



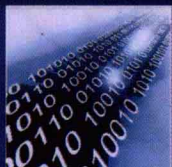
21st CENTURY
实用规划教材

21世纪全国应用型本科计算机案例型规划教材



C#程序开发案例教程

李挥剑 陈小全 钱 哨 主 编



Visual studio 2008



- 注重应用，从项目应用出发，解决实际开发中的常见问题
- 案例丰富，提供完整的案例，讲解设计以及代码实现过程
- 结构严紧，从读者角度思考，便于读者自学、自练、自测



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#程序开发案例教程

主 编 李挥剑 陈小全 钱 哨
副主编 李继哲 林 航 江冰冰 徐 静



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书采用案例教学的模式来讲解 C#语言的编程环境、基本语法、数据类型、面向对象编程、Winform 开发、Web 开发、文件操作、数据访问等技术知识。书中所涉及的案例和程序全部以 Visual Studio 2008 作为开发环境,并采用 C#语言开发,主要涉及“C#.NET 基础”、“.NET Framework”、“Winform 编程”、“ASP.NET”和“ADO.NET”五个大方向的基础知识和技术要点。

本书可供高等院校基于.NET 方向软件开发专业的学生使用,也可供从事.NET 方向软件开发的程序员参考使用。

图书在版编目(CIP)数据

C#程序开发案例教程/李挥剑等主编. —北京:北京大学出版社, 2012.5

(21 世纪全国应用型本科计算机案例型规划教材)

ISBN 978-7-301-20630-0

I. ①C… II. ①李… 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 090667 号

书 名: C#程序开发案例教程

著作责任者: 李挥剑 陈小全 钱 哨 主编

策划编辑: 郑 双

责任编辑: 郑 双

标准书号: ISBN 978-7-301-20630-0/TP · 1220

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@sohu.com pup_6@163.com

印 刷 者: 北京富生印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

经 销 者: 787 毫米×1092 毫米 16 开本 21 印张 483 千字

经 销 者: 2012 年 5 月第 1 版 2012 年 5 月第 1 次印刷

定 价: 39.00 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 | 王联合国 | 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 表示计算机器。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，强化实际操作的训练，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

Visual Studio.NET(通常简称.NET)作为微软新一代软件开发平台,是微软.NET 战略产品的重要部分。Visual Studio.NET 集成了 Visual Basic.NET、Visual C#.NET、Visual C++.NET、Visual J#.NET、ASP.NET 等开发环境,并且微软第一次统一了 Visual Basic 和 Visual C++的底层对象,使 Visual Basic.NET 和 Visual C#.NET 能够访问相同组件的属性和方法,使得编写 C#和编写 Visual Basic.NET 程序同样的简单和高效。

近几年,根据微软的开发战略,C#将不可避免地崛起,在 Windows 平台上成为主角,而 Visual Basic 等语言将慢慢边缘化。尤其是 Visual Studio 2008 的出现,已经成为业界中的主流开发平台。

2009 年以前的调查结果中,软件人才需求主要是以 Java 和.NET 两大平台为主,两者各有千秋。2009 年的调查结果中,.NET 人才需求增大,呈现出上升趋势。在国内的招聘网站中使用.NET 作为职位查询关键字,可以看到,仅在北京每个月需求 1000 人以上,但仍然求大于供,掌握.NET 技术就意味着进入了高薪领域!

编者从事.NET 方向教学多年,并一直辅导学生实训课程,在教学中发现很难找到一套与理论教学结合紧密又能使学生掌握足够开发经验的实训教材。为此,编者集中筛选了多年教学中使用的案例,并结合理论知识和开发经验,汇集成此书。

全书共 10 章。第 1 章介绍 C#语言;第 2、3 章介绍 C#语言的语法及数据类型;第 4、5 章介绍 C#语言的面向对象程序设计;第 6、7、8 章介绍基于 Winform 开发、Web 开发、文件操作中的 C#应用;第 9、10 章介绍 Winform 高级编程和数据访问技术。

本书由交通运输部管理干部学院李挥剑、陈小全、钱哨、李继哲、林航、江冰冰和电子工业出版社徐静编写。其中李挥剑编写第 3~5 章、第 7 章、第 8 章,陈小全编写第 9 章、第 10 章,钱哨编写第 6 章,徐静编写第 1、2 章,林航、江冰冰负责案例程序及课后习题编写和课件的制作,李继哲负责校稿。参加本书编写的还有交通运输部管理干部学院黄少波、王克难、王华、李凤、李强、张光旦、北京工业大学赖见辉、高德软件公司田伟华、叶继久等同仁。

由于时间仓促、编者水平有限,本书错漏之处在所难免,欢迎广大读者批评指正。

编 者
2012 年 3 月

目 录

| | |
|----------------------------------|----|
| 第 1 章 C#概述 | 1 |
| 1.1 初识 C#..... | 1 |
| 1.1.1 课程简介..... | 2 |
| 1.1.2 本门课程体系定位..... | 2 |
| 1.1.3 .NET 平台介绍..... | 2 |
| 1.2 开发环境概述..... | 5 |
| 1.2.1 安装 Visual Studio 2008..... | 5 |
| 1.2.2 C#的开发环境..... | 6 |
| 1.2.3 C#的特点..... | 8 |
| 1.3 第一个 C#程序..... | 8 |
| 本章小结..... | 14 |
| 课后习题..... | 14 |
| 第 2 章 C#数据类型与表达式 | 16 |
| 2.1 C#的基本语法..... | 16 |
| 2.2 基本数据类型..... | 17 |
| 2.2.1 C#数据类型的分类与区别..... | 17 |
| 2.2.2 简单类型..... | 21 |
| 2.2.3 枚举类型..... | 22 |
| 2.2.4 结构类型..... | 23 |
| 2.3 常量..... | 25 |
| 2.4 变量..... | 26 |
| 2.5 表达式..... | 28 |
| 2.5.1 算术运算符..... | 28 |
| 2.5.2 关系运算符..... | 29 |
| 2.5.3 逻辑运算符..... | 31 |
| 2.5.4 位运算符..... | 32 |
| 2.5.5 赋值运算符..... | 33 |
| 2.5.6 三元运算符..... | 35 |
| 2.5.7 运算符的优先级..... | 36 |
| 2.6 数据类型转换..... | 37 |
| 2.6.1 数据类型转换的用途..... | 37 |
| 2.6.2 数据类型的转换方法..... | 37 |
| 2.6.3 简单的数据类型的转换..... | 38 |
| 本章小结..... | 39 |
| 课后习题..... | 40 |
| 第 3 章 C#编程基础 | 41 |
| 3.1 选择语句..... | 41 |
| 3.1.1 if 语句的使用..... | 41 |
| 3.1.2 switch 语句的应用..... | 43 |
| 3.1.3 三元运算符的应用..... | 44 |
| 3.2 循环语句..... | 46 |
| 3.2.1 while 语句..... | 46 |
| 3.2.2 do...while 语句..... | 48 |
| 3.2.3 for 语句..... | 49 |
| 3.2.4 foreach 语句..... | 51 |
| 3.3 跳转语句..... | 53 |
| 3.3.1 break 语句..... | 53 |
| 3.3.2 continue 语句..... | 54 |
| 3.3.3 return 语句..... | 56 |
| 3.4 数组..... | 57 |
| 3.4.1 一维数组的声明和使用..... | 57 |
| 3.4.2 多维数组的声明和使用..... | 59 |
| 3.5 字符串..... | 60 |
| 3.6 函数..... | 63 |
| 3.6.1 值参数..... | 64 |
| 3.6.2 输入引用参数..... | 65 |
| 3.6.3 输出引用参数..... | 66 |
| 3.6.4 数组型参数..... | 67 |
| 3.6.5 局部变量与全局变量..... | 68 |
| 3.6.6 Main()函数..... | 70 |
| 3.6.7 结构函数..... | 71 |
| 3.7 综合应用实例..... | 72 |
| 本章小结..... | 74 |
| 课后习题..... | 74 |

| | | | |
|--------------------------------------|-----|--------------------------------------|-----|
| 第4章 面向对象编程基础 | 75 | 6.2.1 简介 | 167 |
| 4.1 面向对象 | 75 | 6.2.2 基本控件使用 | 167 |
| 4.1.1 面向对象的基本概念 | 75 | 6.3 菜单和菜单组件 | 180 |
| 4.1.2 类与对象 | 76 | 6.3.1 菜单和菜单组件简介 | 180 |
| 4.1.3 面向对象主要特征 | 77 | 6.3.2 菜单的实践操作 | 181 |
| 4.2 类 | 77 | 6.4 多文档界面处理 | 182 |
| 4.2.1 字段 | 78 | 6.4.1 简介 | 182 |
| 4.2.2 构造函数 | 85 | 6.4.2 多文档界面设置及 窗体属性 | 182 |
| 4.2.3 构造函数的重载 | 89 | 6.4.3 多文档界面的窗体 传值技术 | 186 |
| 4.2.4 析构函数 | 90 | 6.5 窗体界面的美化 | 190 |
| 4.3 方法 | 91 | 本章小结 | 191 |
| 4.3.1 静态方法与实例方法 | 93 | 课后习题 | 192 |
| 4.3.2 方法的重载 | 97 | 第7章 Web 应用程序开发 | 193 |
| 4.3.3 方法的重写 | 101 | 7.1 ASP.NET 简介 | 193 |
| 4.4 属性 | 103 | 7.2 使用 ASP.NET 控件 | 195 |
| 4.5 命名空间 | 105 | 7.2.1 Label 控件 | 195 |
| 本章小结 | 108 | 7.2.2 TextBox 控件 | 195 |
| 课后习题 | 108 | 7.2.3 Button 控件 | 198 |
| 第5章 深入了解 C#面向对象编程 | 110 | 7.2.4 HyperLink 控件 | 200 |
| 5.1 C#继承机制 | 110 | 7.2.5 DropDownList 控件 | 205 |
| 5.2 C#多态机制 | 117 | 7.2.6 ListBox 控件 | 208 |
| 5.2.1 方法重写 | 118 | 7.2.7 CheckBox 控件 | 211 |
| 5.2.2 方法的隐藏 | 120 | 本章小结 | 214 |
| 5.2.3 抽象类和抽象方法 | 122 | 课后习题 | 214 |
| 5.3 操作符重载 | 122 | 第8章 文件处理技术 | 216 |
| 5.4 接口 | 127 | 8.1 System.IO 命名空间 | 216 |
| 5.5 委托 | 137 | 8.1.1 System.IO 类介绍 | 216 |
| 5.6 事件 | 141 | 8.1.2 File 类的常用方法 | 217 |
| 5.7 索引器 | 145 | 8.1.3 FileInfo 类的常用方法 | 220 |
| 5.8 异常处理 | 150 | 8.1.4 文件夹类 Directory 的常用 方法 | 223 |
| 5.9 组件与程序集 | 159 | 8.1.5 DirectoryInfo 类的常见 属性 | 226 |
| 本章小结 | 159 | 8.2 FileStream 文件流类 | 230 |
| 课后习题 | 160 | 8.2.1 FileStream 文件流类简介 | 230 |
| 第6章 Windows 编程基础 | 162 | | |
| 6.1 Windows 和窗体的基本概念 | 162 | | |
| 6.1.1 Windows Forms 程序基本 结构 | 162 | | |
| 6.1.2 了解 Winform 程序的代码 结构 | 163 | | |
| 6.2 Winform 中的常用控件 | 167 | | |

| | | | | | |
|--------------------------------|--------------------------------------|------------|------------------------------------|-----------------------------|------------|
| 8.2.2 | FileStream 文件流类常见属性和方法..... | 230 | 9.10 | CheckedListBox 可选列表框控件..... | 274 |
| 8.2.3 | FileStream 文件流类的创建..... | 231 | 9.11 | NumericUpDown 按钮控件..... | 277 |
| 8.3 | 文本文件的流操作..... | 235 | 9.12 | MonthCalendar 控件..... | 279 |
| 8.3.1 | StreamReader 和 StreamWriter 类简介..... | 235 | 9.13 | DateTimePicker 控件..... | 282 |
| 8.3.2 | StreamReader 类常见方法..... | 235 | 9.14 | 为程序添加多媒体功能..... | 284 |
| 8.3.3 | StreamWriter 类常见属性和方法..... | 237 | | 本章小结..... | 286 |
| 8.4 | 读写二进制文件..... | 240 | | 课后习题..... | 286 |
| 8.4.1 | 二进制文件操作..... | 240 | 第 10 章 ADO.NET 数据库访问技术..... | | |
| 8.4.2 | BinaryReader 类介绍..... | 240 | 10.1 | ADO.NET 简介..... | 288 |
| 8.4.3 | BinaryWriter 类介绍..... | 242 | 10.1.1 | ADO.NET 的主要对象..... | 289 |
| | 本章小结..... | 245 | 10.1.2 | ADO.NET 对象的关系..... | 290 |
| | 课后习题..... | 245 | 10.2 | ADO.NET 的对象的的使用..... | 291 |
| 第 9 章 Windows 高级控件..... | | 247 | 10.2.1 | Connection 对象..... | 291 |
| 9.1 | RadioButton..... | 247 | 10.2.2 | Command 对象..... | 294 |
| 9.2 | PictureBox 控件..... | 250 | 10.2.3 | DataReader 对象..... | 298 |
| 9.3 | TabControl 控件..... | 252 | 10.2.4 | DataAdapter 对象..... | 300 |
| 9.4 | ProgressBar 控件..... | 255 | 10.2.5 | DataSet 对象..... | 303 |
| 9.5 | ImageList 控件..... | 257 | 10.3 | DataGridView 控件..... | 310 |
| 9.6 | StatusStrip 控件..... | 260 | 10.3.1 | DataGridView 控件概述..... | 310 |
| 9.7 | Timer 控件..... | 263 | 10.3.2 | DataGridView 控件与存储过程..... | 315 |
| 9.8 | ListView 控件..... | 265 | | 本章小结..... | 318 |
| 9.9 | TreeView 控件..... | 271 | | 课后习题..... | 318 |
| | | | 参考文献..... | | 320 |

C# 概述

本章重点介绍 C# 开发语言所需要的基础知识, 如 .NET Framework 的体系结构、C# 语言特点、Visual Studio 2008 集成开发环境的安装及 Visual Studio 2008 集成开发环境初识。通过简要的介绍、简单实例和详细的步骤, 让初学 C# 的读者快速地了解学习该语言所必需的基础知识。

学习目标

- (1) 了解 Microsoft .NET Framework
- (2) 了解 C# 的特点和开发环境
- (2) 掌握 Visual Studio 2008 开发工具的安装
- (3) 使用 Visual Studio 2008 开发工具
- (4) 编写第一个应用程序

1.1 初识 C#

C#(标准读音为 C sharp)是一种可以用于创建运行在 .NET CLR 上的编程语言, 是专门为 .NET Framework(.NET 框架)而设计的。C# 是从 C 和 C++ 语言派生衍化而来, 吸收了 C 和 C++ 的优点, 并解决了它们的问题, 从而产生的一种简单、功能强大、类型安全, 而且是面向对象的语言。C# 凭借其诸多创新, 在保持 C 样式语言的表示形式和优美特点的同时, 实现了应用程序的快速开发。由于 C# 使用 .NET Framework, 所以没有限制应用的类型。在这里简单了解几个常见的应用程序类型。

- Windows 应用程序, 如 QQ、Microsoft Office 2010。
- Web 应用程序, 如同 Web 浏览器查看的 ASP.NET 网页。
- Web 服务, 如创建各种各样的分布式应用程序。
- 智能设备应用程序, 如手机应用软件。

使用 Visual Studio 2008 为 C# 开发环境, 是通过 Visual Studio 2008 功能齐全的代码编辑器、项目模板、设计器、代码向导、功能强大且易于使用的调试器以及其他工具实现的。通过 .NET Framework 类库, 可以访问多种操作系统服务和其他有用的、精心设计的类, 从而大大缩短应用程序开发的周期。

本书通过 Visual Studio 2008 集成开发环境构建了大量 C# 应用程序实例, 向读者详细阐

述 C#的编程基础和抽象的软件设计思想,使读者能够更快地进入 C#程序设计领域,做到对理论知识的深刻理解和对 C#编程方法的熟练掌握。

1.1.1 课程简介

课程定位目标如下。

- 高等院校计算机相关专业。
- 基于 Visual Studio 2008 开发环境。
- 使用 C#语言开发的应用程序。
- 通过本课程学习,能够熟练地使用 C#开发应用程序。

开设本门课程的先修课程包括 C 程序设计基础、数据结构及算法、数据库基础理论、面向对象的程序设计等,要求学生能够编写简单的符合软件标准的规范代码。

学习完本门课程,学生应掌握以下基本知识点:C#数据类型与表达式,C#语言各类语句,数组、字符串、函数,面向对象编程基础,C#继承、多态机制,结构类型、枚举类型、接口、委托、事件、索引器,异常处理,组件,程序集,Windows 应用程序开发,文件操作与管理,Web 应用程序开发。

1.1.2 本门课程体系定位

本门课程绝非孤立存在,课程的开设必须建立在一整套课程体系的基础之上,具体课程体系定位如图 1.1 所示。

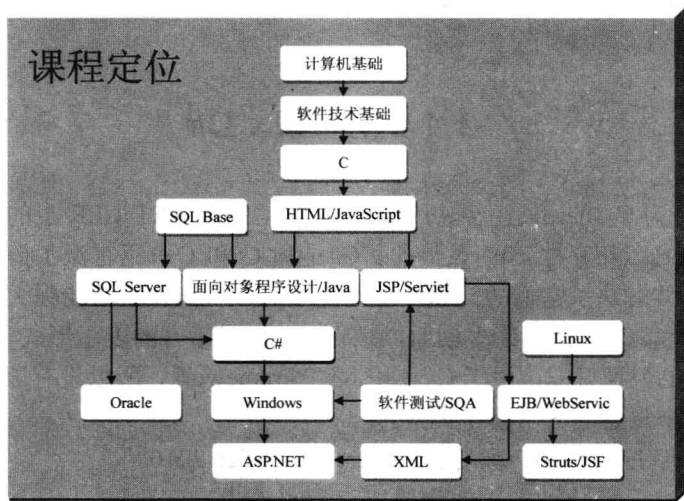


图 1.1 C#程序设计在课程体系中的地位

根据图 1.1, C#程序设计在整体课程体系中处于基础地位, C#程序设计是其他开发工具的基础,为 Winform 程序设计和 Web 程序设计提供语言支持。因而学好本门课程对于软件技术专业的学生意义重大。

1.1.3 .NET 平台介绍

.NET Framework 是一个富有革命性的新平台,在 .NET Framework 下可以创建 Windows

程序、Web 程序、智能设备程序(如手机应用程序)和其他各类程序。同时在 .NET Framework 下可以使用各种语言进行开发应用程序,如本书前边所提到的 C#、C++、Visual Basic 等。这主要得益于 .NET Framework 是一种采用系统虚拟机运行,以公共语言运行时(Common Language Runtime, CLR)为基础的编程平台。

.NET Framework 旨在实现下列目标。

- 提供一个一致的面向对象的编程环境,而无论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,或者是在远程执行,都可以使用。
- 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。
- 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。
- 提供一个可消除脚本环境或解释环境性能问题的代码执行环境。
- 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。
- 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。

.NET Framework 两个主要组件如下。

- CLR。
- 统一的类库集。

由上述描述可知,CLR 是 .NET Framework 的基础。用户可以将运行库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准确性。事实上,代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码;而不以运行库为目标的代码称为非托管代码。

.NET Framework 的另一个主要组件是类库集。它是一个综合性的面向对象的 reusable 类型集合,用户可以使用它开发多种应用程序。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 所提供的最新创新的应用程序(如 Web 窗体和 XML Web Services)。类库集包含的类库有线程、文件输入/输出(I/O)、数据库支持、XML 解析、数据结构等。

Microsoft .NET Framework 的体系结构如图 1.2 所示。可以看到, .NET Framework 基于操作系统之上,是 C# 等各类语言程序运行的支撑架构。其中类库可以被各类语言调用。Microsoft .NET Framework 的组件如图 1.3 所示。

公共语言规范(Common Language Specification, CLS)是一组定义了一种语言的边界的标准,或者说是一种语言(与 CLS 兼容的)必须支持或遵循的一系列语言功能,从而使得其可以与其他的 .NET 语言互操作。

通用类型系统(Common Type System, CTS)用于解决不同语言的数据类型不同的问题,确保这些语言可以相互传送数据,体现了 .NET 语言之间的无缝互操作。

在对 .NET Framework 的学习中,需要了解微软中间语言(Microsoft Intermediate Language, MSIL)和实时编译器(Just-In-Time, JIT)。MSIL 是一种介于高级语言和基于 Intel 的汇编语言的伪汇编语言。在编译和使用 FCL 的代码时,会先将代码编译为 MSIL 代码,产生的 MSIL

代码不属于任何专属的操作系统或者任何专属语言，它是一种中间语言。在.NET的世界中可能出现下面的情况，即一部分代码可以用 EFFIL 实现；另一部分代码使用 C#或 Visual Basic.NET 完成，但是最后这些代码都将被转换为中间语言。这为程序员提供了极大的灵活性，程序员可以选择自己熟悉的语言，并且再也不用为学习不断推出的新语言而烦恼。

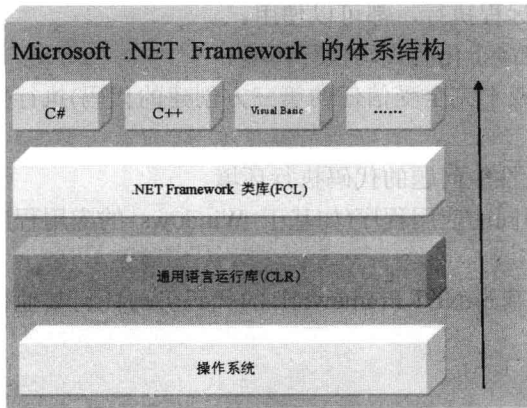


图 1.2 Microsoft .NET Framework 的体系结构

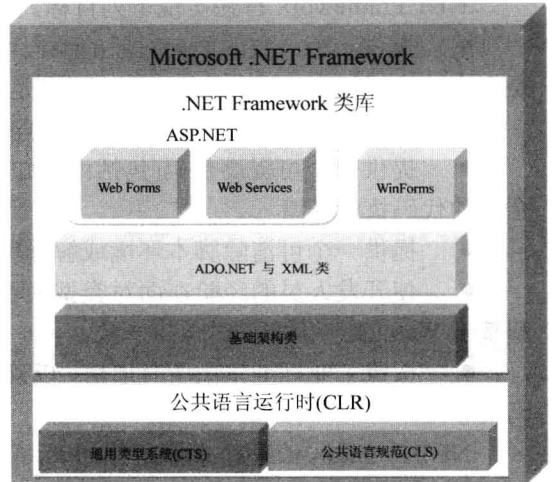


图 1.3 Microsoft .NET Framework 的组件

编译为托管代码时，编译器将源代码翻译为 MSIL，这是一组可以有效地转换为本机代码且独立于 CPU 的指令。MSIL 包括用于加载、存储和初始化对象以及对对象调用方法的指令，还包括用于算术和逻辑运算、控制流、直接内存访问、异常处理和其他操作的指令。要使代码可运行，必须先将 MSIL 转换为特定于 CPU 的代码，通常是通过 JIT 编译器来完成的。由于 CLR 为它支持的每种计算机结构都提供了一种或多种 JIT 编译器，因此同一组 MSIL 可以在所支持的任何结构上进行 JIT 编译和运行。

为了更清楚地了解 CLR 的作用，下面看一下模拟 CLR 和 MSIL 示意图，如图 1.4 所示。CLR 和 MSIL 的工作机理如图 1.5 所示。

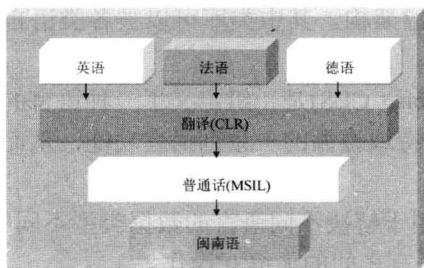


图 1.4 模拟 CLR 和 MSIL 示意图

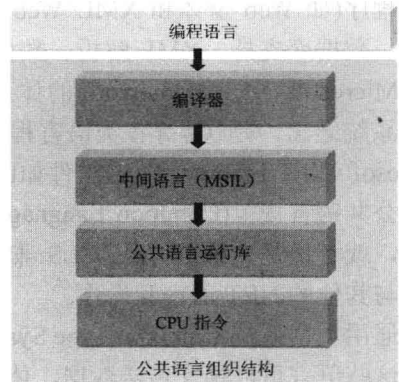


图 1.5 CLR 和 MSIL 的工作机理

- 有了 CLR，保证了.NET 中一种语言具有的功能，其他语言也都具有。
- MSIL 由一组特定的指令组成，这些指令指明如何执行代码。

- JIT 编译器的主要工作是将普通 MSIL 代码转换为可以直接由 CPU 执行的计算机代码。
- 验证进程可以轻松读取 MSIL 代码。

1.2 开发环境概述

1.2.1 安装 Visual Studio 2008

Visual Studio 2008 的安装步骤如下。

(1) 将 Visual Studio 2008 安装光盘放入光驱中，光盘自动播放，弹出 Visual Studio 2008 安装界面，如图 1.6 所示。(如果光盘未自动播放，打开光盘目录，运行 setup.exe 安装程序)

(2) 单击“安装 Visual Studio 2008”链接，进入 Visual Studio 2008 安装向导界面，如图 1.7 所示。



图 1.6 Visual Studio 2008 安装界面



图 1.7 Visual Studio 2008 安装向导界面

(3) 单击“下一步”按钮继续安装，进入如图 1.8 所示的界面，点选“我已阅读并接受许可条款”单选按钮，单击“下一步”按钮继续安装。

(4) 进入如图 1.9 所示的界面，选择安装方式和安装路径，可以查看所需的磁盘空间大小，单击“下一步”按钮继续安装。

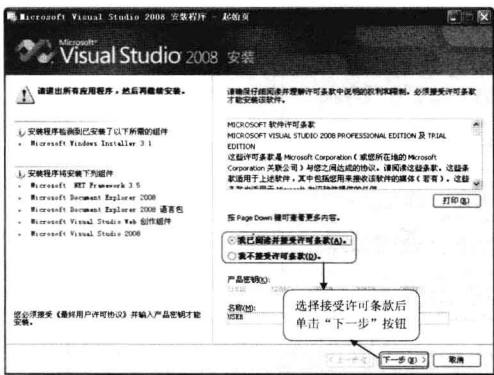


图 1.8 Visual Studio 2008 许可协议界面



图 1.9 Visual Studio 2008 安装选项界面

(5) 进入如图 1.10 所示的界面,可以看到 Visual Studio 2008 正在安装的组件和安装进度。

(6) 进入如图 1.11 所示的界面,单击“完成”按钮,到此已经完成对 Visual Studio 2008 的安装。

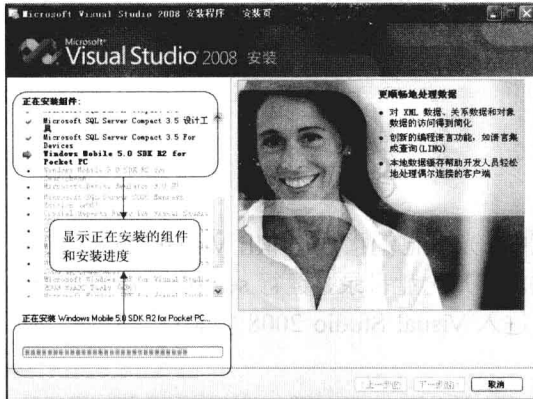


图 1.10 Visual Studio 2008 安装过程界面

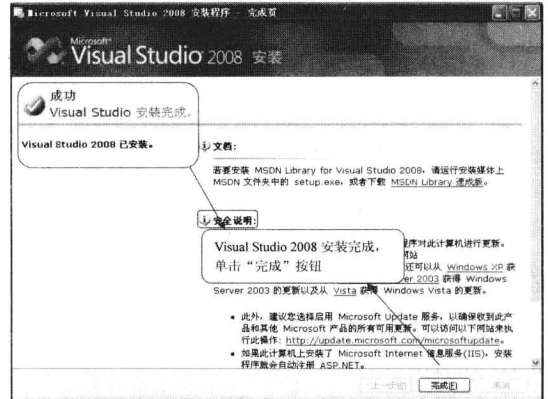


图 1.11 Visual Studio 2008 安装完成界面



小知识

在安装步骤 5 中,需要在“选择要安装的功能”选项组中点选“默认值”、“完全”或者“自定义”单选按钮。选择“默认值”将会安装常用的组件和功能;选择“完全”将会安装全部组件和功能;选择“自定义”将安装用户所选择的组件和功能。在本次安装中选择“默认值”,在通常状况下也只需要常用的组件和功能。

1.2.2 C#的开发环境

Visual Studio 2008 集成开发环境如图 1.12 所示。

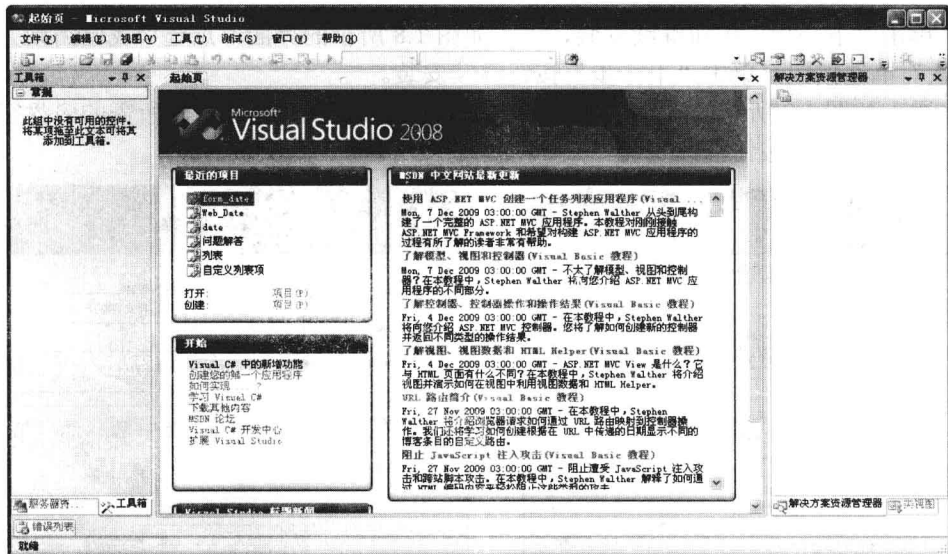


图 1.12 Visual Studio 2008 集成开发环境

“新建项目”对话框如图 1.13 所示。可视化程序设计界面如图 1.14 所示。

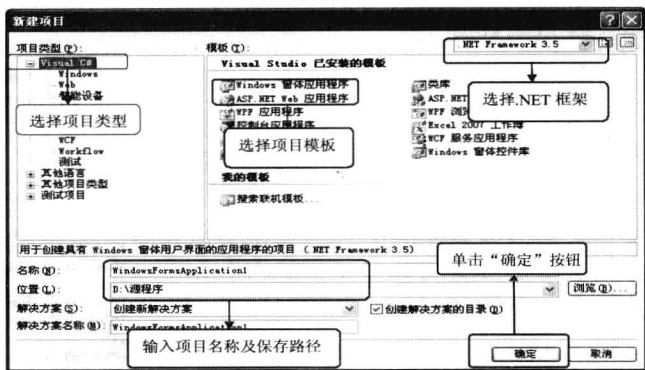


图 1.13 “新建项目”对话框

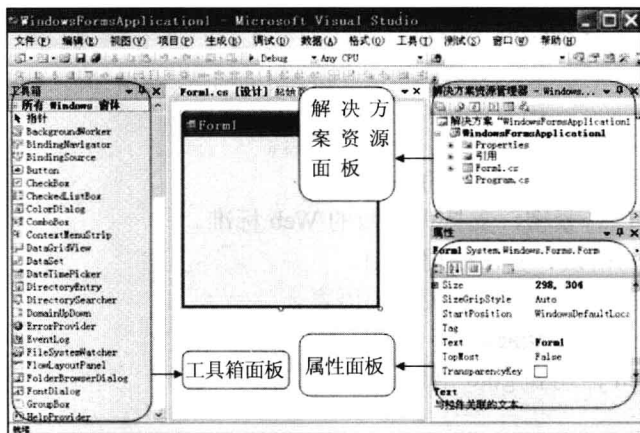


图 1.14 可视化程序设计界面

解决方案资源管理器面板提供了项目及项目文件的视图，并且提供了对项目 and 项目文件的快捷访问，如图 1.15 所示。“类视图”面板如图 1.16 所示。



图 1.15 “解决方案资源管理器”面板

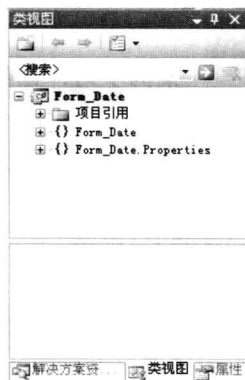


图 1.16 “类视图”面板