



“十二五”职业教育  
国家规划教材  
经全国职业教育教材  
审定委员会审定

# C# 程序设计项目教程

苏传芳 主编

高等教育出版社



“十二五”职业教育国家规划教材  
经全国职业教育教材审定委员会审定

软件技术专业精品教材

C# CHENGXU SHEJI XIANGMU JIAOCHENG

# C# 程序设计项目教程

苏传芳 主编

高等教育出版社·北京

## 内容简介

本书是“十二五”职业教育国家规划教材。

C#是面向对象的程序设计语言,使用C#可以开发Windows平台上的几乎所有的应用程序。本书重在“让项目引领课程”,采用职业情境教学,设计教学模式,精心设计了12个教学项目,以此为载体组织教学与实践内容;以重点培养学生动手能力为目标,适合“自主学习”和“技能训练”,通过精心设计的教学项目阐述了在Visual Studio 2008集成开发环境中使用C#语言进行程序开发的相关知识与技术;项目以简明生动的语言,深入浅出地阐述了C#的基础知识、常用控件、Windows应用程序设计、面向对象编程、文件编程、数据库编程、图形图像编程、网络编程、管理信息系统开发等内容。

本书内容循序渐进、体系合理、易学易用,适合作为高职高专院校计算机类专业教材,也可供各类培训人员、计算机从业人员和计算机爱好者自学或参考使用。

本书配有授课用电子课件,项目任务及实例的源代码等电子资源,教师可发邮件至编辑邮箱1548103297@qq.com索取。

## 图书在版编目(CIP)数据

C#程序设计项目教程/苏传芳主编. --北京:高等教育出版社,2014.8

ISBN 978-7-04-038919-7

I. ①C… II. ①苏… III. ①C语言-程序设计-高等教育-教材 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第277341号

策划编辑 张值胜

责任编辑 张值胜

特约编辑 章兴敏

封面设计 张雨微

版式设计 王艳红

责任校对 陈旭颖

责任印制 赵义民

出版发行 高等教育出版社

社 址 北京市西城区德外大街4号

邮政编码 100120

印 刷 大厂益利印刷有限公司

开 本 787mm×1092mm 1/16

印 张 21.5

字 数 470千字

购书热线 010-58581118

咨询电话 400-810-0598

网 址 <http://www.hep.edu.cn>

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

网上订购 <http://www.landaco.com>

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

版 次 2014年8月第1版

印 次 2014年8月第1次印刷

定 价 36.80元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 38919-00

# 出版说明

教材是教学过程的重要载体，加强教材建设是深化职业教育教学改革的有效途径，推进人才培养模式改革的重要条件，也是推动中高职协调发展的基础性工程，对促进现代职业教育体系建设，切实提高职业教育人才培养质量具有十分重要的作用。

为了认真贯彻《教育部关于“十二五”职业教育教材建设的若干意见》（教职成〔2012〕9号），2012年12月，教育部职业教育与成人教育司启动了“十二五”职业教育国家规划教材（高等职业教育部分）的选题立项工作。作为全国最大的职业教育教材出版基地，我社按照“统筹规划，优化结构，锤炼精品，鼓励创新”的原则，完成了立项选题的论证遴选与申报工作。在教育部职业教育与成人教育司随后组织的选题评审中，由我社申报的1338种选题被确定为“十二五”职业教育国家规划教材立项选题。现在，这批选题相继完成了编写工作，并由全国职业教育教材审定委员会审定通过后，陆续出版。

这批规划教材中，部分为修订版，其前身多为普通高等教育“十一五”国家级规划教材（高职高专）或普通高等教育“十五”国家级规划教材（高职高专），在高等职业教育教学改革进程中不断吐故纳新，在长期的教学实践中接受检验并修改完善，是“锤炼精品”的基础与传承创新的硕果；部分为新编教材，反映了近年来高职院校教学内容与课程体系改革的成果，并对接新的职业标准和新的产业需求，反映新知识、新技术、新工艺和新方法，具有鲜明的时代特色和职教特色。无论是修订版，还是新编版，我社都将发挥自身在数字化教学资源建设方面的优势，为规划教材开发配备数字化教学资源，实现教材的一体化服务。

这批规划教材立项之时，也是国家职业教育专业教学资源库建设项目及国家精品资源共享课建设项目深入开展之际，而专业、课程、教材之间的紧密联系，无疑为融通教改项目、整合优质资源、打造精品力作奠定了基础。我社作为国家专业教学资源库平台建设和资源运营机构及国家精品开放课程项目实施单位，将建设成果以系列教材的形式成功申报立项，并在审定通过后陆续推出。这两个系列的规划教材，具有作者队伍强大、教改基础深厚、示范效应显著、配套资源丰富、纸质教材与在线资源一体化设计的鲜明特点，将是职业教育信息化条件下，扩展教学手段和范围，推动教学方式方法变革的重要媒介与典型代表。

教学改革无止境，精品教材永追求。我社将在今后一到两年内，集中优势力量，全力以赴，出版好、推广好这批规划教材，力促优质教材进校园、精品资源进课堂，从而更好地服务于高等职业教育教学改革，更好地服务于现代职教体系建设，更好地服务于青年成才。

高等教育出版社

2014年7月

# 前言

.NET 是一种用在 Windows 平台上编程的新架构, 可以支持多种编程语言。C# 是专门用于 .NET 的简单、类型安全、面向对象且功能强大的编程语言, 使用 C# 可以开发 Windows 平台上的几乎所有的应用程序。

高职院校的学生一般认为程序设计不好学, 而对面向对象的程序设计更是头疼。本书的编写目的是开发一种适合高职院校学生学习的面向对象的程序设计教材, 使学生自然地接受类、对象等概念, 并能应用于程序设计中, 变抽象为具体, 变思维为实践。编者总结多年的高职院校程序设计语言的教学实践与实际项目开发经验, 采用了“先动手后动脑, 实践带动理论, 项目引领课程”的教学思想, 本书起到了引人入胜、事半功倍的教学效果。

本书的亮点是“让项目引领课程”, 采用职业情境教学, 设计教学模式, 精心设计了 12 个教学项目, 灵活运用任务驱动方法, 以此为载体组织教学与实践内容。例如, 在学习网络编程与多线程教学内容时设计了“网络聊天室”项目。该项目使用了套接字编程、多线程、Socket 类、TcpClient 类等有一定难度的知识点, 教学实践表明, 同学们对“网络聊天室”项目兴趣浓厚, 学习积极踊跃, 通过设置一定的问题情境, 由浅入深地进入项目实践阶段, 可以在较短时间内轻松实现目标, 从而掌握相应知识点, 学会相应技能。其他项目还有 C# 程序的实现、简易计算器、用户注册程序、文本编辑器、信用卡管理、防盗门自动报警程序、文件浏览器、密码修改程序、学生基本信息管理、简单画图程序、企业库存管理系统等, 涵盖了 C# 基础知识、常用控件、Windows 应用程序设计、面向对象编程、文件编程、数据库编程、图形图像编程、网络编程、管理信息系统开发等知识点。

本书有如下特点:

(1) 改变传统的章节编排模式, 采用“项目+任务”的形式, 以项目实施为逻辑顺序编排教学内容。

(2) 每个项目都包括技能目标、知识目标、项目要求、项目分析、举一反三、自己动手、习题等 7 个部分, 其中“项目分析”部分又将项目划分成若干个任务, 每个任务又包括问题情境、具体实现、相关知识和知识扩展 4 个模块。每一个项目的出现, 首先展示给读者一个明确的学习目标, 然后以任务驱动的方式告诉读者如何实现学习目标, 以及实现目标所需要掌握的知识与技能, 最后通过“举一反三”和“自己动手”提高读者的实际动手能力。

(3) 为提高读者的学习效率, 在介绍项目任务相关知识点时使用了丰富的趣味性实例,

并给出了设计思路或实现步骤。

(4) 内容通俗易懂、深入浅出,符合高职学生的学习认知习惯。

本书建议总学时数为 90 学时,项目 12 为综合项目,可以作为课程设计使用。另外,本书还配有授课用电子课件,项目任务及实例的源代码,可以通过发送邮件至编辑邮箱 1548103297@99.com 索取配套资源。

本书由安徽电子信息职业技术学院苏传芳任主编,夏克付任副主编,安徽电子信息职业技术学院教师参与编写,其中项目 1 和项目 4 由苏传芳编写;项目 2 和项目 8 由毕好昌编写;项目 3 和项目 9 由李媛媛编写;项目 5 和项目 6 由苏警编写;项目 7 和项目 11 由夏克付编写;项目 10 和项目 12 由章晓勤编写。苏传芳负责全书的总体设计和统稿。

由于行业知识更新快,书中疏漏和不足之处在所难免,敬请读者批评指正。

编 者

2014 年 2 月

# 目 录

|                                 |     |
|---------------------------------|-----|
| 项目 1 C#程序的实现                    | 1   |
| 任务 1.1 初识.NET Framework<br>与 C# | 1   |
| 任务 1.2 创建控制台应用程序                | 4   |
| 1.2.1 问题情境                      | 4   |
| 1.2.2 具体实现                      | 4   |
| 1.2.3 相关知识                      | 5   |
| 1.2.4 知识扩展                      | 7   |
| 任务 1.3 创建 Windows 应用程序          | 7   |
| 1.3.1 问题情境                      | 7   |
| 1.3.2 具体实现                      | 7   |
| 1.3.3 相关知识                      | 8   |
| 1.3.4 知识扩展                      | 10  |
| 举一反三                            | 10  |
| 自己动手                            | 11  |
| 习题 1                            | 12  |
| 项目 2 简易计算器                      | 13  |
| 任务 2.1 数据的输入与输出                 | 13  |
| 2.1.1 问题情境                      | 13  |
| 2.1.2 具体实现                      | 13  |
| 2.1.3 相关知识                      | 14  |
| 2.1.4 知识扩展                      | 25  |
| 任务 2.2 数据的计算                    | 28  |
| 2.2.1 问题情境                      | 28  |
| 2.2.2 具体实现                      | 28  |
| 2.2.3 相关知识                      | 30  |
| 2.2.4 知识扩展                      | 45  |
| 举一反三                            | 46  |
| 自己动手                            | 47  |
| 习题 2                            | 48  |
| 项目 3 用户注册程序                     | 51  |
| 任务 3.1 创建用户账号                   | 51  |
| 3.1.1 问题情境                      | 51  |
| 3.1.2 具体实现                      | 51  |
| 3.1.3 相关知识                      | 54  |
| 3.1.4 知识扩展                      | 67  |
| 任务 3.2 设置用户信息                   | 68  |
| 3.2.1 问题情境                      | 68  |
| 3.2.2 具体实现                      | 69  |
| 3.2.3 相关知识                      | 71  |
| 3.2.4 知识扩展                      | 78  |
| 举一反三                            | 80  |
| 自己动手                            | 83  |
| 习题 3                            | 84  |
| 项目 4 文本编辑器                      | 86  |
| 任务 4.1 单文档界面的文本编辑               | 86  |
| 4.1.1 问题情境                      | 86  |
| 4.1.2 具体实现                      | 86  |
| 4.1.3 相关知识                      | 95  |
| 4.1.4 知识扩展                      | 104 |
| 任务 4.2 多文档界面的文本编辑               | 106 |

## II 目录

|                   |     |                   |     |
|-------------------|-----|-------------------|-----|
| 4.2.1 问题情境        | 106 | 举一反三              | 171 |
| 4.2.2 具体实现        | 106 | 自己动手              | 172 |
| 4.2.3 相关知识        | 110 | 习题 6              | 172 |
| 4.2.4 知识扩展        | 113 |                   |     |
| 举一反三              | 115 | 项目 7 文件浏览器        | 175 |
| 自己动手              | 119 | 任务 7.1 文件与目录信息的获取 | 175 |
| 习题 4              | 120 | 7.1.1 问题情境        | 175 |
|                   |     | 7.1.2 具体实现        | 175 |
|                   |     | 7.1.3 相关知识        | 179 |
|                   |     | 7.1.4 知识扩展        | 186 |
|                   |     | 任务 7.2 文件内容的读取    | 189 |
|                   |     | 7.2.1 问题情境        | 189 |
|                   |     | 7.2.2 具体实现        | 189 |
|                   |     | 7.2.3 相关知识        | 193 |
|                   |     | 7.2.4 知识扩展        | 200 |
|                   |     | 举一反三              | 201 |
|                   |     | 自己动手              | 204 |
|                   |     | 习题 7              | 204 |
| 项目 5 信用卡管理        | 122 |                   |     |
| 任务 5.1 实现基本账户管理   | 122 | 项目 8 密码修改程序       | 206 |
| 5.1.1 问题情境        | 122 | 任务 8.1 连接数据库      | 206 |
| 5.1.2 具体实现        | 122 | 8.1.1 问题情境        | 206 |
| 5.1.3 相关知识        | 125 | 8.1.2 具体实现        | 206 |
| 5.1.4 知识扩展        | 139 | 8.1.3 相关知识        | 207 |
| 任务 5.2 实现多类账户管理   | 142 | 8.1.4 知识扩展        | 210 |
| 5.2.1 问题情境        | 142 | 任务 8.2 访问数据       | 211 |
| 5.2.2 具体实现        | 142 | 8.2.1 问题情境        | 211 |
| 5.2.3 相关知识        | 145 | 8.2.2 具体实现        | 211 |
| 5.2.4 知识扩展        | 148 | 8.2.3 相关知识        | 214 |
| 举一反三              | 149 | 8.2.4 知识扩展        | 217 |
| 自己动手              | 151 | 举一反三              | 219 |
| 习题 5              | 152 | 自己动手              | 225 |
|                   |     | 习题 8              | 225 |
| 项目 6 防盗门自动报警程序    | 154 |                   |     |
| 任务 6.1 防盗门和报警器的设计 | 154 |                   |     |
| 6.1.1 问题情境        | 154 |                   |     |
| 6.1.2 具体实现        | 154 |                   |     |
| 6.1.3 相关知识        | 155 |                   |     |
| 6.1.4 知识扩展        | 161 |                   |     |
| 任务 6.2 自动报警的实现    | 163 |                   |     |
| 6.2.1 问题情境        | 163 |                   |     |
| 6.2.2 具体实现        | 163 |                   |     |
| 6.2.3 相关知识        | 166 |                   |     |
| 6.2.4 知识扩展        | 170 |                   |     |

|                        |     |                        |     |
|------------------------|-----|------------------------|-----|
| 项目 9 学生基本信息管理·····     | 227 | 11.1.1 问题情境·····       | 282 |
| 任务 9.1 数据管理与绑定·····    | 227 | 11.1.2 具体实现·····       | 282 |
| 9.1.1 问题情境·····        | 227 | 11.1.3 相关知识·····       | 285 |
| 9.1.2 具体实现·····        | 228 | 11.1.4 知识扩展·····       | 291 |
| 9.1.3 相关知识·····        | 229 | 任务 11.2 多用户通信·····     | 292 |
| 9.1.4 知识扩展·····        | 232 | 11.2.1 问题情境·····       | 292 |
| 任务 9.2 管理数据库中的数据·····  | 233 | 11.2.2 具体实现·····       | 292 |
| 9.2.1 问题情境·····        | 233 | 11.2.3 相关知识·····       | 298 |
| 9.2.2 具体实现·····        | 233 | 11.2.4 知识扩展·····       | 302 |
| 9.2.3 相关知识·····        | 236 | 举一反三·····              | 303 |
| 9.2.4 知识扩展·····        | 243 | 自己动手·····              | 306 |
| 举一反三·····              | 245 | 习题 11·····             | 306 |
| 自己动手·····              | 249 |                        |     |
| 习题 9·····              | 250 | 项目 12 企业库存管理系统·····    | 307 |
|                        |     | 任务 12.1 系统分析·····      | 307 |
| 项目 10 简单画图程序·····      | 252 | 12.1.1 企业库存管理系统的       |     |
| 任务 10.1 绘制基本图形·····    | 252 | 数据处理需求·····            | 307 |
| 10.1.1 问题情境·····       | 252 | 12.1.2 企业库存管理系统的       |     |
| 10.1.2 具体实现·····       | 252 | 数据需求·····              | 308 |
| 10.1.3 相关知识·····       | 254 | 任务 12.2 系统设计·····      | 308 |
| 10.1.4 知识扩展·····       | 261 | 12.2.1 系统功能模块设计·····   | 309 |
| 任务 10.2 动态绘图·····      | 262 | 12.2.2 数据库设计·····      | 310 |
| 10.2.1 问题情境·····       | 262 | 任务 12.3 公共类与实体类设计····· | 319 |
| 10.2.2 具体实现·····       | 262 | 12.3.1 公共类设计·····      | 319 |
| 10.2.3 相关知识·····       | 269 | 12.3.2 实体类设计·····      | 321 |
| 10.2.4 知识扩展·····       | 276 | 任务 12.4 主要功能模块设计·····  | 324 |
| 举一反三·····              | 278 | 12.4.1 用户登录模块设计·····   | 324 |
| 自己动手·····              | 280 | 12.4.2 产品入库模块设计·····   | 326 |
| 习题 10·····             | 281 | 12.4.3 入库信息查询模块设计····· | 330 |
|                        |     |                        |     |
| 项目 11 网络聊天室·····       | 282 | 参考文献·····              | 333 |
| 任务 11.1 服务器与客户机通信····· | 282 |                        |     |

# 项目 1

## C#程序的实现

### 技能目标

能创建简单的 C#程序。

### 知识目标

- ① 了解 C#与 .NET Framework。
- ② 掌握开发工具 Visual Studio 2008 的安装。
- ③ 熟悉 C#的集成开发环境。
- ④ 掌握 C#程序的创建方法。

### 项目要求

- ① 安装 Visual Studio 2008 并创建简单的 C#程序。
- ② 程序运行，并输出“欢迎使用 C#”。

### 项目分析

要创建 C#程序，必须了解 C#与 .NET Framework 和学会控制台应用程序与 Windows 应用程序的创建方法。因此，本项目将分成 3 个任务来介绍。任务 1.1 是对 C#与 .NET 平台进行详细介绍，任务 1.2 是创建控制台应用程序，任务 1.3 是创建 Windows 应用程序。

## 任务 1.1 初识 .NET Framework 与 C#

### 1. .NET Framework

.NET Framework 是微软公司为开发应用程序而创建的一个全新的平台，它提供了一个跨语言的统一编程环境。它支持生成和运行下一代应用程序和 XML Web Services 的内部 Windows 组件。其包括 3 个主要部分：公共语言运行库（CLR）、.NET Framework 类库和 ASP.NET。

#### (1) .NET Framework 的主要特点

① 跨平台性。.NET Framework 除了可以在 Windows 操作系统下使用之外，它的其他版本还可以工作在别的操作系统下，如各种 Linux 版本、Mac OS 等，甚至可以在 PDA 和一些智能手机上使用。

② 支持多种语言。.NET Framework 与编程语言无关，任何语言都可以支持，目前支持 C++、Visual Basic.NET、JScript、C#等，将来还会支持更多的语言。这些语言之间还可以相互

调用, Visual Basic 的开发人员可以调用 C#编写的代码, 反之亦然。

③ 支持多种类型的应用程序。利用 .NET Framework 除了可以编写传统的 Windows 应用程序外, 还支持 Web 应用程序、Web 服务和其他各种类型的应用程序。

- Windows 应用程序, 具有用户熟悉的 Windows 应用程序外观和操作方式, 使用 .NET Framework 的 Windows Forms 模块就可以生成这种应用程序。Windows Form 模块是一个控件库, 其中的控件可以用于建立 Windows 用户界面 (UI), 如按钮、工具栏、菜单等。本书重点介绍该类应用程序的开发。

- Web 应用程序, 即 Web 页, 可以通过任何 Web 浏览器查看。 .NET Framework 包括一个动态生成 Web 内容的强大系统, 允许个性化、实现安全性等。该系统称为 Active Server Pages.NET (ASP.NET), 用户可以使用 C#通过 Web Forms 创建 ASP.NET 应用程序。

- Web 服务, 创建各种分布式应用程序的新方式, 使用 Web 服务可以通过因特网虚拟交换数据。无论使用什么语言创建 Web 服务, 也无论 Web 服务驻留在什么系统上, 都使用一样简单的语法。

## (2) .NET Framework 的主要组成部分

① 公共语言运行库 (CLR), 是 .NET Framework 的基础, 它运行代码并且在代码开发过程中提供一系列的服务, 使开发过程更加轻松。其是在执行时管理代码的代理, 并提供内存管理、线程管理、远程处理等核心服务。

② .NET Framework 类库, 为开发人员提供了一个统一、面向对象、层次化、可扩展的类库, 包括类、接口和值类型。 .NET Framework 统一了微软公司的各种不同类框架。这样, 开发人员无需学习多种框架就能顺利编程。利用公共类库, .NET Framework 可实现跨语言继承性、错误处理功能和调试功能。此外, 第三方组件可与 .NET Framework 中的类无缝集成。

③ ASP.NET, 提供了 Web 应用程序模型。开发人员可以直接使用 ASP.NET 控件集, 该控件集封装了公共的、用于超文本标记语言 (HTML) 用户界面的各种小控件, 如文本框、下拉菜单等。利用它, Web 应用程序的构建变得非常容易。

## 2. C#

C#是创建 .NET 程序的语言之一, 从 C 与 C++语言演化而来, 是微软公司专门为使用 .NET 平台而创建的语言。它吸取了其他语言的优点, 并克服了它们的不足之处。微软公司对 C#的定义是: “一种类型安全、现代、简单, 由 C 和 C++衍生出来的面向对象的编程语言, 它是牢牢根植于 C 和 C++语言之上的, 并可立即被 C 和 C++的使用者所熟悉。C#的目的就是综合 Visual Basic 的高生产率和 C++的行动力。” 利用 C#可以创建多种类型的应用程序, 如 Windows 应用程序、Web 应用程序、Web 服务等。

此外, C#还具有一个统一的类型系统, 所有 C#类型都继承于一个唯一的基类型 Object。因此, 所有类型都共享一组通用操作, 并且任何类型的值都能够以一致的方式进行存储、传递和操作。C#同时支持用户定义的引用类型和值类型, 既允许对象的动态分析, 也允许轻量结构的内联存储。

## 3. Visual Studio 2008 介绍

Visual Studio 2008 是一套完整的开发工具集, 用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic、Visual C++、Visual C#全都使用这

个集成开发环境 (IDE), 利用该 IDE 可以共享工具且有助于创建混合语言解决方案。另外, 这些语言利用了 .NET Framework 的功能, 通过此框架可使用简化 ASP Web 应用程序和 XML Web Services 开发的关键技术。

使用 C#开发.NET Framework 应用程序之前, 应先部署.NET Framework 环境, 包括安装.NET Framework、Visual Studio 2008 等。在安装 Visual Studio 2008 时, .NET Framework 会自动安装, 读者也可以从微软公司的站点免费下载.NET Framework。

#### 4. Visual Studio 2008 的安装

① 将 Visual Studio 2008 安装光盘放入光驱, 并运行光盘内 autorun.exe 文件, 打开如图 1-1 所示窗口。

② 选择“安装 Visual Studio 2008”选项, 系统自动加载安装组件, 完成后打开如图 1-2 所示窗口。



图 1-1 Visual Studio 2008 安装主界面



图 1-2 加载安装组件界面

③ 单击“下一步”按钮, 打开如图 1-3 所示窗口, 选择“我已阅读并接受许可条款”单选按钮, 输入密钥后单击“下一步”按钮。

④ 在打开的“选项页”页面中选择要安装的功能及安装路径, 如图 1-4 所示。



图 1-3 输入密钥界面

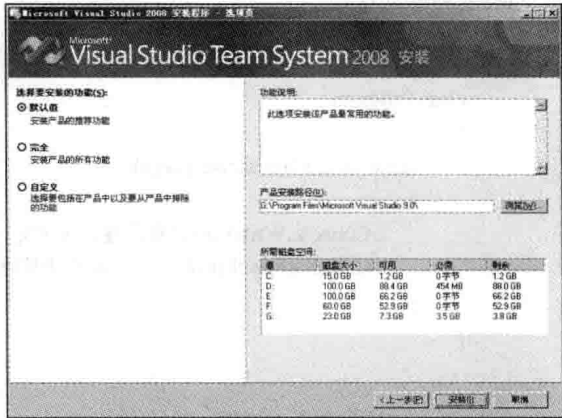


图 1-4 选择安装功能界面

⑤ 单击“安装”按钮，打开安装页面，如图 1-5 所示。

⑥ 安装完成后，会打开如图 1-6 所示窗口，单击“完成”按钮并重新启动计算机完成安装。

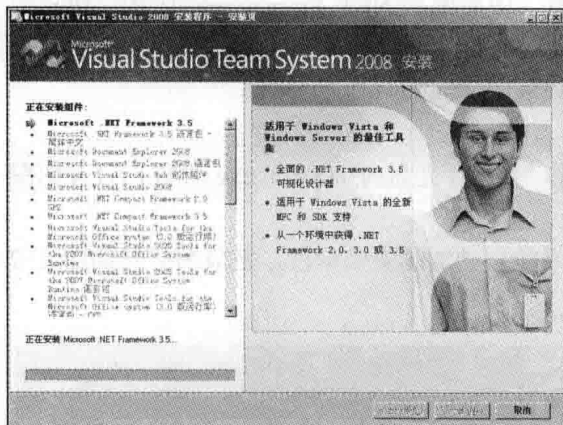


图 1-5 正在安装界面

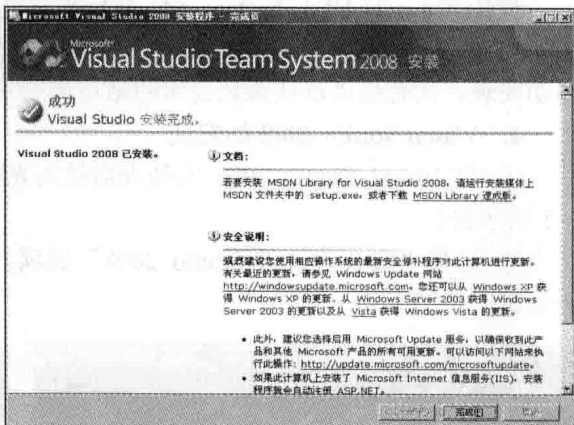


图 1-6 安装完毕界面

## 任务 1.2 创建控制台应用程序

### 1.2.1 问题情境

在控制台输出“欢迎使用 C#”。

### 1.2.2 具体实现

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace FirstConsole
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("欢迎使用 C#");
            Console.ReadLine(); //若不加此语句，程序运行后窗体一闪而过，看不到运行结果
        }
    }
}
```

程序运行界面如图 1-7 所示。



图 1-7 控制台应用程序运行界面

### 1.2.3 相关知识

#### 1. 控制台 C# 的运行环境

(1) 单击“开始”按钮，选择“程序→Microsoft Visual Studio 2008→Microsoft Visual Studio 2008”命令，启动 Microsoft Visual Studio 2008，其起始页如图 1-8 所示。

(2) 选择菜单“文件→新建→项目”命令，打开“新建项目”对话框，如图 1-9 所示。

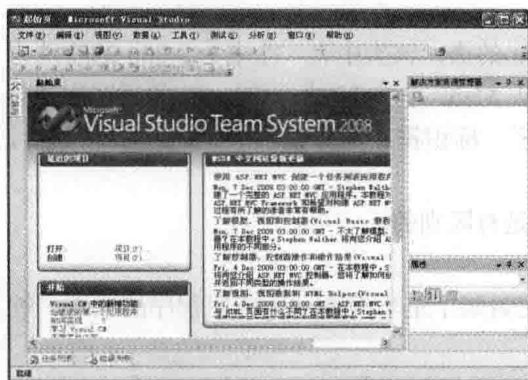


图 1-8 Visual Studio 2008 起始页



图 1-9 “新建项目”对话框

(3) 在“项目类型”列表框中选择“Visual C#→Windows”选项，在“模板”列表框中选择“控制台应用程序”选项。在“名称”文本框中输入项目的名称，如 FirstConsole，单击“确定”按钮，打开如图 1-10 所示窗口。

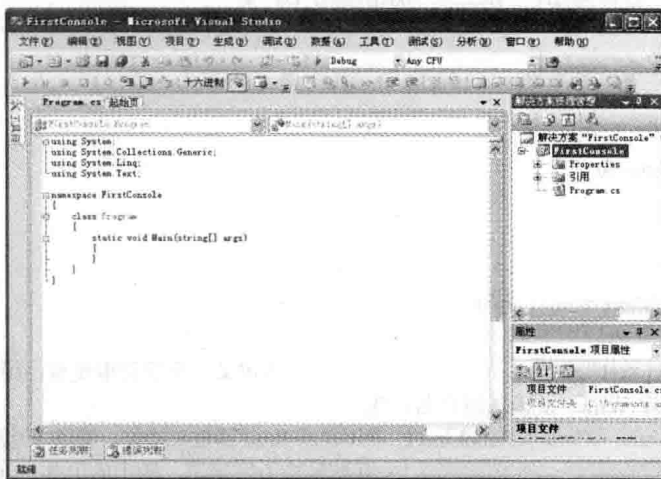


图 1-10 控制台应用程序集成开发环境

(4) 窗口右侧为“解决方案资源管理器”窗口，其中包括的文件，由系统自动创建。

① 解决方案“FirstConsole”：它是解决方案文件，每个应用程序都有一个类似的文件，其扩展名为 sln。每个解决方案文件都包含了一个或多个项目文件的引用。

② FirstConsole：C# 的项目文件，其扩展名为 csproj。每个项目文件都引用一个或多个包

含项目源码的文件。

③ Properties: 它是一个文件夹, 包含一个名为 AssemblyInfo.cs 的文件, 可以在该文件中添加“作者”、“日期”等属性。

④ 引用: 该文件夹包含对程序可用的已编译代码的引用。

⑤ Program.cs: 它是一个 C#源代码文件, 用户编写的代码都包含在这个文件中。

(5) 在 Program.cs 文件中编写代码, 选择菜单“调试→启动调试”命令, 运行程序。

## 2. C#程序的结构

在 Program.cs 文件中, 包含了一个名为 Program 的类, 该类中有一个名为 Main 的特殊方法, 它指定了 C#程序的入口, 即任何 C#程序都是从 Main 方法开始执行的。C#的语法规则与 C 语言类似, 语句以“;”为结束符, 语句由关键字、标识符、运算符和表达式构成。其中, “{”、“}”表示函数体的起点和终点。

注意: C#区分英文字母的大小, 如 abc 与 Abc 是有区别的。

## 3. Console 类

Console 类是 System 命名空间中的一个类, 它封装了基于控制台的应用程序的输入、输出和错误流操作。其常用的方法有:

① Read 方法, 从标准输入流读取下一个字符。

② ReadLine 方法, 从标准输入流读取下一行字符。

③ Write 方法, 将指定的字符串值写入标准的输出流。

④ WriteLine 方法, 将指定格式的字符串值写入标准的输出流, 并回车换行。

【例 1-1】创建一个控制台应用程序。程序运行后, 提示“请输入用户名:”, 用户输入“张三”并按回车键, 程序输出“张三, 欢迎使用 C#”。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace SecondConsole
{
    class Program
    {
        static void Main(string[] args)
        {
            string UserName; //定义一个字符串变量, 用以存储用户的输入
            Console.WriteLine("请输入用户名: ");
            UserName = Console.ReadLine(); /*不使用 Read 函数来接收, 该函数返回类型是整型,
                                           即使转换为字符串, 输出的也是 ASCII 码*/
            Console.WriteLine("{0}, 欢迎使用 C#", UserName); /*{0}表示在该位置输出 UserName
                                                             变量的值*/
            Console.ReadLine();
        }
    }
}
```

### 1.2.4 知识扩展

#### 1. 命名空间

随着程序规模的扩大，代码中会有更多的名称、数据、方法以及更多的已命名类，这就极有可能造成两个或多个名称发生冲突，造成项目生成失败。

为解决该问题，微软公司引入了命名空间（namespace）的概念，它能为各种标识符创建一个已命名的容器，即两个同名的类在不同的命名空间中，是不会发生冲突的。调用命名空间中的类时，采用“命名空间名.类名”的方式。如，Console 是 System 命名空间下的类，用 System.Console 来调用该类。

细心的读者会发现，在任务 1.2 的代码中，并没有使用 System.Console 来调用 Console 类，而是直接使用 Console。这是因为在代码开头部分有 using System 语句，using 语句限定了即将使用的命名空间。在后续的代码中，不需要再用命名空间来显式地限定每一个名称。若在任务 1.2 中将 using System 语句删除，再运行程序，系统将会报错。

#### 2. 智能感知技术

在代码编辑区，将光标定位到 Main 方法体内，并输入 Console。输入单首字母 C 时，会自动弹出一个智能感知列表，列表中包含所有有效的 C#关键字以及在该情况下有效的数据类型。此时可继续输入，或滚动列表做出选择。如在输入 Conso 之后，智能感知列表会自动定位到 Console 条目，可以按 Tab 键或回车键直接选择。在 Console 之后，立即输入“.”符号，会出现另一个智能感知列表，其中列出了 Console 类的方法、属性和字段；选中某个方法、属性或字段后，系统会对其功能进行简单解释。

## 任务 1.3 创建 Windows 应用程序

### 1.3.1 问题情境

创建一个 Windows 应用程序。要求程序运行后，单击“确定”按钮，程序输出“欢迎使用 C#”；单击“取消”按钮，程序结束运行。

### 1.3.2 具体实现

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace FirstWinForm
{
    public partial class Form1 : Form
```

```
{  
    public Form1()  
    {  
        InitializeComponent();  
    }  
    private void button1_Click(object sender, EventArgs e)  
    {  
        textBox1.Text = "欢迎使用 C#";  
    }  
    private void button2_Click(object sender, EventArgs e)  
    {  
        Application.Exit(); //退出程序  
    }  
}
```



程序运行界面如图 1-11 所示。

图 1-11 Windows 应用程序运行界面

### 1.3.3 相关知识

#### 1. Windows 应用程序的开发环境

(1) 运行 Visual Studio 2008, 选择菜单“文件→新建→项目”命令, 打开“新建项目”对话框, 如图 1-9 所示。

(2) 在“项目类型”列表框中选择“Visual C#→Windows”选项, 在“模板”列表框中选择“Windows 窗体应用程序”选项, 并在“名称”文本框中输入 FirstWinForm, 单击“确定”按钮, 打开如图 1-12 所示窗口。

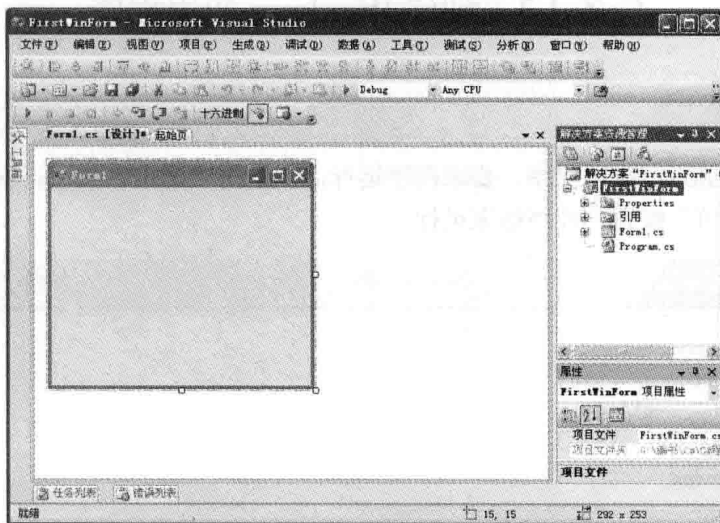


图 1-12 Windows 窗体应用程序集成开发环境

(3) 在设计窗口中显示一个空白的名为 Form1 的 Windows 窗体, 在“解决方案资源管理器”窗口中, 多了一个 Form1.cs 文件, 该文件即 Form1 的窗体文件。