



河南省高等学校计算机教育研究会统编教材

C# 程序设计习题解答与上机指导

刘克成 张凌晓 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

河南省高等学校计算机教育研究会统编教材

C#程序设计习题解答与上机指导

刘克成 张凌晓 主 编

图例 (CIP) 目录头在并图

C#程序设计习题解答与上机指导 刘克成 张凌晓 主编

北京：中国铁道出版社，2003.3

河南省高等学校计算机教育研究会统编教材

ISBN 978-7-113-07180-3

I. C... II. ①刘... ②张... III. C 语言... IV. TP312

计算机—程序设计—习题—解答—参考资料

中国版本图书馆CIP数据核字 (2003) 第 084418 号

并 图：C#程序设计习题解答与上机指导

主 编：刘克成 张凌晓

出版发行：中国铁道出版社 (100044 北京宣武门大街41号)

策 划：张凌晓

责任编辑：王 雷

封面设计：王 雷

版式设计：王 雷

印 刷：北京中德印刷有限公司

开 本：787×1092 1/16 印张：13.25 字 数：303 千

版 次：2003年2月第1版

书 号：ISBN 978-7-113-07180-8/TP·3145

定 价：18.00 元

图例 (CIP) 目录头在并图

中国铁道出版社

CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是《C#程序设计》一书的配套教学参考书，内容全面、覆盖面广，旨在加深学生对教材内容的理解、提高学生的程序设计技巧、积累学生的程序调试经验、培养学生的动手能力。全书内容包括《C#程序设计》一书的思考与练习解答、上机实验项目、实验参考答案以及4个综合实例。

本书突出了《C#程序设计》的知识重点，与理论教学紧密配合，适合作为学习《C#程序设计》的配套教材，同时也可作为上机实验的独立教材。

图书在版编目(CIP)数据

C#程序设计习题解答与上机指导/刘克成, 张凌晓主
编. —北京: 中国铁道出版社, 2007. 5
河南省高等学校计算机教育研究会统编教材
ISBN 978-7-113-07780-8
I. C… II. ①刘… ②张… 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(2007)第064416号

书 名: C#程序设计习题解答与上机指导
作 者: 刘克成 张凌晓 等
出版发行: 中国铁道出版社(100054, 北京市宣武区右安门西街8号)
策划编辑: 严晓舟 秦绪好
责任编辑: 杨 勇 王雪飞
封面制作: 白 雪
责任校对: 王 欣
印 刷: 北京市彩桥印刷有限责任公司
开 本: 787×1092 1/16 印张: 13.25 字数: 303千
版 本: 2007年5月第1版 2007年5月第1次印刷
书 号: ISBN 978-7-113-07780-8/TP·2145
定 价: 18.00元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签, 无标签者不得销售

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

河南省高等学校计算机教育研究会

主任委员：段银田

副主任委员：甘 勇 普杰信 王贺明

秘 书：李学相

委 员：段银田 甘 勇 普杰信 王贺明 李学相

翁 梅 曲宏山 郭清溥 申石磊 周清雷

刘克成 陆桂明 程万里 马占欣 陈 涛

张东升 朱国华 李 敏 黄贻彬 商信华

连卫民 杨立峰 商其坤

序

PREFACE

自 20 世纪 80 年代初到 21 世纪初的 20 年间, 计算机奇迹般地展示出它惊人的运算速度、海量的存储能力和神奇的创造性, 使人类社会深深地感受到了计算机的存在和它的不可或缺性。在这种背景下, 全国各类高等学校已陆续开展了计算机基础教育, 普及了计算机文化基础知识和技术基础知识。相应地, 这两种类型的教材也大量涌现, 为计算机教育和应用的普及提供了丰富的智力资源。

然而, 进入 21 世纪以来, 高等学校的计算机基础教育面临着新的挑战。首先, 一个时期以来, 信息技术自身愈来愈向技术多元化的方向发展。网络、数据库、多媒体等技术已从科学的殿堂里走了出来, 并日益得到应用和普及, 各种信息技术在工程中的综合应用程度越来越高, 这一切促使全社会计算机的应用水平提升到了一个新高度。应用的普及也推动了需求的进一步多样化, 社会也因此更加迫切需要实用型信息技术人才。在这种背景下, 大学现行计算机基础教育教材已远远不能适应技术发展和应用的要求。其次, 由于近年来中小学信息技术教育的普遍开展, 使得原本在大学要完成的信息技术学习任务的一部分已经提前完成, 因此也需要调整当前高等学校计算机应用基础教学的内容, 可见更新现行教材已成为当前一项十分紧迫的任务。作为高等学校计算机基础教育教材改革创作的尝试, 河南省高等学校计算机教育研究会与中国铁道出版社共同策划了这套系列教材。

本套教材的创作以社会对信息技术的应用需求为目标, 学习的方向应瞄准应用, 学习的目的是能够做事的观念。要知道, 仅能在操作层面上使用计算机并不是真正意义上的应用, 开发才是真正的应用, 也就是常说的开发应用, 这也就是大学生学习信息技术的方向和应采取的行动。这些观念应逐步成为教材创作的指导思想。

突出信息技术教育的目的性是本系列丛书内容的最大特色。信息技术何其多! 究竟学什么、写什么? 要改变那种无的放矢的、包罗万象的教材创作模式; 要有目的地去写过程, 摒弃那种遍历知识过程就是一切、没有目的、文字堆砌式的创作观念和方法。应当明白, 学习信息技术是为了做事情, 而不是为了其他目的。此外, 计算机基础教育的教材要提倡精简。要树立信息量观点, 能够释疑解惑的文字构成信息量, 可以写入教材, 不能起到释疑解惑作用的文字或冗余文字只能形成垃圾信息, 应当从教材中剔除出去。

例如, 对于操作技能类的教材来说, 完全可以按照“展示一种目的, 精讲一个案例, 完成一个练习, 创造一个作品”这 4 句话的要求来进行教材创作。对于程序设计类教材来说, 教材应逐步体现并满足从程序设计向软件设计延伸的社会需求。

在教材创作中，应努力完成相关知识的整合，这不仅是本套教材所提倡的创作特色之一，也是信息技术教育改革的出路所在。对于计算机基础教育来说，知识整合主要体现在两个方面。其一，用公用事件整合适用的信息技术。将面向社会大众所发生的信息技术应用事件用其所必需的信息技术，而不是某个领域的全部信息技术加以整合应用。将从目的到技术的逆向思维作为新一代信息技术教材创作的思维方法和行为方法。整个创作过程应按照“目的决定过程，过程决定事件，事件决定对象，对象决定技术”的思路进行。其二，信息技术与其他特定学科相互整合。这种整合开辟了信息技术与专业相结合进行教材创作的途径，更加有利于实现从目的到技术进行教材创作的思想，使特定学科的内容和信息技术实现“我中有你，你中有我”，达到更高层次的融合。这种融合有利于双方共同提高教学效率，拓宽知识领域，增加知识深度，激发创造思维。总之，本丛书的创作特色主要体现在用目的、事件、对象去整合适用的信息技术，帮助读者为了达到目的而学会利用信息技术做一些实实在在的事情。

最后，本人深知新一代计算机基础教育教材的创作远不是一蹴而就的事情，目标的实现尚需时日。序言的目的仅在于简要阐明本套教材在策划过程中提出的一些基本思想和对创作的原则要求，正确与否还须经过实践的检验。望作者和读者在创作与实践中不断斧正。

河南省高等学校计算机教育研究会理事长

段银田

C#是Microsoft公司针对.NET平台而设计的全新的软件开发语言,它完全面向对象、功能强大、安全而灵活、语法结构清晰,拥有优秀的编程开发环境和高效率的编译工具,深受广大用户的喜爱。

学习C#程序设计、上机实验是掌握知识、提高程序设计技巧和积累程序调试经验必不可少的实践环节。我们以实际应用为目的,组织经验丰富的教师,结合课堂讲授的内容和进度编写了本书,它是《C#程序设计》的配套教材,旨在帮助学生加深对教材内容的理解,培养学生的动手能力。

本书源于可视化程序设计的教学实践,集成了一线任课教师的教学经验与科研成果,具有如下特点:

(1) 内容全面。本书中的实验内容涵盖了《C#程序设计》各章节的全部知识点,与教学内容联系紧密,其中第1章给出了《C#程序设计》一书思考与练习的解答,第2章给出了上机实验内容,第3章给出了实验参考答案,第4章为综合实例。本书突出了《C#程序设计》的知识重点,有较强的针对性。

(2) 方法先进。本书选用最新软件版本Microsoft Visual Studio 2005,采取循序渐进的方式,每个实验均安排了不同难度的题目,引导学生发现和解决问题,并增加了综合实例训练,注重培养学生的创新能力和科学思维方式。

(3) 针对性强。本书针对学生的计算机水平有一定差别的特点,既注重对上机实验的要求,安排了习题解答和实验参考答案;又安排了多个提高性实验,以拓宽学生的知识面,力争使每位学生在较短的时间内都有所收获。

本书由刘克成、张凌晓担任主编。由刘克成、张凌晓、杨新锋、袁东锋、黄宪通、孙晓莹、杨彩霞、齐立磊、李亚红、李云强、王秋芬、韩义波、冯东华编写。全书总纂工作由刘克成、张凌晓负责完成。

在本书的编写过程中,承蒙河南省计算机教育研究会和中国铁道出版社的热情支持与指导,在此表示衷心感谢。同时,又参阅了大量的网上资源和其他参考文献,在此对它们的作者和提供者一并表示感谢。

由于编者水平有限,书中难免有不足之处,敬请读者批评指正。

编者

2007年3月

目录

CONTENTS

第1章 习题解答	1
思考与练习 1 C#概述	1
思考与练习 2 C#语法基础	3
思考与练习 3 C#面向对象程序设计	6
思考与练习 4 Windows 程序设计基础	11
思考与练习 5 对话框与多文档编程	14
思考与练习 6 文件操作	17
思考与练习 7 C#图形图像编程基础	17
思考与练习 8 进程和线程	20
思考与练习 9 C#网络编程基础	26
思考与练习 10 数据库编程	33
思考与练习 11 组件编程	37
思考与练习 12 .NET Web 应用	39
第2章 上机实验	50
实验 1 C#概述	50
实验 2 C#语法基础	52
实验 3 C#面向对象程序设计	57
实验 4 Windows 程序设计基础	67
实验 5 对话框与多文档编程	69
实验 6 文件操作	70
实验 7 C#图形图像编程基础	74
实验 8 进程和线程	75
实验 9 C#网络编程基础	76
实验 10 数据库编程	77
实验 11 组件编程	78
实验 12 .NET Web 应用	78
第3章 实验参考答案	80
实验 1 C#概述	80
实验 2 C#语法基础	85
实验 3 C#面向对象程序设计	90
实验 4 Windows 程序设计基础	94
实验 5 对话框与多文档编程	103
实验 6 文件操作	108
实验 7 C#图形图像编程基础	116
实验 8 进程和线程	123
实验 9 C#网络编程基础	132

实验 10	数据库编程.....	134
实验 11	组件编程.....	144
实验 12	.NET Web 应用.....	147
第 4 章 综合实例.....		153
综合实例 1	资源管理器.....	153
综合实例 2	POP 客户端程序.....	166
综合实例 3	专家信息管理系统.....	170
综合实例 4	在线考试系统.....	187
参考文献.....		199

第 1 章 习题解答

思考与练习 1 C#概述

1. 什么是 Microsoft.NET 平台？主要由哪几部分组成？

答：首先，Microsoft.NET 平台是一个开发平台，它定义了一种公用语言子集（Common Language Subset，简称 CLS），这是一种为符合其规范的语言与类库之间提供无缝集成的混合语言；其次，Microsoft.NET 统一了编程类库，提供了对下一代网络通信标准的可扩展标记语言（eXtensible Markup Language，XML）的完全支持；再者，Microsoft.NET 还实现了人机交互方面的革命，如在软件中添加手写和语言识别等功能，增加了对各种用户终端的支持能力。总之，Microsoft.NET 是一种面向网络支持各种用户终端的开发平台环境。Microsoft.NET 平台主要由 5 个部分组成，分别是 Windows.NET、.NET Enterprise Servers（.NET 企业级服务器）、.NET Framework（.NET 框架）、Microsoft Visual Studio.NET 和 .NET Web 服务构件。

2. C#语言与 C/C++语言比较起来有哪些优点？

答：C#语言与 C/C++语言比较起来有以下优点：

（1）简洁的语法。在 .NET 平台上，C#代码不允许直接对内存操作，它所带来的最大特色是没了指针。在 C#中对类和方法的引用操作只用一个简单的“.”运算符实现。

（2）精心的面向对象设计。在 C#类型系统中，每种类型都可以看作一个对象，C#提供装箱和拆箱机制来完成对象操作。C#只允许单继承，从根本上避免了类型定义的混乱，而且 C#没有全局函数，当然也没有全局变量。任何对象都必须封装在类中。这使得代码具有更好的可读性，而且也减少了因命名而带来的冲突。

（3）与 Web 结合紧密。在 C#中通过 SOAP 实现应用程序的解决方案与 Web 标准相统一，使之能与 HTML 和 XML 语言建立联系。

（4）完善的安全性及纠错处理能力。在 C#的设计中，可以消除软件开发过程中许多常见的错误，并提供包括类型安全在内的完整的安全机制。

（5）版本控制技术。C#语言内置了版本控制功能。

（6）较高的灵活性和兼容性。C#允许将某些类（方法）声明为非安全性的，这样便可以使用指针、结构和静态数组，并且这些非安全性的代码不会带来任何问题。另外提供“委托（Delegates）”功能来模拟指针。

3. Visual C#.NET 怎样把多个项目组织起来？

答：“解决方案资源管理器”窗口用于管理和监控方案中的项目。一个解决方案中可包含多个项目，一个项目又包含了多个文件，该窗口采用树形视图显示方案及方案中项目的层次结构，可以使用户能够方便地组织需要开发和设计的项目和文件，以及配置应用程序或组件。

4. Visual C#.NET 集成开发环境有哪些常用窗口？各有什么作用？如何使用？

答：Visual C#.NET 集成开发环境的常用窗口有以下几个：

(1)“起始页”窗口:打开.NET 集成开发环境 (IDE) 后出现的第一个窗口,用于设置首选项、阅读产品新闻和访问其他信息,便于更有效地利用 Visual Studio。

(2)“解决方案资源管理器”窗口:采用树形视图显示方案及方案中项目的层次结构,可以使用户能够方便地组织需要开发和设计的项目和文件,以及配置应用程序或组件。

(3)“类视图”窗口:以树形结构显示了当前项目中的所有类,并在每个类中列出了成员变量和成员函数。

(4)“属性”窗口:使用“属性”窗口可以编辑和查看控件、项目或类的特征属性,还能在设计时查看和修改窗体设计器中被选对象的属性和事件。

(5)“工具箱”窗口:包含了许多可用控件,使用户在应用程序中方便快捷地使用这些控件。

(6)“窗体设计器”窗口:用来设计应用程序的界面。

(7)“代码编辑器”窗口:在该窗口中,可以对应用程序的源代码、HTML 页、CSS 表单进行编辑和开发工作。

(8)“服务器资源管理器”窗口:它是 Visual Studio 2005 中的服务器控制台。通过它可以在 IDE 中集中监控和管理一个应用方案所涉及的所有“服务器”和“数据连接”。

(各窗口的使用方法略。)

5. 编程及上机调试。

(1) 创建一个控制台应用程序,在 Main 方法中输入以下代码。

```
Console.WriteLine("This is a ");
Console.WriteLine("C# program!");
```

运行应用程序并观察结果。

答:程序运行结果如图 1-1 所示。

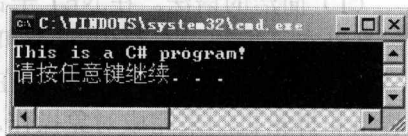


图 1-1 编程及上机调试 (1) 运行结果

(2) 创建一个 Windows 应用程序,在窗体上添加一个按钮控件。设置窗体和按钮控件的属性,如表 1-1 所示。

表 1-1 窗体和控件属性

控 件	属 性	属 性 值
Form1 窗体	Name (名称)	Test
	Text (标题)	Windows 应用程序测试
	Size (大小)	100, 50
	ForeColor (前景色)	白色
	BackColor (背景色)	蓝色
Button1 按钮	Text (标题)	请点击

然后为按钮控件添加 Click 事件,事件响应代码为:

```
MessageBox.Show("嗨! 你好!");
```

运行应用程序并观察结果。

答:程序运行后将出现如图 1-2 所示的窗口界面。当单击其上的“请点击”按钮时,将出现如图 1-3 所示的消息对话框;单击消息对话框中的“确定”按钮,消息对话框消失;单击窗口右上角的“关闭”按钮,窗口关闭。

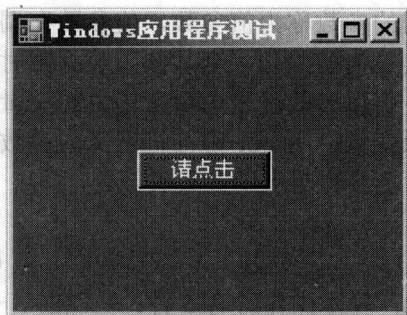


图 1-2 窗口界面

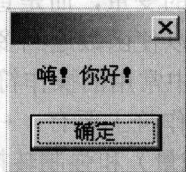


图 1-3 消息对话框

思考与练习 2 C#语法基础

1. 简述值类型数据与引用类型数据的区别。

答：值类型变量直接存储它的数据内容，当把一个值赋给一个值类型时，该值实际上被复制了；引用类型变量不存储实际数据内容，而是存储对实际数据的引用，当把一个值赋给一个引用类型时，仅仅是复制引用，实际的值仍然保留在原来的内存位置，只是赋值后有两个不同的变量指向这个实际的值。

2. 简述 C#语言中的数据类型。

答：C#的数据类型分为值类型、引用类型和指针类型 3 类。值类型包括简单类型、结构类型和枚举类型，其中简单类型分为整数类型、布尔类型、实数类型和字符类型。引用类型包括类类型、接口类型、委托类型和数组类型。指针类型只能用于不安全模式。

3. 简述 C#语言中的数据类型转换。

答：C#区分两种转换：“隐式转换”和“显式转换”。隐式转换仅适用于那些不用仔细检查即可安全实现的转换，这种转换总是会成功，而且从不会导致信息丢失，是不需要加以声明就可以进行的转换。隐式转换分为同一转换、隐式数值转换、隐式枚举转换、隐式引用转换、隐式常量符号转换和用户自定义隐式转换。显式类型转换又叫强制类型转换，可能会导致信息丢失。显式转换包括：所有的隐式转换、显式数值转换、显式枚举转换、显式引用转换、显式接口转换和用户自定义显式转换。

4. 简述 C#语言中的装箱和拆箱。

答：装箱转换允许将值类型隐式转换为引用类型，其工作过程为：首先分配一个对象实例，然后将值类型的值复制到该实例中。拆箱转换允许将引用类型显式转换为值类型，其工作过程为：首先检查对象实例是否为给定值类型的装箱值，然后将该值从实例中复制出来。

5. C#中的变量包含哪几种类型，各有什么特点？

答：在 C# 中有 7 种类型的变量：静态变量、非静态变量、数组元素、局部变量、值参数、引用参数和输出参数。用 `static` 修饰符声明的字段称为静态变量，未用 `static` 修饰符声明的字段称为非静态变量，静态变量不属于任何特定的实例，在任何时候静态变量都只有一个副本；实例变量属于某个实例，即类的每个实例中都包含该类的实例变量的一个副本；数组元素在创建数组实例时开始存在，在没有对该数组实例的引用时被撤销，数组元素的使用与实例变量相同；在语句块中声明的变量称为局部变量，它只有局部作用域，只在该

原书缺页


```

string a="yes";
IsString(a);
int a1=0;
IsString(a1);
Console.Read();
}
static void IsString(object s)
{
    if (s is string)
        Console.WriteLine("It is a string");
    else
        Console.WriteLine("It is not a string");
}
}

```

答：程序运行结果为：

```

It is a string
It is not a string

```

(2) 分析程序，写出运行结果。

```

using System;
class Program
{
    static void Main(string[] args)
    {
        object a="yes",b=5;
        string s=a as string;
        IsString(s);
        string t=b as string;
        IsString(t);
        Console.Read();
    }
    static void IsString(object a)
    {
        if (a!=null)
            Console.WriteLine("It is a string");
        else
            Console.WriteLine("It is not a string");
    }
}

```

答：程序运行结果为：

```

It is a string
It is not a string

```

(3) 分析程序，写出运行结果。

```

using System;
class Program

```

```

{
    static void Main(string[] args)
    {
        int[] data={1,2,3,4,5,6};
        foreach(int m in data)
            Console.WriteLine("{0}",m);
        Console.Read();
    }
}

```

答: 程序运行结果为:

1
2
3
4
5
6

思考与练习 3 C#面向对象程序设计

1. 简述 new 关键字的用途。

答: 在 C# 中, new 关键字有两种用途, 即可用做操作符或者修饰符。当作为操作符时用于创建一个新的类型实例, 不能重载 new 操作符。当作为类的修饰符时, 它仅允许在嵌套类声明时使用, 表明类中隐藏了由基类中继承而来的与基类中同名的成员, 它的作用主要是显式隐藏从基类继承的成员, 即若要隐藏继承的成员, 可使用相同名称在派生类中声明该成员, 并用 new 修饰符修饰它。

2. 哪种类不能被派生?

答: 密封类和静态类不能被派生。

3. 接口中包含哪些成员?

答: 接口成员可以是方法、属性、索引指示器和事件。

4. 什么是接口映射?

答: 在类中定位接口成员的实现称为接口映射。

5. 定义一个基类 c0, 类 c1 从 c0 派生, 在 c0 中定义一个虚方法, 在 c1 中实现。

答: 程序如下:

```

using System;
class c0
{
    public virtual void Print()
    {
        Console.WriteLine("Print in c0");
    }
}
class c1:c0

```

```

{
    public override void Print()
    {
        Console.WriteLine("Print in c1");
    }
}

class Test
{
    static void Main(string[] args)
    {
        c0 b=new c0();
        c1 d=new c1();
        b.Print();
        d.Print();
        b=d;
        b.Print();
        d.Print();
        Console.Read();
    }
}

```

6. 定义一个密封类，在该类中定义一个方法，并在主函数中使用该密封类的方法。

答：程序如下：

```

using System;

sealed class SealedClass
{
    public double x=0,y=0;
    public SealedClass(double a,double b)
    {
        x=a;
        y=b;
    }

    public void Display()
    {
        Console.WriteLine("x={0},y={1}",x,y);
    }

    static void Main(string[] args)
    {
        SealedClass s=new SealedClass(10,20);
        s.Display();
        Console.Read();
    }
}

```

7. 创建一个控制台程序，在程序中声明一个委托类型，再定义一个对应的方法，然后声明一个该委托类型的事件，通过事件的+=操作把一个方法预定到一个事件上。

答：程序如下：

```
using System;
public delegate void TimeEvent(string s);    //委托声明
class MyTime
{
    public event TimeEvent Timer;           //声明事件
    public void OnTimer(string s)
    {
        if(null!=Timer)    Timer(s);        //引发事件
    }
}
class ProcessTime
{
    public void GenerateTime(string s)       //事件处理
    {
        Console.WriteLine("{0}:The time is {1} now!",s,DateTime.Now);
    }
}
class TestTime
{
    static void Main(string[] args)
    {
        ProcessTime p=new ProcessTime();
        MyTime t=new MyTime();
        t.Timer+=new TimeEvent(p.GenerateTime);    //把事件与事件处理联系起来
        t.OnTimer("Peter");                        //使用事件
        Console.Read();
    }
}
```

8. 阅读程序，并分析运行结果

```
using System;
public delegate void TimeEventHandler(string s);
namespace Test
{
    class MyClass
    {
        public const int MyConst=20;
        public int MyField=40;
        public event TimeEventHandler Timer;
        public MyClass()
        {
            Console.WriteLine("Instance Constructor 1");
        }
        public MyClass(int value)
        {
            Console.WriteLine("Instance Constructor 2");
        }
    }
}
```