



普通高等教育“十一五”国家级规划教材

新世纪高职高专实用规划教材

● 计算机系列

C#面向对象 程序设计(第2版)

C# MIANXIANG DUIXIANG CHENGXU SHEJI

邵鹏鸣 编著

赠送
电子课件



清华大学出版社

普通高等教育“十一五”国家级规划教材
新世纪高职高专实用规划教材 计算机系列

C#面向对象程序设计 (第2版)

邵鹏鸣 编著

清华大学出版社

北 京

内 容 简 介

本书使用 Visual C# 2010 详细介绍了如何以 C# 语言进行面向对象编程,把有关面向对象的内容贯穿其中,使初学程序开发的读者能够逐步体会并深刻理解“对象”技术的强大功能。同时,还引入了事件的处理与 GUI 的使用等。通过对这些内容的不断学习,学生们将逐步掌握 GUI 的创建。全书用两章的篇幅通过一个具体的实际应用程序实例讲述使用 ADO.NET 与 SQL 访问数据库的编程技术。通过多个详尽的案例分折,使学生能够尽快地掌握面向对象的编程技巧。此外,书中提供了大量与开发相关的技术要点,同时配合实用、有效的 GUI 应用程序,使读者能够迅速掌握并巩固所学到的知识。本书汲取了很多来自实际编程中的体验,这将为培养读者获得良好的编程习惯打下一个坚实的基础。

本书可作为高职高专计算机专业和高等技术型院校相关专业进行 C# 编程语言教学的教材,也可供程序设计与开发人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C#面向对象程序设计/邵鹏鸣编著.--2 版.--北京:清华大学出版社,2011.9

(新世纪高职高专实用规划教材 计算机系列)

ISBN 978-7-302-26634-1

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

中国版本图书馆 CIP 数据核字(2011)第 177006 号

责任编辑:黄 飞 宋延清

封面设计:山鹰工作室

版式设计:杨玉兰

责任校对:李玉萍

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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

印 刷 者:北京密云胶印厂

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185×260 印 张:32.5 字 数:786 千字

版 次:2011 年 9 月第 2 版 印 次:2011 年 9 月第 1 次印刷

印 数:1~4000

定 价:58.00 元

产品编号:040816-01



读者回执卡

欢迎您立即填写回函

您好！感谢您购买本书，请您抽出宝贵的时间填写这份回执卡，并将此页剪下寄回我公司读者服务部。我们会在以后的工作中充分考虑您的意见和建议，并将您的信息加入公司的客户档案中，以便向您提供全程的一体化服务。您享有的权益：

- ★ 免费获得我公司的新书资料；
- ★ 寻求解答阅读中遇到的问题；

- ★ 免费参加我公司组织的技术交流会及讲座；
- ★ 可参加不定期的促销活动，免费获取赠品；

读者基本资料

姓 名 _____ 性 别 ☐男 ☐女 年 龄 _____
电 话 _____ 职 业 _____ 文化程度 _____
E-mail _____ 邮 编 _____
通讯地址 _____

请在您认可处打√ (6至10题可多选)

- | | | | | |
|-------------------|-----------------------------------|---|----------------------------------|----------------------------------|
| 1、您购买的图书名称是什么： | | | | |
| 2、您在何处购买的此书： | | | | |
| 3、您对电脑的掌握程度： | ☐ 不懂 | ☐ 基本掌握 | ☐ 熟练应用 | ☐ 精通某一领域 |
| 4、您学习此书的主要目的是： | ☐ 工作需要 | ☐ 个人爱好 | ☐ 获得证书 | |
| 5、您希望通过学习达到何种程度： | ☐ 基本掌握 | ☐ 熟练应用 | ☐ 专业水平 | |
| 6、您想学习的其他电脑知识有： | ☐ 电脑入门 | ☐ 操作系统 | ☐ 办公软件 | ☐ 多媒体设计 |
| | ☐ 编程知识 | ☐ 图像设计 | ☐ 网页设计 | ☐ 互联网知识 |
| 7、影响您购买图书的因素： | ☐ 书名 | ☐ 作者 | ☐ 出版机构 | ☐ 印刷、装帧质量 |
| | ☐ 内容简介 | ☐ 网络宣传 | ☐ 图书定价 | ☐ 书店宣传 |
| | ☐ 封面、插图及版式 | ☐ 知名作家（学者）的推荐或书评 | | ☐ 其他 |
| 8、您比较喜欢哪些形式的学习方式： | ☐ 看图书 | ☐ 上网学习 | ☐ 用教学光盘 | ☐ 参加培训班 |
| 9、您可以接受的图书的价格是： | ☐ 20 元以内 | ☐ 30 元以内 | ☐ 50 元以内 | ☐ 100 元以内 |
| 10、您从何处获知本公司产品信息： | ☐ 报纸、杂志 | ☐ 广播、电视 | ☐ 同事或朋友推荐 | ☐ 网站 |
| 11、您对本书的满意度： | ☐ 很满意 | ☐ 较满意 | ☐ 一般 | ☐ 不满意 |
| 12、对我们的建议： | | | | |

1 0 0 0 8 4

北京100084—157信箱

读者服务部

收

贴 票 邮 处

邮政编码：□□□□□□

请剪下本页填写清楚，放入信封寄回，谢谢！

第 1 版 前言

随着 21 世纪的到来, 计算机技术的发展更加迅猛, 在各行各业的应用更加广泛, 面对日新月异的新技术、新方法, 我们必须对现有计算机课程的设置和教学内容进行调整, 以适应技术进步和市场变化的需要, 使讲授的内容是市场上最需要的。

今天, 应用程序已由驻留在用户硬盘上的独立可执行文件发展为由 Web 服务器在 Internet 上传送的分布式应用程序; 相应地, 任何一种开发平台及程序设计语言都必须适应这种变化。微软的 .NET 是一种开发平台, C#(读作 C sharp)是微软特别为 .NET 平台设计的一种现代编程语言, .NET 有着广阔的应用前景, .NET 的应用必将对整个计算机产业产生重要而深刻的影响。 .NET 不但改变了开发人员开发应用程序的方式及思维方式, 而且使开发人员能创建出各种全新的应用程序, 大幅度提高软件的生产率。未来 .NET 将无处不在。

微软以 .NET 平台和 C#、Visual Basic .NET 编程语言的形式推出了与 Java 全面竞争的对手。这是微软在编程语言和编程环境这块竞技场上所取得的最出色的成果。当然, 微软具备很有利的条件: 可以看到 Java 在什么方面做得好, 在什么方面做得还不够好, 然后, 基于此去构建 C#, 使其具备 Java 所不具备的优点。

目前, 与 Java 相比, 对于 .NET 平台, 人们所关心的最重要的问题是, 微软是否会将它完全移植到其他平台上。微软宣称这没有问题, 而且 Mono 项目(www.go-mono.com)已经有了一个在 Linux 上运行的 .NET 的部分实现。

C#是一种简单的、现代的、面向对象的且类型安全的编程语言, C#语言从 C 和 C++语言演化而来, C#同时具备“快速应用程序开发(RAD)”语言的高效率和 C++固有的强大能力。同时它吸收了 Java 语言的特点和精华, 熟悉 Java 的人会觉得它很像 Java。比尔·盖茨曾说过“Java 是最卓越的程序设计语言”, 不过从 C#诞生的那一时刻起, 这已成为过去。C#是微软将 Java 集成到 .NET 中的产物, 它是整个 .NET 平台的基础, 也是未来主流的编程语言。

我们知道 C 语言是一种代码效率很高但不易于进行快速开发的程序设计语言。面向对象程序设计(Object-Oriented Programming, OOP)出现后, OOP 与 C 语言融合, 产生了 C++, 继而有了 C++集成开发环境。但 C++的出现, 并没有使 C 语言家族在应用开发方面获得突飞猛进的发展, 基本上仍然只占据着 UNIX、Linux 以及底层应用开发的天地, 而在 Windows 大型应用软件特别是数据库和 Web 开发上, 由于 C 语言编程固有的复杂性和缺乏针对性, 就不如 Visual Basic(一般可简称为 VB)等具有很强针对性的开发工具。那些使用过包括 C 和 C++在内的多种程序设计语言的人, 相信会深深地体会到它们之间的区别。比如与 Visual Basic 相比, Visual C++(一般可简称为 VC++)程序员为实现同样的功能就要花费更长的开发周期。

另一方面,虽然C和C++为我们带来了高度的灵活性,但必须忍受学习的艰苦和开发的长期性,特别对VC++来说,大部分的程序结构都被封装在MFC(微软基础类库)中。所以对于初学者来说,程序结构显得十分混乱,学习将变得十分艰苦。而且从VC++ 2.0起,为了适应不断更新的技术(例如COM、ATL等),以及为了与前一个版本兼容,VC++的每一次升级都给程序员带来一份痛苦——程序结构变得越来越复杂,而且出现了越来越多的变量类型,从而带来了更多的问题。

C#的出现弥补了C语言家族的上述不足,它借鉴了Java、C++、C甚至VB的优点,既具有C、C++那样的强大功能,又具有Java那样的面向对象机制,同时还具有类似于VB开发的高效率和方便性。

学习一种先进的语言和一种先进的编程方法将激起学生更大的兴趣。这些知识在他们离开学校,进入一个信息占据重要地位的世界时可以立即发挥作用。当今的学生必须同时掌握基础语言、面向对象编程和类库知识,而C#课程适合这类知识的学习。

很多院校都设立了软件专业,有的学校的学生,在几年的学习时间里,不仅学习了C、C++,而且也学习了Java、VB或VB.NET甚至C#,同时还学习了Web应用程序设计及其他有关软件开发的课程。

但据作者对一些院校的调查,却发现有些院校软件专业的学生在毕业找工作时,竟然很难找到一份对口的工作,所找到的工作与软件无关、甚至与计算机无关——不是企业不需要软件专业人才,而是没有物色到所需要的人才。

我认识一位学生,学习成绩不错,学习也很勤奋,在毕业前夕还通过了全国计算机技术与软件专业技术资格(水平)考试,获得了软件设计师资格。但当我交给他一份实际任务时,他却根本不知从何下手,尽管该任务很简单。这些都说明,我们的教学体系、教学内容、教学模式和方法都存在着很多需要改进的地方。

多年来,作者一直都对计算机软件专业课程的课程体系、结构、课程内容和教学模式、方法进行研究,从宏观和微观的角度把握每门课程,从市场和可持续性的角度设计每门课程。将基本技能培养与主流技术相结合,把软件工程的思想融入到教学体系中,在职业能力分析的基础上,以工作任务为中心来整合相应的知识、技能,从应用与工程实践的角度出发来设计课程体系、课程内容和教学模式。

检验成功与否的标准是“不管白猫黑猫,抓到老鼠就是好猫”,也就是说,不管你怎么说,怎么做,学生学完后,能够将知识转化为能力、能够将知识转化为应用,并具有可持续性,就是成功的。否则,难免是纸上谈兵、言之无物、空中楼阁。根据这种指导思想,作者努力将教学理念和思想转化为实际应用,对计算机软件专业主干课程的课程体系、结构、课程内容和教学模式、方法进行了改革和实践。2006年作者在清华大学出版社出版了两部教材。在机械工业出版社出版了一部教材,2007年在清华大学出版社又出版了作者项目研究中的另外两部教材,它们是作者的教学理念和思想的结晶,是作者多年来教学研究与实践的结晶。

作者反对用“空洞的例程”来说明概念、原理的教学方法。那些例子完全脱离实际,看起来很幼稚,尽管它能够说明一个概念,学生也容易看懂,上课也容易听懂,但是检查

学习好坏的标准,不是“知道还是不知道,听懂还是没有听懂,看懂还是没有看懂”,而是“会不会应用”。真实例程同样能够说明概念、原理和知识点,只不过是面对复杂的技术实践体系,需要花大量的时间和精力精心地去设计这样的例程,只要设计得好,真实例程同样能够说明概念、原理和知识点,学生同样容易看懂、听懂,并且懂得在实际应用中如何运用这些概念、原理和知识点,获得事半功倍的效果。

作者的目标很明确:写一本实用的、有一定深度且易读的程序设计语言教材。无论读者有无编程经验,要做到开卷有益,重要的是能够激发他们的学习兴趣,能够将他们所学的知识应用于实际。

本书将基本技能培养与主流技术相结合,以市场对人才的需求为依据,从应用与工程实践的角度出发进行组织和撰写,其主要特点如下。

(1) 新体系、新内容、新手段、新思路。无论是内容体系、编写的思路、教学的模式还是理念等都具有一定的新意。

(2) 是基于工作过程的教学过程的设计。在职业能力分析的基础上,以工作任务为中心来整合相应的知识、技能,从应用与工程实践的角度出发来设计课程体系、课程内容和教学模式。教学项目不是对企业项目简单地再现与复制,而是源于企业并高于企业,是在真实性基础上构建学习性的环境。

(3) 具有先进性及实用性。本书的内容反映出最新的实用程序设计方法及技术,顺应并符合新世纪教学发展的规律,书中讲授的程序设计方法与现代编程方法步调一致,具有很强的实用性。

(4) 采用以问题、模型为中心的教学方法和面向实际的应用程序驱动的教学方式,并追求理论和实践一体化。本书完全通过示例(任务)来学习。书中通过大量的完整示例来介绍相关的主题。本书不是采用传统的“提出概念→解释概念→举例说明”的方法,而是以实践为主线,以应用为目标,通过完成实际应用程序的方式来学习程序设计知识。整个学习是由许多小的教学模块和任务组成的。每个教学模块和任务首先提出一个来自实际的问题,然后学生可“尝试”解决该问题的应用程序并了解其工作过程,接着教师介绍该问题的解决方案及方法和步骤,然后归纳出一般的规律、概念及知识点。随后学生参照教师的解决方案及方法和步骤来解决该问题,最后教师对学生提出一个新的实际问题,学生应用所学的方法和步骤来解决该问题。

学生通过这样一种“实践→学习→实践→提高”的过程,不仅可以更快、更深入地理解和掌握课程的内容,而且可以提高自己的实践能力和独立解决实际问题的能力。很好地贯彻了在实践中学习、在学习中实践、理论和实践教学一体化的思想。

针对每一个新概念、知识点的提出,都伴随着一个完整的、可实际运行的实用程序及其输入、输出。学生通过完成实际应用程序,来掌握概念和知识点。

(5) 以独特的体系结构、设计编排和内容组织模式,将复杂的技术体系以简洁易懂的方式呈现给学生。教学内容在设计时贯穿了一个思想:面向实际;使学生容易理解、明白所包含的全部细节。

因此本书在安排时,循序渐进、由浅入深、综合应用。后续章节在讲授新知识点的同

时综合应用了前面章节所学的知识。这样做有利于温故知新,既巩固了前面所学的知识,又懂得了如何综合应用已学的知识来解决问题。书中很多实际应用程序示例不只是讲授一个新的知识点,而是在讲授新知识点的同时,应用了前面所学过的知识。

(6) 使学生从一开始就编写有用的程序,这有助于保持学生的学习兴趣及动力。书中的示例不是只有几行代码的小程序,这些示例来源于实际应用程序或其中的一部分,它们与我们的生活相关,或是大家都感兴趣的内容。本书包含大量这样的示例,对这些示例及其代码的分析与讨论是本书的精髓。学生可先按照所描述的方法和步骤编写代码,并运行自己所创建的应用程序,“尝试”该应用程序并了解其工作过程。然后通过代码分析和讨论来掌握其包含的概念、原理和方法。

(7) 逐步展现主题鲜明的各章。本书将集中重要的主题并进行充分的论述,而不是肤浅地涉及许多问题,罗列许多概念、术语、语法。本书的内容及体系结构使学生感到它既具有易读性又能增进知识,具有很强的实用性。

(8) 讲述面向实际的技术。学习程序设计语言的目的是为了开发程序,本书不只讲授C#的语言要素、语法,而且教会读者如何用C#开发程序,书中的每一个面向实际的应用程序示例都说明了开发过程及包含的知识点。通过学习如何开发程序来掌握语言要素和语法。

(9) 传授面向对象的程序设计技能。本书自始至终贯穿着面向对象编程的思想,目标之一是使学生习惯于现实世界的程序设计,仅仅知道面向对象的概念是不够的,学生们必须能够运用这些知识开发现实世界的程序。教学的核心概念之一是在成为一名对象设计者之前,必须首先成为一名对象用户。换句话说,在有效地设计自己的类之前,必须首先学会使用预定义的类。书中从第一个程序开始就接触如何使用类。学习使用类库中的类与编写自己的类二者相辅相成、互相促进。学习使用类库中的类有助于学习编写自己定义的类,学习自定义类又有助于学会使用类库中的一些类,并加深对类库中的一些类的理解。当今学生必须同时掌握基础语言、面向对象编程和类库,我们力争将这三者结合在一起。

任何一本真正讲授面向对象方法的教材都必须从对象出发,即所有处理过程都需要依照面向对象的术语来进行讨论。然而,这并不意味着学生看到的第一个程序就涉及多个类及方法。在掌握类和方法的编写之前,应教会学生如何使用它们。本书采用的是一个自然推进的教学方式,目的是最终能够让学生在实践中设计出面向对象的问题解决方案。

(10) 重视 GUI 及控件。带有图形用户界面的程序总能激发学生的学习欲望,以这类程序作为讲授面向对象概念的示例会取得很不错的效果。因此,在本书中,示例采用图形用户界面,并介绍这方面的相关知识,包括事件的处理与 GUI 的使用等。通过对这些内容的不断学习,学生们将逐步掌握 GUI 的创建技能。

(11) 重视结合数据库。数据库应用程序如今对所有企业来说是至关重要的,因此本书用两章的篇幅通过具体的实际应用程序案例来讨论使用 ADO.NET 和 SQL 访问数据库的编程技术,并应用三层架构的应用程序设计和开发过程。

针对 GUI 及控件概念的提出,我们采用的是一个循序渐进的过程。在讨论 GUI 及控件之前,学生已对事件的处理与 GUI 的使用有了一个基本的认识。因此在学习 GUI 及控件一章时,我们引入了访问数据库的一些关键性要素。通过创建数据库应用程序来学习 GUI 及

控件。这里采用的是自然推进的教学方式，学生在学习这部分内容时没有感觉困难，而是充满了成就感，激发了学生的学习欲望。

(12) 带给学生真实的编程体验。本书汲取了很多来自实际编程中的经验，这将为培养学生良好的编程习惯打下一个坚实的基础。这些经验会贯穿于所有示例中，并在探讨中得到进一步强调，使学生最终学会如何解决实际问题。在本书中，还将引入并吸收许多来自软件工程方面的基础知识。

我们建议读者遵循书中介绍的方法和步骤实际建立案例程序，然后对案例程序进行修改或扩展，并通过对案例代码进行分析和讨论以掌握案例背后所包含的概念、原理、知识点和方法等，这是学习程序设计最稳妥、最有效、最快捷的途径。

通过本书的学习，读者应达到下列目标：

- 掌握面向对象的程序设计。
- 掌握 Windows 应用程序设计。
- 掌握 ADO.NET 及数据库应用程序设计。
- 学会文件的输入输出。
- 熟练掌握以上内容的综合应用。

尽管作者以细致入微的创作态度来努力完善本书，但错误之处仍在所难免，欢迎读者提出宝贵意见(pmshao@163.com)。本书的源代码和课件可在 <http://www.tup.com.cn> 站点下载。如果我们发现书中的内容和代码有不当之处，我们也会在该站点发布勘误信息。

邵鹏鸣

第2版 前言

C# 是一种简洁、类型安全的面向对象的语言。C# 语法表现力强，而且简单易学。任何熟悉 C、C++ 或 Java 的人都可以立即上手。了解上述任何一种语言的开发人员通常在很短的时间内就可以开始使用 C# 高效地进行工作。C# 语法简化了 C++ 的诸多复杂性，并提供了很多强大的功能，例如属性(充当私有成员变量的访问器)、索引器、委托、Lambda 表达式和匿名方法等，这些都是 Java 所不具备的。C# 的生成过程比 C 和 C++ 简单，比 Java 更为灵活。没有单独的头文件，也不要求按照特定顺序声明方法和类型。Visual C# 2010 提供了高级代码编辑器、方便的用户界面设计器等许多工具，使用户可以更轻松地在 C# 语言 4.0 版和 .NET Framework 4 版的基础上开发应用程序。

近年来，各高职院校和应用型本科院校对计算机课程进行了改革，很多教学方法都被提了出来，也出现了项目课程和项目教材。但一般项目课程存在忽视理论知识或理论知识被割裂、零散化的倾向，存在基础理论知识讲解与项目实践相分离或不能够有机融合的诟病，企业真实项目存在不易学习的缺点。在教学实践中，我们认识到，在组织教学内容，向学生传授知识、方法和技术时，一定要符合学生的认知规律，一定要符合教学规律，这样才能够取得好的效果。在项目课程中，让学生做项目的目的不只是完成项目，不只是为做项目而做项目，而是通过学生完成项目，掌握项目背后包含的原理、方法、技术和知识点，能够让学生做到举一反三。这就是我们的真正目的。学生在完成项目任务时，不仅要让学生明白做什么？怎么做？而且要让学生明白为什么这样做？学生只有理解这样做的“原因”后，才能真正掌握需要掌握的东西，才能够做到举一反三。

针对以上的理解和认识，本书在初期创作的基础上(2005 年 4 月出版的《Visual C#程序设计基础教程》重印 9 次，2008 年 4 月出版的《C#面向对象程序设计》重印 4 次，共计 36000 册)对内容和体系结构做了较大的改进。不仅采纳了全国的老师和学生的反馈建议，而且增加了 C# 4.0 版和 Visual C# 2010 的一些新的特性。

1. 本版增加的新的内容

- (1) 命名实参、可选形参。增加了 C# 4.0 提供的命名实参、可选形参。
- (2) 使用可变数目的参数。
- (3) 增加了 C# 4.0 支持的具有可选形参的实例构造函数。
- (4) 匿名方法和 Lambda 表达式。增加了 C# 2.0 提供的匿名方法。
- (5) Lambda 表达式和 Lambda 语句。在 C# 3.0 及更高版本中，Lambda 包括 Lambda 表达式和 Lambda 语句，它们都是匿名函数，本版增加了该部分的内容。
- (6) 使用匿名方法创建委托。
- (7) 使用 Lambda 创建委托。

(8) 使用匿名函数创建事件处理程序。

(9) 增加了 C# 4.0 提供的逆变和协变。

(10) 使用布局面板进行动态布局。

(11) MenuStrip 控件、ToolStripMenuItem 和 ContextMenuStrip。MenuStrip 控件替换了以前版本的 MainMenu 控件并向其中添加了功能。ToolStripMenuItem 替换了以前版本的 MenuItem 并添加了功能。ContextMenuStrip 替换了以前版本的 ContextMenu。

(12) 场景引入。在每一章的开始，增加了场景引入。在每一章的开始采用如提出问题、回答问题、描述、解释原因等多种方式引入本章要讲解的主要内容。

(13) 分析与设计。在问题描述之后，提出解决方案之前，增加了分析与设计部分。

(14) 独立实践。对每章的每一节增加了独立实践部分。

(15) 项目实践。对每一章增加了一节：项目实践。

(16) 在编写代码之前，增加了算法。

(17) 进一步完善了各部分的代码分析与讨论。

(18) 更换了部分任务。

(19) 对较复杂任务将它分解为若干小任务。

(20) 对章节进行了重新编排，对章进行了改名，对部分节也进行了改名。

(21) 对第 1 章和第 3 章进行了修改，并增加了新的内容。

2. 本书结构

本书由如下所示的一系列任务组成，每个教学模块都有一个任务，通过这一系列任务将学生需要的理论知识穿起来：

- 场景引入
- 任务
- 问题描述
- 分析与设计
- 解决方案
- 分析与讨论
- 独立实践
- 项目实践

3. 本书的主要特点

(1) 本书体系结构、设计编排和内容组织模式有重要创新。

贯穿于教材始末的工作任务具有实用性、教学性，将学生需要的理论知识用工作任务穿起来，保证学习过程有序、连贯、符合学生的认知规律。教材独特的体系结构、设计编排和内容组织模式将复杂的技术体系以简洁易懂的方式呈现给学生。

教材设计了一系列的按任务组织在一起的、复杂程度逐步提高的问题。开始时，教师引导学生解决一些基本问题，随着学生对软件和解决问题方法的熟悉以及所学知识的增加，

问题的复杂程度逐步提高。最后,学生将独立解决问题。

(2) 本教材是基于任务的教材,但避免了一般项目任务教材存在的理论知识被割裂、零散化的倾向,避免了一般项目任务教材基础理论知识讲解与实践相分离或不能够有机融合的诟病,克服了企业真实项目任务不易学习的缺点。

在任务选取过程中考虑了以下几个问题:

- 任务是否真实实用,具有代表性?
- 任务活动怎样调用学生的已有知识?
- 任务内容能否渗透或涉及课程所需的知识点?
- 任务内容能否渗透或涉及课程所需的关键技术?
- 各个任务组合在一起,能否保证学习过程有序,连贯?
- 任务能否让学生感兴趣?是否能激发学生的学习自觉性?

(3) 学生对教材中某个关键技术的掌握是阶梯式的,即先体验认识,然后随着学生对软件 and 解决问题方法的熟悉以及所学知识的增加逐步掌握。教材中的很多关键技术都采用了这种自然推进阶梯式的教学方式。

教材采用了“实践—学习—实践—提高”的教学模式,这种教学模式也是阶梯式的。

4. 教学方法建议

以下是作者在采用本教材授课时采用的主要教学方法,仅供参考。

(1) 在介绍某个主题时,如果可以的话,通过解释为什么需要它,来引入该主题。然后可以对该主题进行最简要的说明。

(2) 提出要完成的任务,描述要解决的问题。在该步骤中,教师可以演示解决该问题的应用程序,让学生了解其工作过程,或让学生自己“尝试”解决该问题的应用程序,了解其工作过程。

(3) 分析问题,设计出算法。

(4) 在解决方案中,按照设计的算法,写出代码。

(5) 代码的分析与讨论。重点分析与讨论所包含的新的知识点,已经学过的知识点一带而过。

(6) 归纳和总结。重点归纳总结出通过完成该任务学生应该掌握什么、应该掌握哪些知识点。

(7) 让学生按照解决方案完成任务。

(8) 提出新任务和新的实际问题,让学生独立实践。新任务和新的实际问题在教材的独立实践部分中。

5. 学习方法建议

应当贯彻“从实践中学习,在学习中实践”原则,“实践”是学习程序设计最稳妥、最有效、最快捷的途径。

书中针对每一个新概念、知识点的提出,都伴随着一个任务,任务是构成本书的主要



部分。强烈建议读者仔细研究这些任务，认真体会完成任务的分析与设计，按照书中提出的解决方案创建完成任务的应用程序。然后，仔细研读代码分析与讨论部分，看懂应用程序中的每行代码。读者在完成任务时，不仅要明白做什么、怎么做，而且要明白为什么这样做。要弄清楚整个的编程思想。然后对应用程序进行修改或扩展。通过这些方式来掌握应用程序背后所包含的概念、原理、知识点和方法等。

接着，应用完成任务时所学的方法、技术，独立完成独立实践中的各项任务。

书中的代码及代码分析与讨论是本书的精华，强烈建议读者认真研究。

另外，建议读者多读别人写的规范的代码，多写自己的代码，如果读不懂别人写的规范的代码，就无从谈起写自己的代码。要多花时间编写程序、测试和调试程序，即使程序有错误也没有关系，如果能够找出错误的所在，找出错误的原因，并改正错误，也能够从错误中学到很多知识。

6. 感谢

陆丽丽负责本书部分文稿的录入、编辑和排版工作，李奋的负责本书第1章至第6章的审阅工作，陆飞鹏负责本书第7章至第13章的审阅工作，肖洋、张升强、黄佳鹏、张连顺、段林飞根据书中的代码，使用 Visual C# 2010，重新对各应用程序的代码进行了调试和测试。对他们的辛勤工作表示由衷的感谢。

受作者水平的限制，加上时间的关系，书中可能会有纰漏和不足，希望读者能够指正和谅解(pmsao@163.com)。

邵鹏鸣

目 录

第 1 章 认识 C#..... 1

1.1 场景引入 1

1.2 什么是.NET..... 1

1.3 .NET 应用程序的编译..... 2

1.3.1 从用户代码到机器代码..... 2

1.3.2 中间语言介绍..... 3

1.4 .NET Framework 平台体系结构..... 3

1.5 C#语言..... 4

1.6 第一个简单的控制台应用程序..... 4

1.6.1 任务 1.1: 打印客户信息..... 4

1.6.2 独立实践: 打印邮寄地址..... 6

1.7 创建简单的 Windows 应用程序..... 7

1.7.1 任务 1.2: 在对话框中显示

客户信息 7

1.7.2 任务 1.3: 显示图像..... 8

1.7.3 任务 1.4: 在文本框中显示

客户信息 10

1.7.4 独立实践: 显示当前时间..... 15

1.8 认识使用程序逻辑解决问题..... 15

1.8.1 程序开发的步骤..... 15

1.8.2 描述算法的工具..... 17

任务 1.5: 计算门票收益和捐款额..... 17

1.8.3 算法实例 19

任务 1.6: 数的排序..... 20

任务 1.7: 投资计算..... 20

1.8.4 独立实践: 画活动图和

指出伪代码 21

1.9 项目实践 21

第 2 章 基本数据的存储与运算..... 24

2.1 场景引入 24

2.2 变量与常数 25

2.2.1 变量的含义和作用 25

2.2.2 标识符..... 25

2.2.3 变量声明..... 26

任务 2.1: 计算路程..... 26

2.2.4 常数..... 29

2.2.5 独立实践: 计算多项式的值 ... 29

2.3 基本数据类型..... 29

2.3.1 整型..... 30

任务 2.2: 产品销售..... 30

2.3.2 字符数据类型..... 33

任务 2.3: 字符检查..... 33

2.3.3 非整型..... 36

任务 2.4: 贷款计算器..... 37

2.3.4 格式化输出..... 40

2.3.5 算术运算..... 41

2.3.6 基本数据类型的相互转换..... 41

任务 2.5: 隐式和强制数值转换 43

2.3.7 算术溢出及显式转换溢出 44

任务 2.6: 算术溢出及

显式转换溢出 44

2.3.8 布尔类型..... 45

任务 2.7: 数值比较..... 46

2.3.9 独立实践..... 46

2.4 项目实践..... 47

第 3 章 使用对象存储数据及完成任务 ... 49

3.1 场景引入..... 49

3.2 类和对象..... 49

3.3 消息和方法..... 50

3.4 使用现有的类..... 51

3.4.1 Form 类 51

任务 3.1: 以编程方式创建窗体..... 51

3.4.2 Form 类的常用属性及方法 53



3.4.3	StringBuilder 类.....	54	4.2.9	独立实践.....	89
任务 3.2:	文字游戏.....	54	4.3	循环语句.....	91
3.4.4	StringBuilder 类的常用		4.3.1	while 语句.....	91
	属性及方法	57	任务 4.7:	计算复利存款(版本 1)	91
3.5	创建自己的类	57	4.3.2	do ... while 语句	94
3.5.1	创建圆类	57	任务 4.8:	计算复利存款(版本 2)	94
任务 3.3:	使用圆类计算圆的面积		4.3.3	for 语句	95
	(版本 1).....	58	任务 4.9:	打印字母表及对应的	
3.5.2	使用 set 方法和 get 方法访问			ASCII 码(版本 1).....	96
	字段	63	4.3.4	嵌套循环.....	98
任务 3.4:	使用圆类计算圆的面积		任务 4.10:	计算销售额.....	98
	(版本 2).....	63	4.3.5	独立实践.....	100
3.5.3	使用可实例化类的		4.4	跳转语句.....	100
	贷款计算器	66	4.4.1	goto 语句.....	100
任务 3.5:	贷款计算器(版本 2).....	66	4.4.2	break 语句.....	103
3.5.4	独立实践	72	4.4.3	continue 语句	104
3.6	项目实践	72	4.4.4	运算符的优先级	105
第 4 章	程序流控制.....	73	4.5	项目实践.....	105
4.1	场景引入	73	第 5 章	使用数组存储一系列数据	107
4.2	选择语句	73	5.1	情景引入.....	107
4.2.1	if 语句	74	5.2	数组	107
任务 4.1:	计时工人工资的计算		任务 5.1:	计算月平均降雨量	108
	(版本 1).....	74	5.2.1	数组初始化.....	112
4.2.2	if ... else 语句	76	任务 5.2:	显示月名称.....	112
任务 4.2:	猜数游戏.....	76	5.2.2	可变长度的数组的声明	113
4.2.3	条件运算符	78	任务 5.3:	创建可变长度的数组	114
任务 4.3:	显示时间(版本 1).....	78	5.2.3	独立实践.....	115
4.2.4	if ... else if ... else 语句	80	5.3	多维数组.....	116
任务 4.4:	工资发放(版本 1).....	81	5.3.1	多维数组的声明和创建	117
4.2.5	if 语句的嵌套	83	5.3.2	多维数组初始化.....	117
任务 4.5:	求数的绝对值.....	83	5.3.3	实例研究.....	117
4.2.6	switch 语句	85	任务 5.4:	测试二维数组	117
任务 4.6:	计算器(版本 1).....	86	任务 5.5:	数据查询.....	119
4.2.7	复合赋值运算符.....	89	任务 5.6:	学生考试成绩统计	121
4.2.8	条件逻辑运算符和		5.3.4	独立实践.....	122
	逻辑运算符	89	5.4	值类型与引用类型	123

5.5 独立实践	126	7.1 场景引入	152
5.6 项目实践	126	7.2 类、对象和封装	153
第 6 章 使用方法完成任务	128	7.3 字段	154
6.1 场景引入	128	任务 7.1: 贷款计算器(版本 3)	154
6.2 传值方式	129	7.3.1 常数和只读字段	157
6.2.1 以传值方式传递值类型		7.3.2 成员访问控制	157
参数	129	7.4 属性	158
任务 6.1: 移动矩形	129	任务 7.2: 贷款计算器(版本 4)	158
6.2.2 以传值方式传递引用类型		7.4.1 类作用域	163
参数	134	7.4.2 索引器	163
任务 6.2: 以传值方式传递数组	134	任务 7.3: 贷款分析	163
6.2.3 独立实践	137	7.4.3 属性与字段、属性和方法的	
6.3 传引用方式	138	比较	168
6.4 以传引用方式传递值类型参数	138	7.4.4 使用 this 关键字	169
6.4.1 引用参数(ref 参数)	138	任务 7.4: 雇员税金计算	169
任务 6.3: 调用方法获得多个值		7.4.5 独立实践	171
(使用 ref 参数)	139	7.5 实例构造函数	171
6.4.2 输出参数(out 参数)	140	7.5.1 默认实例构造函数	171
任务 6.4: 调用方法获得多个值		任务 7.5: 定义 Person 类	172
(使用 out 参数)	140	7.5.2 默认初始化学段	173
6.4.3 ref 参数与 out 参数的比较	142	7.5.3 显式初始化学段	173
任务 6.5: 改变矩形位置	142	7.5.4 实例构造函数声明	174
6.4.4 独立实践	143	任务 7.6: 定义矩形(版本 1)	174
6.5 以传引用方式传递引用类型参数	144	7.5.5 独立实践	177
任务 6.6: 调用方法为实参创建新的		7.6 实例构造函数重载	177
对象	144	7.6.1 使用重载的实例构造函数	177
6.6 独立实践	146	任务 7.7: 定义矩形(版本 2)	177
6.7 命名实参	146	7.6.2 调用同类中的其他构造	
任务 6.7: 投资计算(版本 1)	146	函数	179
6.8 给参数指定默认值	147	任务 7.8: 定义矩形(版本 3)	179
任务 6.8: 投资计算(版本 2)	147	7.6.3 独立实践	180
6.9 使用可变数目的参数	148	7.7 静态成员与实例成员	180
任务 6.9 平均成绩	149	7.7.1 静态字段和实例字段	180
6.10 项目实践	151	任务 7.9: 自动编号	180
第 7 章 使用类与对象设计程序	152	7.7.2 静态构造函数	182
		任务 7.10: 自动编号从随机整数	
		开始	182



7.7.3 静态方法	183	8.4 隐藏从基类继承的成员	210
7.7.4 静态成员和实例成员的特征	183	任务 8.6: 隐藏继承字段	210
7.7.5 独立实践	184	8.5 含直接基类构造函数的构造函数声明	211
7.8 以对象为参数与以返回值为对象	185	8.5.1 任务 8.7: 定义派生类 Student (版本 2)	211
7.8.1 以对象为参数	185	8.5.2 独立实践	212
任务 7.11: 放大矩形	185	8.6 虚拟方法与重写方法	213
任务 7.12: 以传引用方式传递对象参数	187	任务 8.8: 销售员工资系统 (版本 1)	213
7.8.2 以返回值为对象	187	8.6.1 继承中构造函数的执行过程	223
任务 7.13: 放大矩形的副本	187	8.6.2 重载、重写和隐藏的比较	224
7.8.3 独立实践	189	8.6.3 独立实践	225
7.9 方法的重载	189	8.7 垃圾回收和析构函数	225
7.9.1 任务 7.14: 定义矩形 (版本 4)	189	8.7.1 使用析构函数	226
7.9.2 独立实践: 查找数据	191	任务 8.9: 保存状态信息	226
7.10 使用 ArrayList 类	191	8.7.2 声明派生类的析构函数	227
任务 7.15: 地址簿	191	任务 8.10: 保存雇员状态信息	227
7.10.1 ArrayList 类的常用属性	196	8.8 项目实践	228
7.10.2 ArrayList 类的常用方法	196		
7.11 项目实践	197	第 9 章 实现多态性	230
第 8 章 使用继承重用代码	199	9.1 场景引入	230
8.1 场景引入	199	9.2 抽象方法与抽象类	231
8.2 直接基类与派生类	200	9.2.1 抽象方法和抽象属性	231
8.2.1 任务 8.1: 定义基类 Person	200	任务 9.1: 雇员工资系统	231
8.2.2 任务 8.2: 定义派生类 Student (版本 1)	201	9.2.2 抽象类继承	237
8.2.3 独立实践	203	任务 9.2: 多态性及实现	237
8.3 派生类实例构造函数声明	203	9.2.3 独立实践: 绘制图形	242
任务 8.3: 复数加法	203	9.3 接口	243
8.3.1 运算符重载	204	9.3.1 声明和实现接口	243
任务 8.4: 复数减法	205	任务 9.3: 创建和使用接口	244
8.3.2 显式调用直接基类实例构造函数	208	9.3.2 独立实践: 比较对象	247
任务 8.5: 调用基类实例构造函数	209	9.3.3 接口成员的显式实现	248
8.3.3 独立实践	209	任务 9.4: 用接口实现不同的度量衡系统	248
		9.3.4 接口和抽象类	249