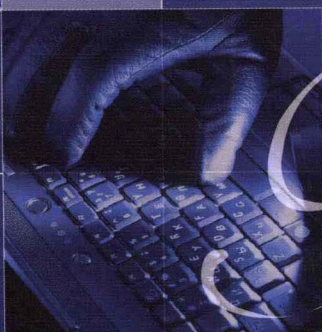
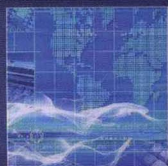


C#面向对象程序设计 及实践教程

唐 燕 主 编



Programming



- 零起点学习C#编程，做到从入门到精通
- 丰富的实用性案例，真实传授编程技巧
- 多样的习题和素材，着实提高实战能力



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国应用型本科计算机案例型规划教材

C#面向对象程序设计及实践教程

主 编 唐 燕

副主编 韩爱庆 王 丽



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书介绍了 C#语言概述, C#语法基础, 类, 继承和多态, 接口、委托和事件, 集合和泛型, 异常处理, Windows 应用程序及常用控件, 图形图像编程, 文件和流, C#数据库编程以及 C# Web 应用程序基础等内容。全书提供了丰富的实例代码, 并在每章精心安排了与本章内容相关、与实际开发紧密结合的案例。各章通过案例介绍, 结合大量实例, 层层深入, 介绍相关知识点。本书为读者免费提供教学及学习资料, 包括电子课件和书中涉及的项目代码。

本书可作为高等院校计算机、信息管理等相关专业的教学用书, 也可作为工程技术人员的参考用书。

图书在版编目(CIP)数据

C#面向对象程序设计及实践教程/唐燕主编. —北京: 北京大学出版社, 2012.10

(21 世纪全国应用型本科计算机案例型规划教材)

ISBN 978-7-301-21271-4

I. ①C… II. ①唐… 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(2012)第 222072 号

书 名: C#面向对象程序设计及实践教程

著作责任者: 唐 燕 主编

策 划 编 辑: 郑 双

责 任 编 辑: 郑 双

标 准 书 号: ISBN 978-7-301-21271-4/TP · 1247

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.cn>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电 子 邮 箱: pup_6@163.com

印 刷 者: 北京鑫海金澳胶印有限公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 23.75 印张 546 千字

2012 年 10 月第 1 版 2012 年 10 月第 1 次印刷

定 价: 45.00 元

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有 侵权必究

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

21 世纪全国应用型本科计算机案例型规划教材

专家编审委员会

(按姓名拼音顺序)

主 任 刘瑞挺

副主任 陈 钟 蒋宗礼

委 员 陈代武 房爱莲 胡巧多 黄贤英

江 红 李 建 娄国焕 马秀峰

祁亨年 王联国 汪新民 谢安俊

解 凯 徐 苏 徐亚平 宣兆成

姚喜妍 于永彦 张荣梅

信息技术的案例型教材建设

(代丛书序)

刘瑞挺

北京大学出版社第六事业部在 2005 年组织编写了《21 世纪全国应用型本科计算机系列实用规划教材》，至今已出版了 50 多种。这些教材出版后，在全国高校引起热烈反响，可谓初战告捷。这使北京大学出版社的计算机教材市场规模迅速扩大，编辑队伍茁壮成长，经济效益明显增强，与各类高校师生的关系更加密切。

2008 年 1 月北京大学出版社第六事业部在北京召开了“21 世纪全国应用型本科计算机案例型教材建设和教学研讨会”。这次会议为编写案例型教材做了深入的探讨和具体的部署，制定了详细的编写目的、丛书特色、内容要求和风格规范。在内容上强调面向应用、能力驱动、精选案例、严把质量；在风格上力求文字精练、脉络清晰、图表明快、版式新颖。这次会议吹响了提高教材质量第二战役的进军号。

案例型教材真能提高教学的质量吗？

是的。著名法国哲学家、数学家勒内·笛卡儿(Rene Descartes, 1596—1650)说得好：“由一个例子的考察，我们可以抽出一条规律。(From the consideration of an example we can form a rule.)”事实上，他发明的直角坐标系，正是通过生活实例而得到的灵感。据说是在 1619 年夏天，笛卡儿因病住进医院。中午他躺在病床上，苦苦思索一个数学问题时，忽然看到天花板上有一只苍蝇飞来飞去。当时天花板是用木条做成正方形的格子。笛卡儿发现，要说出这只苍蝇在天花板上的位置，只需说出苍蝇在天花板上的第几行和第几列。当苍蝇落在第四行、第五列的那个正方形时，可以用(4, 5)来表示这个位置……由此他联想到可用类似的办法来描述一个点在平面上的位置。他高兴地跳下床，喊着“我找到了，找到了”，然而不小心把国际象棋撒了一地。当他的目光落到棋盘上时，又兴奋地一拍大腿：“对，对，就是这个图”。笛卡儿锲而不舍的毅力，苦思冥想的钻研，使他开创了解析几何的新纪元。千百年来，代数与几何，井水不犯河水。17 世纪后，数学突飞猛进的发展，在很大程度上归功于笛卡儿坐标系和解析几何学的创立。

这个故事，听起来与阿基米德在浴缸洗澡而发现浮力原理，牛顿在苹果树下遇到苹果落到头上而发现万有引力定律，确有异曲同工之妙。这就证明，一个好的例子往往能激发灵感，由特殊到一般，联想出普遍的规律，即所谓的“一叶知秋”、“见微知著”的意思。

回顾计算机发明的历史，每一台机器、每一颗芯片、每一种操作系统、每一类编程语言、每一个算法、每一套软件、每一款外部设备，无不像闪光的珍珠串在一起。每个案例都闪烁着智慧的火花，是创新思想不竭的源泉。在计算机科学技术领域，这样的案例就像大海岸边的贝壳，俯拾皆是。

事实上，案例研究(Case Study)是现代科学广泛使用的一种方法。Case 包含的意义很广：包括 Example 例子，Instance 事例、示例，Actual State 实际状况，Circumstance 情况、事件、境遇，甚至 Project 项目、工程等。

我们知道在计算机的科学术语中，很多是直接来自日常生活的。例如 Computer 一词早在 1646 年就出现于古代英文字典中，但当时它的意义不是“计算机”而是“计算工人”，即专门从事简单计算的工人。同理，Printer 当时也是“印刷工人”而不是“打印机”。正是

由于这些“计算工人”和“印刷工人”常出现计算错误和印刷错误，才激发查尔斯·巴贝奇(Charles Babbage, 1791—1871)设计了差分机和分析机，这是最早的专用计算机和通用计算机。这位英国剑桥大学数学教授、机械设计专家、经济学家和哲学家是国际公认的“计算机之父”。

20 世纪 40 年代，人们还用 Calculator 表示计算机。到电子计算机出现后，才用 Computer 表示计算机。此外，硬件(Hardware)和软件(Software)来自销售人员。总线(Bus)就是公共汽车或大巴，故障和排除故障源自格瑞斯·霍普(Grace Hopper, 1906—1992)发现的“飞蛾子”(Bug)和“抓蛾子”或“抓虫子”(Debug)。其他如鼠标、菜单……不胜枚举。至于哲学家进餐问题，理发师睡觉问题更是操作系统文化中脍炙人口的经典。

以计算机为核心的信息技术，从一开始就与应用紧密结合。例如，ENIAC 用于弹道曲线的计算，ARPANET 用于资源共享以及核战争时的可靠通信。即使是非常抽象的图灵机模型，也受到二战时图灵博士破译纳粹密码工作的影响。

在信息技术中，既有许多成功的案例，也有不少失败的案例；既有先成功而后失败的案例，也有先失败而后成功的案例。好好研究它们的成功经验和失败教训，对于编写案例型教材有重要的意义。

我国正在实现中华民族的伟大复兴，教育是民族振兴的基石。改革开放以来，我国高等教育在数量上、规模上已有相当的发展。当前的重要任务是提高培养人才的质量，必须从学科知识的灌输转变为素质与能力的培养。应当指出，大学课堂在高新技术的武装下，利用 PPT 进行的“高速灌输”、“翻页宣科”有愈演愈烈的趋势，我们不能容忍用“技术”绑架教学，而是让教学工作乘信息技术的东风自由地飞翔。

本系列教材的编写，以学生就业所需的专业知识和操作技能为着眼点，在适度的基础知识与理论体系覆盖下，突出应用型、技能型教学的实用性和可操作性，强化案例教学。本套教材将会有机融入大量最新的示例、实例以及操作性较强的案例，力求提高教材的趣味性和实用性，打破传统教材自身知识框架的封闭性，强化实际操作训练，使本系列教材做到“教师易教，学生乐学，技能实用”。有了广阔的应用背景，再造计算机案例型教材就有了基础。

我相信北京大学出版社在全国各地高校教师的积极支持下，精心设计，严格把关，一定能够建设出一批符合计算机应用型人才培养模式的、以案例型为创新点和兴奋点的精品教材，并且通过一体化设计、实现多种媒体有机结合的立体化教材，为各门计算机课程配齐电子教案、学习指导、习题解答、课程设计等辅导资料。让我们用锲而不舍的毅力，勤奋好学的钻研，向着共同的目标努力吧！

刘瑞挺教授 本系列教材编写指导委员会主任、全国高等院校计算机基础教育研究会副会长、中国计算机学会普及工作委员会顾问、教育部考试中心全国计算机应用技术证书考试委员会副主任、全国计算机等级考试顾问。曾任教育部理科计算机科学教学指导委员会委员、中国计算机学会教育培训委员会副主任。PC Magazine《个人电脑》总编辑、CHIP《新电脑》总顾问、清华大学《计算机教育》总策划。

前 言

C#语言是 Microsoft 公司为推行 .NET 战略而发布的一种先进的、简单的、面向对象的编程语言。在 Visual Studio .NET 框架下使用 C#语言,不仅可以编写 Windows 应用程序、数据库应用程序、Web 应用程序,还可以进行组件开发、多线程开发等。

C#语言是纯粹的面向对象语言,简单易学,只要有一些 C/C++/Java 程序设计基础,就可以快速上手;即使没有任何语言基础,也可以快速入门。因此,越来越多的高校选用 C#语言讲授面向对象程序设计课程。

本书编者都是长期在一线从事 C#程序设计教学工作的教师,有着丰富的教学经验和编程经验。编者力图从普通院校本科学生的实际出发,结合实际案例,深入浅出地对 C#语言基础及面向对象编程的理论、思想和方法进行讲解,使学生掌握 C#基本语法,掌握面向对象的编程思想和良好的编程风格,为成长为一名高层次的计算机软件专业人才打下夯实的基础。

全书由 12 章组成,第 1 章介绍 C#语言概述,第 2 章介绍 C#语法基础,第 3 章介绍类,第 4 章介绍继承和多态,第 5 章介绍接口、委托和事件,第 6 章介绍集合和泛型,第 7 章介绍异常处理,第 8 章介绍 Windows 应用程序及常用控件,第 9 章介绍图形图像编程,第 10 章介绍文件和流,第 11 章介绍 C#数据库编程,第 12 章介绍 C# Web 应用程序基础。

与同类教材相比,本书具有以下特色:

(1) 内容难易适中。市面上的同类教材,有些是高职高专教材,对本科生来说,内容过于简单;有些是针对有一定编程经验的程序员编写的,内容较深,非常全面,章节较多,不适合本科生的教学。本书在内容安排上难易适中,讲解最基本的、最常用的编程技术,有效避免这两种情况,编写真正适合应用型本科专业的 C#面向对象程序设计教材。

(2) 案例学习。本书除了有丰富的实例代码外,每章都安排了一个实际应用案例,通过使用本章所学知识完成项目,不但加深了对所学内容的理解,而且可以逐步掌握面向对象的编程思想和良好的编程风格,实现了理论知识与实践能力的无缝结合。

(3) 丰富的课后习题和解答。每章最后精编了课后习题,供读者进一步巩固所学知识。

本书可作为高等院校计算机、信息管理等相关专业的 C#程序设计课程的教材,也可作为软件开发人员的参考用书。对于初学者,建议从第一章按顺序系统地学习各个章节的内容,对于已学习过 C#语言基础知识的读者可挑选其中的某些篇章学习。在本科教学中可安排 72 学时,建议以每周 4 学时的进度进行,18 周可完成,每章结束时做适当的小结和复习。

本书由北京中医药大学唐燕、韩爱庆、王丽老师编写,其中第 1 章、第 2 章和第 3 章由王丽老师编写,第 4 章、第 5 章、第 7 章和第 8 章由韩爱庆老师编写,其余章节由唐燕老师编写。全书由唐燕、韩爱庆老师统稿。

在此,感谢北京大学出版社的编辑人员对本书认真细致地审稿。作者在编写本书的过

程中，查阅参考了大量的资料，诚恳地感谢这些资料的作者们无私的奉献；编写本书时，花费了大量的时间，感谢家人对我们工作的理解和支持！

教学 PPT 和所有实例代码下载网址：<http://www.pup6.cn>。

由于编者水平有限、时间仓促，不足之处在所难免，敬请读者指正，以激励我们继续为广大读者编写更加优秀的教材。编者电子邮箱为 tangyan18@yahoo.cn。

编 者

2012 年 6 月

目 录

第 1 章 C#语言概述..... 1

- 1.1 .NET 平台和 C#语言简介..... 1
 - 1.1.1 .NET Framework 概述..... 1
 - 1.1.2 C#语言..... 2
- 1.2 Visual Studio 2010 开发环境..... 3
 - 1.2.1 Visual Studio 2010 的新特性..... 3
 - 1.2.2 安装 Visual Studio 2010..... 4
 - 1.2.3 Visual Studio 2010 集成开发环境..... 7
- 1.3 创建 C#程序..... 11
 - 1.3.1 创建控制台应用程序..... 11
 - 1.3.2 C#程序基本结构..... 13
 - 1.3.3 C#程序的基本调试..... 15
- 习题..... 17

第 2 章 C#语法基础..... 18

- 2.1 数据类型..... 19
 - 2.1.1 变量和常量..... 19
 - 2.1.2 简单值类型..... 22
 - 2.1.3 复合值类型..... 26
 - 2.1.4 字符串..... 30
 - 2.1.5 类型转换..... 31
- 2.2 运算符和表达式..... 35
 - 2.2.1 算术运算符和表达式..... 36
 - 2.2.2 赋值运算符和表达式..... 38
 - 2.2.3 关系运算符和表达式..... 39
 - 2.2.4 逻辑运算符和表达式..... 40
 - 2.2.5 位运算符和表达式..... 41
 - 2.2.6 条件运算符和表达式..... 41
 - 2.2.7 其他运算符..... 42
- 2.3 程序控制语句..... 44
 - 2.3.1 选择结构..... 44
 - 2.3.2 循环结构..... 50

- 2.3.3 跳转结构..... 55
- 2.4 数组..... 59
 - 2.4.1 一维数组..... 60
 - 2.4.2 二维数组..... 63
 - 2.4.3 交错数组..... 65
- 2.5 案例..... 68
- 习题..... 71

第 3 章 类..... 75

- 3.1 类和对象..... 76
- 3.2 类的概述..... 76
- 3.3 构造函数和析构函数..... 80
 - 3.3.1 构造函数..... 80
 - 3.3.2 析构函数..... 82
- 3.4 常量和字段..... 83
- 3.5 类的属性..... 85
- 3.6 类的方法..... 89
- 3.7 索引器..... 99
- 3.8 案例..... 100
- 习题..... 105

第 4 章 继承和多态..... 110

- 4.1 类的继承..... 110
 - 4.1.1 派生类的声明格式..... 110
 - 4.1.2 隐藏基类方法..... 112
 - 4.1.3 base 关键字..... 113
- 4.2 重写和多态性..... 114
 - 4.2.1 重写..... 114
 - 4.2.2 多态性..... 115
- 4.3 抽象类和抽象方法..... 116
- 4.4 密封类和密封方法..... 118
- 4.5 静态类和静态方法..... 119
- 4.6 案例..... 120
- 习题..... 122

第5章 接口、委托和事件 126

5.1 接口	126
5.1.1 接口的声明和实现	126
5.1.2 显式接口实现	128
5.1.3 接口映射	129
5.2 委托	130
5.2.1 委托的声明和实现	130
5.2.2 多播委托	131
5.3 事件	132
5.4 案例	134
习题	135

第6章 集合和泛型 139

6.1 集合	139
6.1.1 ArrayList 类	140
6.1.2 Hashtable 类	142
6.1.3 Stack 类	145
6.1.4 Queue 类	147
6.2 泛型	148
6.2.1 List<T>类	149
6.2.2 Dictionary<K,V>类	151
6.3 案例	153
习题	158

第7章 异常处理 159

7.1 异常处理的概念	159
7.1.1 异常控制机制	160
7.1.2 使用 try...catch 语句	160
7.1.3 使用 finally 语句	161
7.1.4 使用 throw 语句	162
7.2 .NET Framework 中的异常类	163
7.3 案例	165
习题	166

第8章 Windows 应用程序及常用
控件 168

8.1 Windows 应用程序的基本结构	168
-----------------------------	-----

8.1.1 最简单的 Windows 应用程序	168
----------------------------------	-----

8.1.2 Windows 应用程序 项目组成	170
----------------------------------	-----

8.1.3 Windows 编程基础	170
--------------------------	-----

8.2 标签控件	174
----------------	-----

8.3 文本框控件	175
-----------------	-----

8.4 富文本框控件	177
------------------	-----

8.5 按钮控件	179
----------------	-----

8.6 单选按钮和复选框	180
--------------------	-----

8.7 列表框控件	183
-----------------	-----

8.8 组合框控件	186
-----------------	-----

8.9 图片框控件	188
-----------------	-----

8.10 计时器控件	190
------------------	-----

8.11 菜单栏控件	191
------------------	-----

8.12 工具栏	193
----------------	-----

8.13 MDI 界面设计	196
---------------------	-----

8.13.1 创建 MDI 主窗体和 子窗体	196
---------------------------------	-----

8.13.2 主窗体和子窗体程序设计	197
--------------------------	-----

8.14 通用对话框	198
------------------	-----

8.14.1 消息框	198
------------------	-----

8.14.2 打开文件和保存文件 对话框	199
-------------------------------	-----

8.14.3 颜色对话框	201
--------------------	-----

8.14.4 字体对话框	201
--------------------	-----

8.15 案例	203
---------------	-----

习题	207
----------	-----

第9章 图形图像编程 210

9.1 GDI+ 概述	211
-------------------	-----

9.2 Graphics 类	211
----------------------	-----

9.2.1 使用 Graphics 类绘图的 基本步骤	211
--------------------------------------	-----

9.2.2 窗体的 Paint 事件	214
--------------------------	-----

9.2.3 常用绘图对象	214
--------------------	-----

9.3 基本图形的绘制和填充	217
----------------------	-----

9.3.1 绘制直线	217
------------------	-----

9.3.2 绘制矩形	218	10.3 使用 Directory 类和 DirectoryInfo 类 操作文件夹	264
9.3.3 绘制多边形	219	10.4 案例	269
9.3.4 绘制圆和椭圆	221	习题	272
9.3.5 绘制弧线	222	第 11 章 C#数据库编程	274
9.3.6 绘制扇形	222	11.1 ADO.NET 数据库访问	275
9.3.7 绘制曲线	223	11.1.1 ADO.NET 概述	275
9.4 常用画刷的创建及使用	225	11.1.2 设计数据库应用程序的基本 步骤	277
9.4.1 SolidBrush 类	225	11.2 ADO.NET 数据访问对象	278
9.4.2 HatchBrush 类	225	11.2.1 Connection 对象	278
9.4.3 TextureBrush 类	227	11.2.2 Command 对象	280
9.4.4 LinearGradientBrush 和 PathGradientBrush 类	228	11.2.3 DataReader 对象	286
9.5 绘制文本	229	11.2.4 DataAdapter 对象	288
9.6 Bitmap 类	230	11.2.5 DataSet 对象	289
9.7 图像的处理	231	11.3 C#数据库的 Windows 编程	298
9.7.1 显示图像	231	11.3.1 BindingSource 控件	298
9.7.2 保存图像	232	11.3.2 数据绑定控件	300
9.7.3 彩色图片变为黑白图片	233	11.3.3 BindingNavigator 控件	301
9.7.4 图片的翻转和旋转	235	11.3.4 DataGridView 控件	302
9.8 案例	237	11.4 ADO.NET 访问常用数据库	305
习题	242	11.5 案例	306
第 10 章 文件和流	244	习题	324
10.1 用流读写文件	245	第 12 章 C# Web 应用程序基础	326
10.1.1 FileStream 类读写字节	245	12.1 ASP.NET 概述	327
10.1.2 BinaryReader 类和 BinaryWriter 类读写基本数据类型	249	12.2 ASP.NET 的开发环境配置	327
10.1.3 StreamReader 类和 StreamWriter 类读写字符串	252	12.3 编写 ASP.NET Web 应用程序	330
10.1.4 Stream 类的其他派生类	255	12.3.1 第一个 ASP.NET Web 应用程序	330
10.2 使用 File 类和 FileInfo 类 操作文件	255	12.3.2 ASP.NET Web 应用程序的 结构	332
10.2.1 File 类的常用方法	256	12.4 ASP.NET 技术基础	332
10.2.2 复制文件	256	12.4.1 使用记事本编写 动态网页	332
10.2.3 移动文件	258	12.4.2 ASP.NET 的页面 生命周期	334
10.2.4 删除文件	259		
10.2.5 设置文件的属性	260		
10.2.6 获得文件的属性	262		

12.4.3 ASP.NET 的事件处理	334	12.6 案例	357
12.4.4 Code Behind 技术	335	习题	360
12.5 Web 服务器控件	336	参考文献	362

C#语言概述

教学目标

- 了解 .NET Framework 的构成
- 了解 C# 的概念及特点
- 了解 Visual Studio 2010 开发环境
- 掌握使用 C# 创建简单的控制台应用程序
- 掌握 C# 应用程序的基本结构
- 掌握 C# 程序的基本调试方法

1.1 .NET 平台和 C# 语言简介

1.1.1 .NET Framework 概述

.NET 是 Microsoft 公司创建的适合网络编程和网络服务的开发平台。平台内部封装了大量的应用程序编程接口(Application Programming Interface, API), 通过这些函数可以开发各类 Windows 应用程序, 同时支持标准的 Internet 协议, 可以实现应用程序在不同平台上沟通的功能。

.NET 的核心是 .NET Framework, 又称 .NET 框架, 它提供了建立和运行 .NET 应用程序所需要的编辑、编译等核心服务。

.NET Framework 具有两个主要组件: 公共语言运行时(Common Language Runtime, CLR) 和 .NET Framework 类库。公共语言运行时是 .NET Framework 的基础, 它提供了程序代码可以跨平台运行的机制。CLR 可看做一个在执行时管理代码的代理, 它提供内存管理、线程管理和远程处理等核心服务, 并且强制实施严格的类型安全以及可提高安全性和可靠性的其他形式的代码。.NET Framework 的另一个主要组件是类库, 它提供了一个可以被多种 .NET 编程语言调用的、与 CLR 紧密集成的、面向对象的可重用类型集合。通过类库可以开发多种应用程序, 这些应用程序包括传统的命令行或图形用户界面(Graphical User Interface, GUI)应用程序, 也包括基于 ASP.NET 所提供的最新应用程序(如 Web 窗体和 XML Web Services)。

.NET Framework 可以实现以下的功能。

(1) 提供一个一致的面向对象的编程环境，而不管对象代码是在本地存储和执行，还是在本地执行但在 Internet 上分布，或是在远程执行。

(2) 提供一个将软件部署和版本控制冲突最小化的代码执行环境。

(3) 提供一个可提高代码(包括由未知的或不完全受信任的第三方创建的代码)执行安全性的代码执行环境。

(4) 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。

(5) 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。

(6) 按照工业标准生成所有通信，以确保基于 .NET Framework 的代码可与任何其他代码集成。

1.1.2 C#语言

C#(C Sharp)是 Microsoft 公司在 2000 年 6 月发布的专门为 .NET 应用而开发的一种面向对象的程序设计语言。C#从 C/C++语言发展而来，与 Java 也有很多相似之处。C#既继承了 C 语言的语法风格，又继承了 C++的面向对象的特性，这使得它容易被已掌握 C/C++语言的学习者接受，同时它简洁的语法和快速的应用开发功能对于初学者来说更容易被吸引和接受。C#还综合了 Visual Basic 简单的可视化操作的优点，以及 Delphi 与 COM(Component Object Model, 组件对象模型)直接集成的特点，程序员可以快速地编写各种基于 .NET 平台的应用程序。而且 Microsoft.NET 提供了一系列的工具和服务来最大限度地应用于计算与通信领域。

1. C#的特点

(1) 简洁的语法。C#与 C、C++、Java 在很多方面有相似之处，同时 C#又对它们进行了改进。例如，语句、表达式、操作符等一些功能直接来自于 C/C++，同时它又摒弃了 C/C++ 中的指针、多父类继承等一些复杂或不常用元素。C#还提供了一些实用类型，如 decimal 类型，这为企业级的应用提供了方便。

(2) 面向对象。C#是一种面向对象的程序设计语言，它支持面向对象方法中的封装、继承和多态三种主要特性。C#将全局的概念转变成类的静态成员，可以减少命名冲突。C#语言只允许单继承，通过多个接口隐式实现多继承，从而避免类型定义的混乱。

(3) 与 Web 紧密结合。C#与 ASP.NET 的相互融合，使得程序员在面对复杂的 Web 编程时也像对本地对象进行操作一样。使用 C#语言构建的组件可以方便地为 Web 服务，并可以通过 Internet 被运行在任何操作系统的任何语言所调用。

(4) 完整的安全性和错误处理。C#语言可以消除许多软件开发中的常见错误，并提供了包括类型安全在内的完整的安全性能。C#语言的自动资源回收机制也大大减轻了程序员的负担。

(5) 版本控制技术。C#语言内置版本控制功能。通过内建语言支持和 CLR 环境的支持，可以保证当前开发的类升级时与现有客户应用保持二进制兼容，并防止其他开发者的入侵。

(6) 灵活性和兼容性。C#语言的灵活性体现在多个方面，如在 C#语言中取消了指针，

指针的功能由代理概念来模拟等。C#语言支持对 C 语言形式的 API(DLL)的交互,也支持对传统 COM 对象的访问,还支持所有的 OLE(Object Linking Embedding, 对象连接与嵌入)特性。

2. 使用 C#可以编写的程序

C#语言是 Microsoft 专门为 .NET 平台而设计的一种语言,它能够使用 .NET Framework 代码库提供的各种功能。使用 C#语言编写的常见的应用程序类型如下。

(1) Windows 应用程序。这类应用程序可以使用 .NET Framework 的 Windows Forms 模块生成,是与 Windows 界面相似的图形窗口界面。在窗体中可以添加多种类型的控件,并可以为窗体和控件编写事件处理过程。窗体项目可编译为可执行文件,可执行文件与 .NET 运行库一起安装在终端计算机上。另外,Windows 应用程序还可以使用(Windows Presentation Foundation)WPF 建立,WPF 使用 XAML(Extensible Application Markup Language,可扩展应用程序标记语言)建立应用程序。

(2) Web 应用程序。这类应用程序是可以在 Web 浏览器中查看的 Web 页面。可以使用 C#通过 Web Forms 创建 ASP.NET 应用程序,也可以使用 Silverlight 编写在浏览器内部运行的应用程序。

(3) Web 服务。这是创建各种分布式应用程序的新方式,使用 Web 服务可以通过 Internet 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,都使用一种简单的语法。对于高级功能,还可以创建 WCF(Windows Communication Foundation)服务。

1.2 Visual Studio 2010 开发环境

Visual Studio 2010 是由 Microsoft 推出的一个强大的集成开发环境(Integrated Development Environment, IDE),从 1998 年的 Visual Studio 6.0 到 Visual Studio 2010 无论是在界面上还是在功能上都有了显著的不同和改进。Visual Studio 2010 支持多种开发语言,Visual Basic、Visual C++、Visual C#和 Visual F#都可以使用。

除 Visual Studio 2010 外,Microsoft 还提供 Visual Studio 2010 Express 产品,在其官方网站中可以免费获得。Visual Studio 2010 Express 产品在功能上可以看作 Visual Studio 2010 的删节版本,它们的外观和操作与 Visual Studio 2010 是相同的。其中,Visual C# 2010 Express(VCE)和 Visual Web Developer 2010 Express 也可以用来进行 C#应用程序的创建。

1.2.1 Visual Studio 2010 的新特性

Visual Studio 2010 支持开发面向 Windows 7 的应用程序,还支持 DB2 和 Oracle 数据库。在功能上的改进和增加主要体现在以下几个方面。

- (1) C# 4.0 中的动态类型和动态编程。
- (2) 支持多显示器。
- (3) 使用 Visual Studio 2010 的特性支持 TDD。
- (4) 支持 Office。
- (5) Quick Search 特性。

- (6) C++ 0x 新特性。
- (7) IDE 增强。
- (8) 使用 Visual C++ 2010 创建 Ribbon 界面。
- (9) 新增基于 .NET 平台的语言 F#。

Visual Studio 2010 包含了创建 C#项目所需的全部功能，下面介绍 Visual Studio 2010 的安装和启动过程。

1.2.2 安装 Visual Studio 2010

在安装 Visual Studio 2010 时，要考虑系统硬件和软件环境需求，下面给出 Visual Studio 2010 安装环境需求。

- (1) CPU：最低要求 1.6GHz 主频，建议 2.6GHz。
- (2) 内存：最低要求 512MB，建议 1GB。
- (3) 硬盘：最低要求 3GB，建议 20GB。
- (4) 操作系统：Windows Server 2003/2008，Windows Vista，Windows XP，Windows 7。
- (5) 平台架构：32 位，64 位。
- (6) Web 服务器：IIS5.0 或以上版本。
- (7) IE：IE 6 或以上版本。
- (8) 运行环境：Microsoft .NET Framework SDK。
- (9) 开发环境：Microsoft Visual Studio.NET。

在 Microsoft 的官方网站中提供了 Visual Studio 2010 的多种版本，本书以“Visual Studio 2010 专业版”为例进行演示。

(1) 打开 Visual Studio 2010 的安装文件，双击“setup.exe”，运行安装程序，如图 1.1 所示。



图 1.1 Visual Studio 2010 安装程序界面

(2) 选择“安装 Microsoft Visual Studio 2010”选项，启动 Microsoft Visual Studio 2010 安装向导，下载安装组件，如图 1.2 所示。

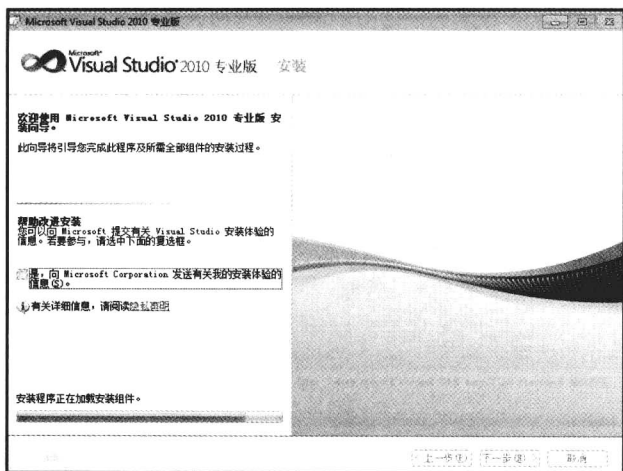


图 1.2 Visual Studio 2010 专业版安装向导

(3) 安装组件加载完成后,单击“下一步”按钮,进入安装程序的起始页,如图 1.3 所示。在这个界面中点选“我已阅读并接受许可条款”单选按钮,然后单击“下一步”按钮,进入如图 1.4 所示的安装程序的选项页。

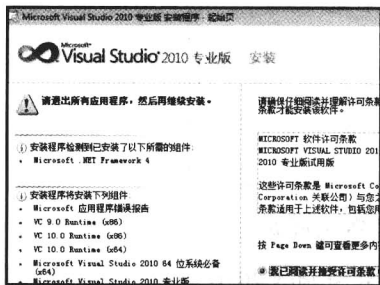


图 1.3 Visual Studio 2010 安装程序-起始页

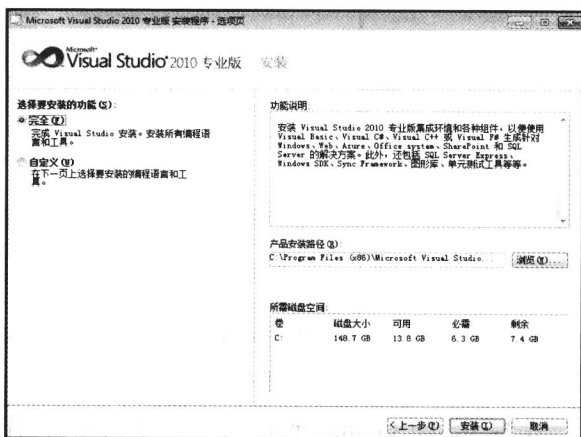


图 1.4 Visual Studio 2010 安装程序-选项页

(4) 在图 1.4 中,右侧窗格允许用户选择安装路径,左侧窗格需要用户选择完全安装还是自定义安装。如果点选“完全”单选按钮,则会将 Visual Studio 2010 的全部组件都在本机中安装;如果点选“自定义”单选按钮,则会由用户选择要安装的组件。如果用户选择完全安装,可以单击“安装”按钮,进入如图 1.5 所示的安装程序的安装页;如果用户选择自定义安装,则单击“下一步”按钮,进入选择安装功能的界面,在自定义安装界面中,用户可以在左侧勾选相关复选框选择要安装的组件,然后单击“安装”按钮,进入如图 1.5 所示的安装程序的安装页开始安装。

(5) 安装完成后会进入如图 1.6 所示的安装程序的完成页。单击“完成”按钮, Visual Studio 2010 安装成功。