适应人才市场要求。本套教材贴近市场对于计算机人才的能力要求，注重理论技术与实际应用的结合，注重实际操作和实践动手能力的培养，为学生快速适应企业实际需求做好准备。

（6）配套服务完善，共促提高。对于每一本教材，我们在教材出版的同时，都将提供完备的PPT课件，并根据需要提供书中的源程序代码、习题答案、教学大纲等内容，部分教材还将在作者的配合下，提供疑难解答、教学交流等服务。

在本套教材的策划组织过程中，我们获得了来自清华大学、北京大学、人民大学、浙江大学、吉林大学、武汉大学、哈尔滨工业大学、东南大学、四川大学、上海交通大学、西安交通大学、电子科技大学、西安电子科技大学、北京邮电大学、北京林业大学等院校老师的大力支持和帮助，同时获得了来自信息产业部电信研究院、联想、华为、中兴、同方、爱立信、摩托罗拉等企业和科研单位的领导和技术人员的积极配合。在此，人民邮电出版社向他们表示衷心的感谢。

我们相信，“21世纪高等学校计算机规划教材——精品系列”一定能够为我国高等院校计算机课程教学做出应有的贡献。同时，对于工作欠缺和不妥之处，欢迎老师和读者提出宝贵的意见和建议。

人民邮电出版社

前　言

计算机专业开设的第一门高级程序设计语言课程一般是 C 或 C++，学习高级程序设计语言基本知识，编写控制台或 DOS 应用程序。在此之后应进一步学习面向对象 Windows 应用程序设计，一方面为操作系统、数据库、计算机网络、计算机图形学等后续课程提供程序设计基础，另一方面也尽早为学生课外实践提供一个实用工具。本课程是为上述目的而设置的，是计算机导论和 C 或 C++课程的后续课程。单纯讲授面向对象理论的效果并不好，结合某个面向对象语言的方法会有更好的教学效果。本书用 C#语言介绍面向对象程序设计概念。

学习面向对象 Windows 应用程序设计应采用广泛使用又适合教学的工具和语言。考虑到 Visual C++学习难度较大，Visual Basic（以下简称 VB）虽然易学，但还要耗费很紧张的学时学习一门新语言，是得不偿失的。而 C#语言是为.NET 架框量身定做的新一代面向对象的语言，语法和 C++非常相似，但比 C++简单，使用组件编程，和 VB 一样容易使用。在第一门高级程序设计语言课程是 C 或 C++的情况下，学习 C#语言可节约很多学时，对于根本没有接触过 VB 的大学生，把 C#作为学习面向对象 Windows 应用程序设计语言是合适的。当前用 C#语言开发.NET 系统的商用软件较多，这也为学生毕业后工作打下较好基础。

.NET 系统不断在发展，从 1.0 版开始到现在的 3.5 版。3.5 版是.NET 系统最新发展成果，该系统基于 2.0 版类库和语法，增加了许多新功能，如 WPF（Windows 界面基础架构）、WCF（Windows 通信基础架构）和 WF（工作流基础架构）等。其中对程序员影响最大的应是 WPF。WPF 有许多新概念和新思想，WPF 控件和 2.0 版的 WinForm 控件在使用上有很大不同，使用 WPF 感觉似乎在学习一门较难的新语言，给学习带来不少困难。但 WPF 是采用 DirectX 技术的下一代显示系统，必定会成为设计图形界面和图形图像开发的主流。微软新推出的 Silverlight 可作为浏览器的一个插件，是 WPF 的一个子集，可在浏览器中运行真正的.NET 代码，播放动画、音频，显示 3D 图形；微软手机操作系统 Windows Mobile7 也使用 Silverlight。为使学生毕业后知识不落后，本书重点介绍 WPF 控件，简述 WinForm 控件。由于 WPF 内容较多，不可能在一学期将所有 C#程序设计内容讲完，因此本书不包括 ASP.NET 方面的内容。

本书采用实例教学法，在讲清基本知识点的基础上，尽量使用实例加以说明，因此书中包含了大量实用例子。在例子中尽量避免不相关的知识点和无关的代码，使例子代码短小精悍，容易理解。书中绝大部分例子都给出了详细的设计步骤，并对每一步所实现的代码给出详细的解释，读者可按照书中步骤完成书中例子，因此本书也可看做是学生的实验指导手册。本书所有例子代码都可以从人民邮电出版社教学服务与资源网（www.ptpedu.com.cn）上免费下载。

本课程的教学参考学时为讲课 54 学时，上机实验 36 学时；本书中的大部分习题需要上机完成，因此如有可能也可适当增加上机实验学时。如学时不够，可根据需要删减部分内容，例如，不讲述 XML 和 Socket，或者将 WCF 放到 ASP.NET 中讲授。

本书主要由耿肇英完成，周真真编写了第 1～3 章，耿燚编写了图形图像及多媒体设计的相应章节。杜伟浩、杜建文、施良之、朱晓莲、姜志敏、韩忠政、冯东等也参加了本书部分代码的编写、调试、整理，以及录入校对等工作，在此一并表示感谢。

由于时间仓促，加之水平有限，书中的缺点和不足之处在所难免，敬请读者批评指正。

编者

2010 年 7 月

目 录

第 1 章 C#语言基础

本章介绍 C#语言的基础知识，希望具有 C 语言基础的读者能够基本掌握 C#语言，并以此为基础，能够进一步学习用 C#语言编写 Windows 应用程序。当然仅靠一章的内容就完全掌握 C#语言是不可能的；如需进一步学习 C#语言，还需要阅读 C#语言的专著。

1.1 C#语言特点

Microsoft.NET FrameWork（微软.NET 架框，以下简称.NET FrameWork）是微软提出的新一代软件开发模型，C#语言是.NET FrameWork 中新一代的开发工具。C#语言是一种现代的、面向对象的语言，它简化了 C++语言在类、命名空间、方法重载和异常处理等方面的操作，摒弃了 C++的复杂性，更易使用，更少出错。它使用组件编程，和 VB 一样容易使用。C#语法和 C++、Java 语法非常相似，如果用过 C++和 Java，学习 C#语言应是比较轻松的。

用 C#语言编写的源程序，必须用 C#编译器编译为公共中间语言（Common Intermediate Language，CIL）代码，形成扩展名为.exe 或.dll 的文件。公共中间语言代码不是 CPU 可执行的机器码，在程序运行时，必须由通用语言运行环境（Common Language Runtime，CLR）中的即时编译器（Just In Time，JIT）将公共中间语言代码翻译为 CPU 可执行的机器码，由 CPU 执行。CLR 为 C#语言公共中间语言代码运行提供了一种运行时环境，C#语言的 CLR 和 Java 语言的虚拟机类似。这种执行方法使运行速度变慢，但带来其他一些好处。

（1）通用语言规范（Common Language Specification，CLS）。.NET 系统包括 C#、C++、VB、J#，它们都遵守通用语言规范。任何程序设计语言只要遵守通用语言规范，其源程序都可编译为相同的中间语言代码，由 CLR 负责执行，这样的代码叫托管代码。只要为其他操作系统编制相应的 CLR，中间语言代码也可在其他系统中运行。

（2）自动内存管理。CLR 内建垃圾收集器，当堆中实例的生命周期结束时，垃圾收集器负责收回不被使用的实例占用的内存空间。也就是说，CLR 具有自动内存管理功能。而 C 和 C++语言，用语句在堆中建立的实例，必须用语句释放实例占用的内存空间。

（3）交叉语言处理。由于任何遵守通用语言规范的程序设计语言源程序，都可编译为相同的中间语言代码，不同语言设计的组件可以互相通用，可以从其他语言定义的类派生出本语言的新类。由于中间语言代码由 CLR 负责执行，因此异常处理方法是一致的，这在调试一种语言调用另一种语言的子程序时，显得特别方便。

（4）更加安全。C#语言不支持指针，一切对内存的访问都必须通过对象的引用变量来实现，

只允许访问内存中允许访问的部分，这就防止病毒程序使用非法指针访问私有成员，也避免指针的误操作产生的错误。CLR 执行中间语言代码前，要对中间语言代码的安全性、完整性进行验证，防止病毒对中间语言代码的修改。

（5）版本支持。以前系统中的组件或动态链接库如要升级，由于这些组件或动态链接库都要在注册表中注册，因此可能带来一系列问题，例如，安装新程序时自动安装新组件替换旧组件，有可能使某些必须使用旧组件才可以运行的程序，使用新组件运行不了。在.NET 中这些组件或动态链接库不必在注册表中注册，每个程序都可以使用自带的组件或动态链接库。由于不需要在注册表中注册，软件的安装也变得容易了，一般将运行程序及库文件复制到指定文件夹中就可以了。

（6）完全面向对象。C++语言既支持面向过程程序设计，又支持面向对象程序设计，而 C#语言是完全面向对象的。在 C#中不再存在全局函数、全局变量，所有的函数、变量和常量都必须定义在类中，避免了命名冲突。C#语言不支持多重继承。

1.2 编写控制台应用程序

在 DOS 操作系统中运行的程序被称做 DOS 程序。在 Windows 2000 及以后的操作系统中，“命令提示符”程序类似 DOS 操作系统界面，是 Windows 操作系统的一个任务，在这个任务中运行的程序被称做控制台应用程序。由于 DOS 程序也可以在这种方式下运行，因此也被称做控制台应用程序。下面是一个最简单的控制台应用程序，程序首先让用户通过键盘输入自己的名字，然后程序在屏幕上输出一条欢迎信息。程序的代码如下。

```
using System;                    //导入命名空间，参见 1.21 节。//为 C#语言新增注释方法，注释到本行结束
class Welcome                    //声明了一个类，类的名字叫做 Welcome，类的概念见 1.3 节
{  /*解释开始，和 C 语言解释用法相同，解释可以有多行。
   和 C 语言相同，C#语言是区分大小写的。解释结束*/
   static void Main()                    //主程序，程序入口函数，主程序必须在类中定义，必须是静态的
   {   Console.WriteLine("请输入你的姓名：");      //显示器输出字符串
       Console.ReadLine();               //从键盘读入数据，输入回车结束
       Console.WriteLine("欢迎！");      //等价于 System.Console.WriteLine("欢迎！");
   }                                     //Console 类在 System 命名空间中
}                                        //WriteLine()和 ReadLine()是 Console 类方法，参见 1.3.1 节
```

【例 1.1】 可使用 Microsoft.Net Framework SDK 内置的 C#编译器 csc.exe 或集成开发环境编译以上程序。本例介绍使用微软集成开发环境 Visual C# 2008 Express Edition（以后简称 VS2008）生成控制台程序的步骤。VS2008 是微软推出的免费版本，适合初学者学习使用 C#语言编写 Window 应用程序。具体步骤如下。

（1）运行 VS2008 程序，单击菜单“文件（F）|新建项目（P）…”菜单项，打开“新建项目”对话框如图 1.1 所示。在“模板（T）”列表框中选择“控制台应用程序”，在“名称（N）”编辑框中输入 e1_1，单击“确定”按钮，创建项目。

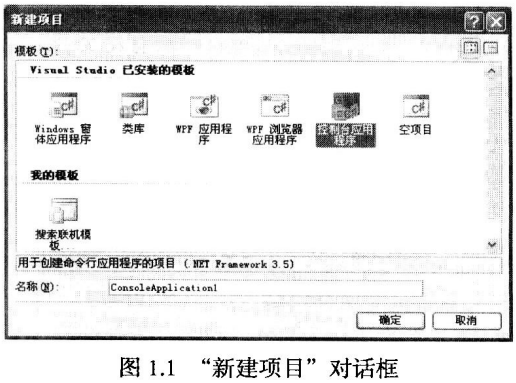

图 1.1 “新建项目”对话框

出现类似如图 1.2 所示界面（解决方案窗体未打开，main 方法无实现代码），按图中那样修改 Program.cs 文件。编写一个应用程序可能包含多个文件，才能生成可执行文件，所有这些文件的集合叫做一个项目。项目名称可以是任何标识符，这里是例子编号。

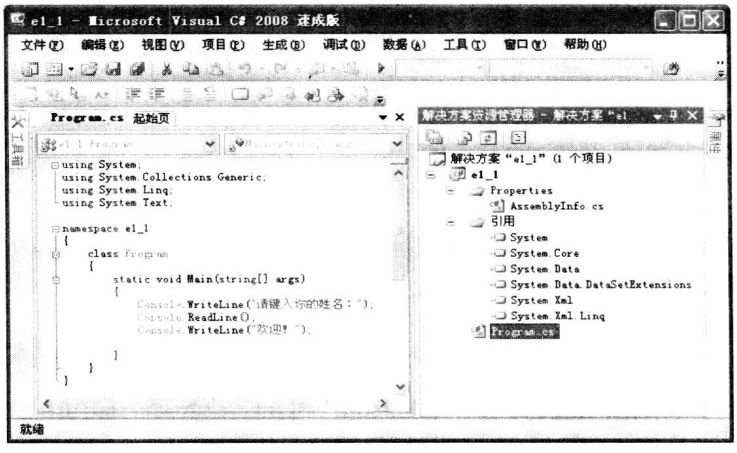

图 1.2　VS2008 集成环境界面

（2）按 CTRL+F5 组合键，运行程序，运行结果如图 1.3 所示。屏幕上出现一行字符："请输入你的姓名:"，提示输入姓名。输入任意字符并按回车键，屏幕将打印出欢迎信息："欢迎!"。输入回车退出程序。需要注意的是，和以往使用过的绝大多数编译器不同，C#编译器只执行编译过程，不经过链接直接生成扩展名为.exe 的可执行文件或扩展名为.dll 的动态链接库，C#编译器中不包含链接器。

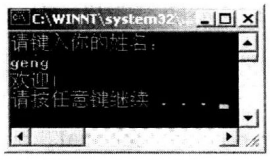

图 1.3　运行结果

（3）单击菜单"文件（F）|全部保存（L）"菜单项，打开"保存项目"对话框，在"名称（N）"编辑框中输入保存的项目文件名称，在"解决方案名称（M）"编辑框中输入保存的解决方案名称，这里这两项都不做修改，即为 e1_1。单击"位置（L）"编辑框后的"浏览（B）"按钮，选择保存的文件的路径，在该路径下将创建文件夹 e1_1，项目的所有文件都保存到这个文件夹下。这里请注意，必须保存项目的所有文件，将来才能重新打开项目。如仅保存 Program.cs，将不能重新打开项目。

1.3　类的基本概念

C#语言是一种现代的、面向对象的语言。面向对象程序设计方法提出了一个全新的概念——类，它的主要思想是将数据（数据成员）及处理这些数据的相应方法（函数成员）封装到类中，类的实例则称为对象。这就是人们常说的封装性。

1.3.1　类的基本概念

类可以认为是对 C 语言结构的扩充，它和 C 语言结构最大的不同是：类中不但可以包括数据，还包括处理这些数据的函数。类是对数据和处理数据的方法（函数）的封装。类是对一些具有相同特性和行为的事物的描述。下面是定义一个描述个人情况的类 Person 的例子。

```
using System;
class Person                              //类的定义，class是关键字，表示定义一个类，Person是类名
{   private string name="张三";                    //类的数据成员声明，字符串类型（string）数据
    private int age=12;                           //private表示私有数据成员
    public void Display()                         //类的方法（函数）声明，显示姓名和年龄
    {   Console.WriteLine("姓名:{0},年龄: {1}",name,age); }
    public void SetName(string PersonName)  //修改姓名的方法（函数）
    {   name=PersonName;     }
    public void SetAge(int PersonAge)         //public表示公有函数成员
    {   age=PersonAge;       }
}
```

Console.WriteLine（"姓名:{0}，年龄：{1}", name, age）的意义是将第二个参数变量name的值变为字符串填到{0}位置，将第三个参数变量age的值变为字符串填到{1}位置，将第一个参数表示的字符串在显示器上输出。

值得注意的是，定义Person类实际上定义了一个新的数据类型，为自定义数据类型，是对某一类人的特性和行为的描述，它的类型名为Person，和int、char等一样为一种数据类型。用定义新数据类型Person类的方法把数据和处理数据的函数封装起来。Person类是对某一类人情况和行为的描述，而不是一个具体的人，不代表张三，也不代表李四，Person类用来描述抽象的人。只有生成Person类的实例，才能代表具体的人。

类的声明格式如下：

附加声明 类修饰符 class 类名{类体}

其中，关键字class、类名和类体是必需的，其他项是可选项。类修饰符包括new、public、protected、internal、private、abstract和sealed，这些类修饰符以后介绍。类体用于定义类的成员。

1.3.2 类成员的存取控制

一般希望类中一些数据不被随意修改，只能按指定方法修改，即隐蔽一些数据。同样，一些函数也不希望被其他类的程序调用，只能在类内部使用。解决这个问题的方法是使用访问权限控制字，常用的访问权限控制字为private（私有）和public（公有）。在数据成员或函数成员前增加访问权限控制字，可以指定该数据成员或函数成员的访问权限。

私有数据成员只能被类内部的函数使用和修改，私有函数成员只能被类内部的其他函数调用。类的公有函数成员可以被类的外部程序调用，类的公有数据成员可以被类的外部程序直接使用修改。公有函数实际是一个类和外部通信的接口，外部函数通过调用公有函数，按照预先设定好的算法修改类的私有成员，外部函数只要会使用公有函数，也就掌握了类的使用方法；即使类的公有函数优化了修改类内部的私有成员算法，只要保证公有函数声明不变，就能保证类的使用方法不变。对于上述例子，name和age是私有数据成员，只能通过公有函数SetName()和SetAge()修改，即它们只能按指定方法修改。类成员默认为private。

这里再一次解释一下封装，它有两个意义：第一是把数据和处理数据的方法同时定义在类中；第二是用访问权限控制字使数据隐蔽。

1.3.3 类的对象

Person类仅是一个自定义的新数据类型，用来描述抽象的某一类人，由它可生成Person类的实例，C#语言叫对象，用来代表某一个具体的人。用如下方法声明类的对象。

```
Person OnePerson=new Person();
```

此语句的意义是建立 Person 类对象，变量 OnePerson 是对 Person 类对象的引用。也可以分两步创建 Person 类的对象。

```
Person OnePerson;
OnePerson=new Person();
```

OnePerson 虽然用地址引用 Person 类对象，但不是 C 语言中的指针，不能像指针那样进行加减运算，也不能转换为其他类型地址，它是引用类型变量，只能引用（代表）Person 对象，具体意义参见以后章节。和 C、C++不同，C#只能用此种方法生成类对象。外部程序用 OnePerson.公有方法名和 OnePerson.公有数据成员名访问对象的成员（注意 C#语言中不使用 C++语言中的->符号引用对象），例如 OnePerson.Display()，公有数据成员也可以这样访问。

1.3.4　类的构造函数、构造函数重载和析构函数

在建立类的对象时，需做一些初始化工作，例如对数据成员初始化。这些工作可用构造函数来完成。每当生成类的对象时，自动调用类的构造函数。可把初始化工作放到构造函数中完成。构造函数和类名相同，没有返回值。例如可定义 Person 类的构造函数如下。

```
public Person(string Name,int Age)     //类的有参数构造函数，函数名和类同名，无返回值
{    name=Name;            age=Age;              }
```

当用 Person OnePerson=new Person（“张五”，20）语句生成 Person 类对象时，将自动调用以上构造函数。请注意如何把参数传递给构造函数。

在 C#语言中，同一个类中的函数，如果函数名相同，而参数的类型或个数不同，被认为是不同的函数，这叫函数重载。仅返回值不同，不能看做不同的函数。这样，可以在类定义中，定义多个构造函数，名字相同，参数类型或个数不同。根据生成类的对象方法不同，调用不同的构造函数。例如，可以定义 Person 类没有参数的构造函数如下。

```
public Person()                        //类的无参数构造函数，函数名和类同名，无返回值
{    name="张三";          age=12;               }
```

用语句 Person OnePerson=new Person（"李四",30）生成对象时，将调用有参数的构造函数，而用语句 Person OnePerson=new Person()生成对象时，调用无参数的构造函数。

变量和类的对象都有生命周期；生命周期结束，这些变量和对象就要被撤销。类的对象被撤销时，将自动调用析构函数。一些善后工作可放在析构函数中完成。析构函数的名字为～类名，无返回类型，也无参数。Person 类的析构函数为～Person()。C#中类的析构函数不能被自己编写的代码调用，当垃圾收集器撤销不被使用的对象时，自动调用不被使用对象的析构函数。由于析构函数无参数，因此，析构函数不能重载。

1.3.5　使用 Person 类的完整的例子

【例 1.2】　下边用一个完整的例子说明 Person 类的定义及使用。

```
using System;
namespace e1_2                                          //定义以下代码所属命名空间，意义见以后章节
{    class Person
     {    private string name="张三";                   //类的数据成员声明
          private int age=12;
          public void Display()                         //类的方法（函数）声明，显示姓名和年龄
          {    Console.WriteLine("姓名:{0},年龄: {1}",name,age);          }
```