



高职高专计算机技能型紧缺人才培养规划教材

计算机软件技术专业

C#程序设计 实用教程

张晓蕾 主编

免费提供

教学相关资料



人民邮电出版社
POSTS & TELECOM PRESS

高职高专计算机技能型紧缺人才培养规划教材
计算机专业

C#程序设计实用教程

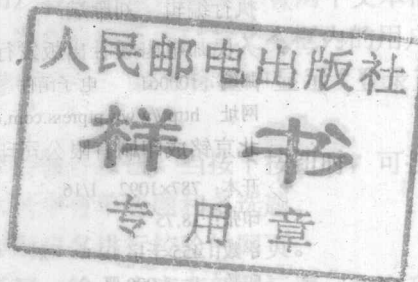
张晓蕾 主编

2. 程序框图

- (1) 参考【案例46】创建一个用户登录界面，包含两个文本框、两个标签和一个按钮。文本框用于输入用户名和密码，标签用于提示输入内容，按钮用于提交。如果用户名为“Admin”，密码为“123456”，则显示“登录成功”，否则显示“用户名或密码错误，请重新输入”。
- (2) 制作一个可以显示用户登录成功的界面。
- (3) 制作一个可以显示用户登录错误的界面。
- (4) 制作一个可以显示用户登录成功的界面。
- (5) 制作一个可以显示用户登录错误的界面。

人民邮电出版社

北京



图书在版编目 (CIP) 数据

C#程序设计实用教程 / 张晓蕾主编. —北京: 人民邮电出版社, 2008.9
高职高专计算机技能型紧缺人才培养规划教材
ISBN 978-7-115-18033-9

I. C… II. 张… III. C 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 095121 号

内 容 提 要

本书主要介绍 C#程序设计的基础知识、C#面向对象程序设计、基于 Windows Forms 的应用程序开发、ADO.NET 数据库应用开发和 ASP.NET 动态网站设计等内容。全书采用案例驱动的方式进行讲解, 通过 48 个实用案例由浅至深、层层引导, 以案例带动知识点的学习, 并注意保证知识的相对完整性和系统性, 使读者能够快速掌握 C#, 提高程序设计技能。

本书可作为高职高专院校计算机相关专业的教材, 也可作为初学者的自学用书。

高职高专计算机技能型紧缺人才培养规划教材

计算机软件技术专业

C#程序设计实用教程

◆ 主 编 张晓蕾

责任编辑 潘春燕

执行编辑 刘雁斌

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京铭成印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 18.75

字数: 455 千字

印数: 1—3 000 册

2008 年 9 月第 1 版

2008 年 9 月北京第 1 次印刷

ISBN 978-7-115-18033-9/TP

定价: 29.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

高职高专计算机技能型紧缺人才培养

规划教材编委会

主 任 武马群

副主任 王泰峰 徐民鹰 王晓丹

编 委 (以姓氏笔画为序)

马 伟 安志远 向 伟 刘 兵 吴卫祖 吴宏雷

余明辉 张晓蕾 张基宏 贺 平 柳 青 赵英杰

施晓秋 姜 锐 耿 壮 郭 勇 曹 炜 蒋方纯

潘春燕

目前,人才问题是制约我国软件产业发展的关键。为加大软件人才培养力度和提高软件人才培养质量,教育部继在2003年确定北京信息职业技术学院等35所高职院校试办示范性软件职业技术学院后,又同时根据《教育部等六部门关于实施职业院校制造业和现代服务业技能型紧缺人才培养培训工程的通知》(教职成[2003]5号)的要求,组织制定了《两年制高等职业教育计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》。示范性软件职业技术学院与计算机应用与软件技术专业领域技能型紧缺人才培养工作,均要求在较短的时间内培养出符合企业需要、具有核心技能的软件技术人才,因此,对目前高等职业教育的办学模式和人才培养方案等做较大的改进和全新的探索已经成为学校的当务之急。

据此,我们认为做一套符合上述一系列要求的切合学校实际的教学方案尤为重要。遵照教育部提出的以就业为导向,高等职业教育从专业本位向职业岗位和就业为本转变的指导思想,根据目前高等职业教育院校日益重视学生将来的就业岗位,注重培养毕业生的职业能力的现状,我们联合北京信息职业技术学院等几十所高职院校和普拉内特计算机技术(北京)有限公司、福建星网锐捷网络有限公司、北京索浪计算机有限公司等软件企业共同组建了计算机应用与软件技术专业领域技能型紧缺人才培养教学方案研究小组(以下简称研究小组)。研究小组对承担计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的79所院校的专业设置情况做了细致的调研,并调查了几十所高职院校计算机相关专业的学生就业情况以及目前软件企业的人才市场需求状况,确定首批开发目前在高职院校开设比较普遍的计算机软件技术、计算机网络技术、计算机多媒体技术和计算机应用技术4个专业方向的教学方案。

同时,为贯彻教育部提出的要与软件企业合作开展计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的精神,使高等职业教育培养出的软件技术人才符合企业的需求,研究小组与许多软件企业的专家们进行了反复研讨,了解到目前高职院校的毕业生的实际动手能力和综合应用知识方面较弱,他们和企业需求的软件人才有着较大的差距,到企业后不能很快独当一面,企业需要投入一定的成本和时间进行项目培训。针对这种情况,研究小组在教学方案中增加了“综合项目实训”模块,以求强化学生的实际动手能力和综合应用前期所学知识的能力,探索将企业的岗前培训内容前移到学校的教学中的实验之路,以此增强毕业生的就业竞争力。

在上述工作的基础上,研究小组于2004年多次组织召开了包括企业专家、教育专家、学校任课教师在内的各种研讨会和方案论证会,对各个专业按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”一步步进行了认真的分析和研讨:

- 列出各专业的岗位群及核心技能。针对教育部提出的以就业为导向,根据目前高职高专院校日益关心学生将来的就业岗位的现状,在前期大量调研的基础上,首先提炼各个专业的岗位群。如对某专业的岗位群进行研究时,首先罗列此专业的各个岗位,以便能正确了解

每个岗位的职业能力，再根据职业能力进行有意义的合并，形成各个专业的岗位群，再对每个岗位群总结和归纳出其核心技能。

- 根据岗位群及核心技能做出教学方案。在岗位群及核心技能明确的前提下，列出此岗位应该掌握的知识点，再依据这些知识点推出应该学习的课程、学时数、课程之间的联系、开课顺序并进行必要的整合，最终形成一套科学完整的教学方案。

为配合学校对技能型紧缺人才的培养工作，在研究小组开发上述4个专业的教学方案的基础上，我们组织编写了这套包含计算机软件技术、计算机网络技术、计算机多媒体技术及计算机应用技术4个专业的教材。本套教材具有以下特点：

- 注重专业整体策划的内涵。对各专业系列教材按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”的思路组织开发教材。
- 按照“理论够用为度”的原则，对各个专业的基础课进行了按需重新整合。
- 各专业教材突出了实训的比例，注重案例教学。每本教材都配备了实验、实训的内容，部分专业的教材配备了综合项目实训，使学生通过模拟具体的软件开发项目了解软件企业的运行环境，体验软件的规范化、标准化、专业化和规模化的开发流程。

为了方便教学，我们免费为选用本套教材的老师提供部分专业的整体教学方案及教学相关资料。

- 所有教材的电子教案。
- 部分教材的习题答案。
- 部分教材中实例制作过程中用到的素材。
- 部分教材中实例的制作效果以及一些源程序代码。

本套教材以各个专业的岗位群为出发点，注重专业整体策划，试图通过对系列教材的整体构架，探索一条培养技能型紧缺人才的有效途径。

经过近两年的艰苦探索和工作，本套教材终于正式出版了，我们衷心希望，各位关心高等职业教育的读者能够对本套教材的不当之处给予批评指正，提出修改意见，也热切盼望从事高等职业教育的教师以及软件企业的技术专家和我们联系，共同探讨计算机应用与软件技术专业的教学方案和教材编写等相关问题。来信请发至 panchunyan@ptpress.com.cn。

前言

C#是 Microsoft 公司推出的新型体系结构 .NET Framework 的一部分, 它全新的技术架构让应用程序和网站开发变得更为简单、功能更为强大、数据更加安全。C#是专门为 .NET 应用而开发的语言, 这从根本上保证了 C#与 .NET 框架的完美结合, 在 .NET Framework 的支持下, .NET 框架的各种优点在 C#中得以充分表现。C# 语法简洁、功能强大、类型安全, 而且是面向对象的。C#凭借它的许多创新功能, 非常容易地实现了应用程序的快速开发。

C#与 C/C++一脉相承, 作为网络时代、面向对象的程序开发语言, 它使得程序员能够在 .NET 平台上快速开发种类丰富的应用程序。与此同时, C#并没有牺牲 C/C++程序员所关心的各种特性, 它忠实地继承了 C 和 C++的很多特点, C 或 C++开发人员可以很快地适应 C#的开发。即使是一位新手, C#也能够让你快速上手。

本书采用案例驱动的方式进行讲解, 以案例设计为主导, 将知识点融入案例, 带动知识点的学习。在按案例进行讲解时, 充分注意保证知识的相对完整性和系统性。

全书共分 7 章。第 1 章为 .NET Framework 与 C#基本概念、Visual Studio 2005 开发环境应用等内容; 第 2 章为 C#语言基础, 这里将学习 C#语言的程序架构、基本语法元素和语句等; 第 3 章将介绍 C#面向对象程序设计, 学习 C#中类与对象、继承与接口等基本概念; 第 4 章将介绍基于 Windows Forms 的应用程序开发, 学习如何通过窗体、控件来快速开发应用程序; 第 5 章将介绍 C#中的图形与动画应用, 学习如何使用 GDI+进行图形图像程序的设计; 第 6 章将介绍数据库应用程序的开发, 学习如何使用 ADO.NET 2.0 及相关的数据库绑定控件来设计数据库应用程序; 第 7 章将介绍 ASP.NET 网站的开发, 学习在 Visual Studio 2005 中, 如何通过 C#语言来实现 ASP.NET 动态网页的设计。

全书通过 48 个实用案例的分析讲解, 再利用 100 余道习题的练习与巩固, 由浅至深, 层层引导, 能让读者快速掌握 C#应用程序的开发技能。各个案例都有详细的讲解, 容易看懂、便于教学。本书内容丰富、结构清晰、图文并茂, 适于教学与个人自学。

本书由张晓蕾主编。参加本书编写工作的主要人员有: 张晓蕾、杨旭、刘宇昕、耿大伟、耿文杰、张铮等。

本书可作为高职高专院校计算机相关专业的教材, 也可作为初学者的自学用书。

编者
2008 年 6 月

目 录

第1章 Visual C#程序设计基础.....1

1.1 .NET Framework 与 Visual C#.....1

1.1.1 .NET Framework 简介.....1

1.1.2 CLI、CLR 与 MSIL.....2

1.1.3 Visual C#语言简介.....3

1.1.4 Visual Studio 2005 集成

开发环境.....6

【案例1】基于控制台的应用程序.....12

【案例2】基于 Windows Forms 的 应用程序.....15

1.2 程序中简单错误的调试.....19

1.2.1 程序中的错误.....19

1.2.2 MSDN Library 的应用.....21

1.2.3 养成良好的程序设计风格.....22

【案例3】C#程序中简单错误的 调试.....25

思考与练习1.....28

第2章 C#语言基础.....30

2.1 C#的基本语法.....30

2.1.1 命名空间.....30

2.1.2 基本数据类型.....31

2.1.3 变量与常量.....33

2.1.4 运算符与表达式.....34

2.1.5 类型转换.....37

2.1.6 代码注释.....40

【案例4】简易计算器.....40

2.2 程序流程控制——条件结构.....43

2.2.1 算法与程序流程控制.....43

2.2.2 关系运算、逻辑运算与 条件运算.....46

2.2.3 if 语句.....48

2.2.4 switch 语句.....49

【案例5】计算方程的根.....51

【案例6】用户登录.....52

【案例7】计算运费.....53

2.3 程序流程控制——循环结构.....55

2.3.1 循环结构.....55

2.3.2 while 语句.....56

2.3.3 do...while 语句.....56

2.3.4 for 语句.....57

2.3.5 流程转向语句.....58

2.3.6 System.Console 类.....60

【案例8】成绩统计.....61

【案例9】乘法表.....64

【案例10】素数.....65

2.4 复杂数据类型.....67

2.4.1 数组.....67

2.4.2 枚举.....70

2.4.3 结构.....71

2.4.4 字符串.....73

【案例11】学生成绩分类统计.....74

【案例12】求矩阵对角线上元素 之和.....76

【案例13】职工工资统计.....77

【案例14】计算天数.....78

【案例15】图书查询.....80

2.5 函数.....81

2.5.1 函数的定义.....81

2.5.2 函数的调用.....82

2.5.3 函数的返回.....84

2.5.4 函数的参数.....85

2.5.5 函数重载.....87

2.5.6 Main 函数.....88

【案例16】验证哥德巴赫猜想.....89

【案例17】兔子问题.....90

【案例18】函数重载.....92

【案例 19】命令行计算器	93	4.2.3 列表框 ListBox	141
思考与练习 2	94	4.2.4 组合框 ComboBox	143
第 3 章 面向对象程序设计	97	4.2.5 面板 Panel 和群组框 GroupBox	143
3.1 面向对象程序设计简介	97	【案例 27】电子试卷	144
3.1.1 面向对象的基本概念	97	【案例 28】课程选择	146
3.1.2 类的定义	98	【案例 29】字体设置	149
3.1.3 类的成员	99	4.3 图像控件	151
3.1.4 构造函数	103	4.3.1 图片框 PictureBox	151
3.1.5 析构函数	104	4.3.2 图像列表组件 ImageList	153
3.1.6 域、属性和方法	105	【案例 30】图像浏览器 (一)	153
【案例 20】温度转换	108	【案例 31】散步的猫	156
【案例 21】圆形计算	109	4.4 消息对话框和通用对话框	158
3.2 继承、多态与抽象类	110	4.4.1 Windows 中的对话框	158
3.2.1 继承	110	4.4.2 消息对话框 MessageBox	159
3.2.2 多态	112	4.4.3 通用对话框组件	161
3.2.3 抽象类	114	4.4.4 其他对话框控件	164
【案例 22】圆与圆柱	116	【案例 32】图像浏览器 (二)	164
3.3 接口	117	4.5 菜单、工具栏与状态栏	168
3.3.1 接口的定义	117	4.5.1 菜单、工具栏与状态栏 简介	168
3.3.2 接口的成员	118	4.5.2 菜单设计	168
3.3.3 接口的实现	119	4.5.3 工具栏设计	171
【案例 23】绘图类接口	120	4.5.4 状态栏设计	173
思考与练习 3	121	【案例 33】弹出菜单	174
第 4 章 基于 Windows Forms 的 程序设计	123	【案例 34】图像浏览器 (三)	176
4.1 Windows Forms 程序设计基础	123	4.6 SDI 与 MDI 应用程序	183
4.1.1 Windows Forms 概述	123	4.6.1 SDI 与 MDI 应用程序 简介	183
4.1.2 控件基础	124	4.6.2 MDI 多文档应用程序	184
4.1.3 属性	127	【案例 35】图像浏览器 (四)	188
4.1.4 事件	128	思考与练习 4	196
4.1.5 窗体与常用控件	131	第 5 章 图形与动画	198
【案例 24】有个性的按钮	134	5.1 GDI+入门基础	198
【案例 25】动态显示的窗口	136	5.1.1 图形绘制基础	198
【案例 26】文本格式编辑	138	5.1.2 图形坐标系统	199
4.2 选择控件	140	5.1.3 辅助绘图对象	199
4.2.1 单选按钮 RadioButton	140	5.1.4 图形对象 Graphics	202
4.2.2 复选框 CheckBox	141		

5.1.5 画笔 (Pen) 与画刷 (Brush)	205
【案例 36】绘制基本图形	210
【案例 37】函数图案	213
【案例 38】渐变图形	215
5.2 图像处理	218
5.2.1 Image 类与 Bitmap 类	218
5.2.2 图形绘制与变换	220
【案例 39】图像变换	221
5.3 图形文字	223
5.3.1 Font 类	223
5.3.2 文本的绘制	224
【案例 40】特效文字	224
思考与练习 5	227
第 6 章 数据库应用开发	228
6.1 数据库开发基础	228
6.1.1 数据库基础知识	228
6.1.2 MS SQL Server 与 Access 数据库	229
【案例 41】在 Visual Studio 2005 集 成开发环境中创建数据库	230
【案例 42】在 Microsoft Access 中 创建数据库	232
6.2 通过数据组件访问数据库	235
6.2.1 在 Visual Studio 2005 中 创建数据连接	235
6.2.2 BindingSource 组件	236
6.2.3 数据绑定控件	239
【案例 43】商品信息浏览	240
【案例 44】商品信息查询	243
6.3 ADO.NET 数据应用	245
6.3.1 ADO.NET 简介	245
6.3.2 数据连接对象 Connection	247
6.3.3 数据适配器 DataAdapter	249
6.3.4 数据集 DataSet	250
6.3.5 Command 与 DataReader	251
6.3.6 其他常用 ADO.NET 对象	252
6.3.7 SQL 查询	253
【案例 45】商品信息编辑	256
思考与练习 6	259
第 7 章 ASP.NET Web 应用开发	260
7.1 ASP.NET 基础	260
7.1.1 ASP.NET 概述	260
7.1.2 HTML 语言概述	261
7.1.3 Web Form 基本概念	266
7.1.4 ASP.NET 的事件处理流程	269
7.1.5 页面指令与页面事件	270
7.1.6 ASP.NET 服务器环境	272
【案例 46】用户登录页面	273
7.2 Web 服务器控件	276
7.2.1 Web 服务器控件概述	276
7.2.2 HTML 服务器控件	277
7.2.3 Web 服务器控件	279
7.2.4 数据验证控件	280
【案例 47】用户注册页面	283
【案例 48】用户输入信息有效性 验证	286
思考与练习 7	289

第 1 章

Visual C#程序设计基础

1.1 .NET Framework 与 Visual C#

1.1.1 .NET Framework 简介

.NET Framework (又称.NET 框架) 是 Microsoft (微软) 公司所创建的新一代应用开发平台, 它能够让软件不仅适用于传统的 PC, 而且也能够满足目前网络时代提出的新任务的需要。

与微软传统的开发软件不同的是, .NET Framework 不仅可用于 Windows 平台, 也逐步推出可用于其他平台的版本。例如 Mono, 它是 .NET Framework 的开源代码版本, 可以用于多种操作系统下的应用开发, 包括 Linux 和 Mac OS 等。此外, .NET Framework 还广泛用于嵌入式开发, 如移动电话、个人数字助理 (PDA) 等。因此, 在一定程度上, 可以将微软的 .NET Framework 战略看作是对 Java 所称的“一次编写, 随处运行”的反击。微软的目标是让 .NET 战略彻底改变软件的开发方式、发行方式和使用方式等, 并且, 不只是针对微软自身的开发, 而是面向所有开发商与运营商的开发。

.NET Framework 是一种面向网络、支持各种用户终端的开发平台环境, .NET 的核心内容之一就是要搭建第 3 代因特网平台, 这个网络平台将解决网站之间的协同合作问题, 从而最大限度地获取信息。在 .NET 平台上, 不同网站之间通过相关的协定联系在一起, 网站之间形成自动交流, 协同工作, 提供最全面的服务。

.NET Framework 是生成、部署和运行 XML Web Services 和应用程序的多语言环境, 是运行所有 .NET 应用程序的基础, 它包括以下 3 个主要组成部分。

1. 公共语言运行时

公共语言运行时 (Common Language Runtime, CLR) 也称通用语言运行库, 是 .NET 框架的基础。可以将运行库看作一个在执行程序时管理其代码的代理, 它提供核心服务 (如内存管理、线程管理和远程处理), 而且还强制实施严格的类型安全, 来确保程序的安全性和可靠性。事实上, 代码托管的概念是公共语言运行库的基本原则。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。

除了运行时服务, .NET Framework 还包含一个由 4000 多个类组成的内容详尽的库, 这些类被组织为“命名空间”, 为从文件输入和输出到字符串操作、到 XML 分析、到 Windows 窗体控件的所有内容提供多种有用的功能。

2. 统一编程类

在早期的开发中, 各种应用开发人员使用各自平台所提供的工具类库, 如, C++ 开发人员使用的是 Microsoft 基类库 (MFC), Java 开发人员使用的是 Windows® 基类库 (WFC),

而 Visual Basic 用户使用的又是 VBRUN。开发适用于不同平台的应用时,则要求开发人员掌握多种类库的使用方法,造成大量的资源浪费,也降低了开发人员的效率。

.NET 框架的关键作用在于,它提供了一个跨编程语言的统一编程环境,它为开发人员提供了一个统一、面向对象、层次化、可扩展的类库集(API),即统一编程类。

统一编程类是一个综合性的面向对象且可重复使用的类型集合,只要掌握了统一编程类,开发人员就可以在.NET 框架下使用各种不同的语言来进行应用开发,可以使用它开发包含传统的命令行或图形用户界面(GUI)应用程序、基于 ADO.NET 的数据库应用,以及 ASP.NET 所提供的最新的应用(如 Web 窗体和 XML Web 服务)在内的应用程序。

3. ASP.NET

ASP.NET 建立在.NET 框架的编程类之上,它提供了一个 Web 应用程序模型,并且包含使生成 Web 应用程序变得简单的控件集和结构。ASP.NET 包含封装公共 HTML 用户界面控件(如文本框和下拉菜单)的控件集。这些控件在 Web 服务器上运行,并以 HTML 的形式将它们的用户界面传送到浏览器。在服务器上,这些控件公开一个面向对象的编程模型,可以让 Web 开发人员编写丰富的面向对象程序。ASP.NET 还提供结构服务(如会话状态管理和进程回收),进一步减少了开发人员必须编写的代码量并提高了应用程序的可靠性。

总之,微软的.NET 战略是一场软件开发的革命,对开发人员而言,.NET 能自动在软件编写过程中进行“智能拼接”,使得开发人员可以集中精力编写业务逻辑,而不必编写 COM 基本结构,不但会改变开发人员开发应用程序的方式,而且使得开发人员能创建出全新的各种应用程序,并大幅提高了软件生产率;对最终用户来说,计算机的功能将会得到大幅度提升,提供了通过任何设备随时随地操作数据和进行通信的能力,同时计算机操作也会变得非常简单;对公司而言,.NET 把雇员客户和商务应用程序整合成一个协调的、能进行智能交互的整体,而公司无疑将是这场效率和生产力革命的最大受益者。

1.1.2 CLI、CLR 与 MSIL

通用语言基础构造(Common Language Infrastructure, CLI)是微软开发的一个允许编程者用多种编程语言创建 Web 服务和其他应用的技术,包括运行程序所需的虚拟执行系统和定义不同数据类型如何相互作用的基类库。

CLI 指的是一种支持动态组件编程模型的多重结构。在许多情况下,这代表了一个与原来的 C++对象模型完全颠倒了的模式。它通过一个实时的软件层,在底层操作系统与程序之间有效地运行。CLR 即是 CLI 在微软.NET 战略中的实现。

公共语言运行时(Common Language Runtime, CLR)是.NET 的核心,CLR 的功能通过编译器和工具公开,可以编写利用此托管执行环境的代码。使用基于公共语言运行库的语言编译器开发的代码称为托管代码,它与编译为面向特定系统的本机机器语言的“非托管代码”相对应。托管代码具有许多优点,例如:跨语言集成、跨语言异常处理、增强的安全性、版本控制和部署支持、简化的组件交互模型、调试和分析服务等。通过托管代码,可以很容易地实现程序代码的与平台无关性,并提高程序运行性能。

有了公共语言运行库,就可以很容易地设计出对象能够跨语言交互的组件和应用程序。也就是说,用不同语言编写的对象可以互相通信,并且它们的行为可以紧密集成。

在 CLR 执行开发的源代码之前,需要进行编译。在.NET 中,编译分为两个阶段:

① 把源代码编译为 Microsoft 中间语言 (IL);

② CLR 把 IL 编译为平台专用的代码。

这两个阶段的编译过程非常重要,因为 Microsoft 中间语言(托管代码)是提供 .NET 的许多优点的关键。

在 .NET 框架中,CLI 允许不同的语言使用 .NET 框架。因此在 .NET 框架中,所有的语言(C#、Visual C++.NET、Visual Basic.NET 等)最后都被转换为了一种通用语言:微软中间语言 (MSIL)。

MSIL 是将 .NET 代码转化为机器语言的一个中间过程。它是一种介于高级语言和基于 Intel 的汇编语言的伪汇编语言。当用户编译一个 .NET 程序时,编译器将源代码翻译成一组可以有效地转换为本机代码且独立于 CPU 的指令。当执行这些指令时,实时 (Just-In-Time, JIT) 编译器将它们转化为 CPU 特定的代码。由于 CLR 支持多种实时编译器,因此同一段 MSIL 代码可以被不同的编译器实时编译并运行在不同的结构上。

从理论上来说,MSIL 将消除多年以来业界中不同语言之间的纷争。在 .NET 的应用中可能出现下面的情况:一部分代码可以用 C# 实现,另一部分代码使用 Visual Basic.NET 完成,但是最后这些代码都将被转换为中间语言。这给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了。

1.1.3 Visual C#语言简介

1. Visual C#概述

.NET Framework 支持多种语言进行的应用开发,微软自身就提供了 Visual C# (简称 C#, 读作 C Sharp)、Visual C++.NET、Visual Basic.NET 以及 Visual J#.NET 等多种开发语言,其中,最重要的就是 Visual C#。

长期以来,C 和 C++ 在开发应用中一直是最有生命力的程序设计语言,它们为程序员提供了丰富的功能、高度的灵活性和强大的底层控制能力,但是这一切都不得不在开发效率上作出牺牲。在网络时代,当 Java 出现后,以其良好的网络开发能力及快速开发的效率,在基于网络的开发方面不断地攻城略地,一步步地蚕食 C 和 C++ 的份额,一举成为网络开发应用最多的开发语言。但是,Java 在提供强大网络应用开发能力的同时,也失去了 C/C++ 在灵活性和底层控制能力方面的优势。

C# 与 C/C++ 一脉相承,作为网络时代的、面向对象的程序开发语言,它使得程序员能够在 .NET 平台上快速开发种类丰富的应用程序,.NET 平台提供的大量的工具和服务能够最大限度地发掘和使用计算及通信能力。Visual C# 在带来对应用程序的快速开发能力的同时,并没有牺牲 C/C++ 程序员所关心的各种特性,它忠实地继承了 C 和 C++ 的优点,C 或 C++ 开发人员可以很快地就适应 C# 的开发。即使是一位新手,Visual C# 也能够让他快速地上手,快速应用程序开发 (Rapid Application Development, RAD) 的思想与简洁的语法将会使新手迅速成为一名熟练的开发人员。

C# 是专门为 .NET 应用而开发出的语言,这从根本上保证了 C# 与 .NET 框架的完美结合。在 .NET Framework 的支持下,.NET 框架的各种优点在 Visual C# 中得以充分表现。C# 语法简洁、功能强大、类型安全,而且是面向对象的。C# 凭借它的许多创新,在保持 C 语言风格的表示形式和优美的同时,实现了应用程序的快速开发。

C#语法表现力很强,只有不到 90 个关键字,而且简单易学。C#语法简化了 C++ 的诸多复杂性,同时提供了很多强大的功能。C#还支持泛型方法和类型,从而提供了更出色的类型安全和性能。C#还提供了迭代器,允许集合类的实现者定义自定义的迭代行为,简化了客户端代码对它的使用。

作为一种面向对象的语言,C#支持封装、继承和多态性概念。所有的变量和方法,包括 Main 方法(应用程序的入口点),都封装在类定义中。类可能直接从一个父类继承,但它可以实现任意数量的接口。

除了这些基本的面向对象的原理,C#还通过几种创新的语言结构加快了软件组件的开发,其中包括:封装的方法签名(称为委托),它实现了类型安全的事件通知;属性(Property),充当私有成员变量的访问器;属性(Attribute),提供关于运行时类型的声明性元数据;内联 XML 文档注释。

语言互操作性是 .NET Framework 的一个关键功能。因为由 C#编译器生成的 IL 代码符合公共类型规范(CTS),因此从 C#生成的 IL 代码可以与从 Visual Basic、Visual C++、Visual J# 的 .NET 版本或者其他 20 多种符合 CTS 的语言中任何一种语言生成的代码进行交互。单一程序集可能包含用不同 .NET 语言编写的多个模块,并且类型可以相互引用,就像它们是用同一种语言编写的。

.NET Framework 与 Web 的紧密结合,使开发者能够利用已有的知识与技巧快速地开发 Web 服务,C#能够方便地提供面向 Web 的应用开发,能够快速开发基于 ASP.NET 的 Web 应用。

简而言之,C#具有如下特点:简洁的语法,强大的面向对象开发能力,与 Web 的紧密结合,完整的安全性及错误处理,较强的灵活性与兼容性等。

2. Visual C#程序的编译与运行原理

用 C#编写的源代码在编译时,会被编译为一种符合 CLI 规范的中间语言。IL 代码与资源(如位图和字符串)一起作为一种被称为程序集的可执行文件存储在磁盘上,通常具有的扩展名为 .exe 或 .dll。程序集包含清单,它提供关于程序集的类型、版本、区域性和安全等信息。

执行 C#程序时,程序集将加载到 CLR 中,这可能会根据清单中的信息执行不同的操作。然后,如果符合安全要求,CLR 执行 JIT 编译以将 IL 代码转换为本机机器指令。CLR 还提供与自动垃圾回收、异常处理和资源管理有关的其他服务。图 1-1 表现了 C#源代码文件、基类库、程序集和 CLR 在编译时与运行时的关系。

3. Visual C# 2005 的新特性

Visual C# 2005 是 C#的新版本,与原来的 Visual C# 2003 相比,Visual C# 2005 增加了许多新功能,大大简化了开发步骤,提高了开发效率,让开发人员进行应用开发更加得心应手。

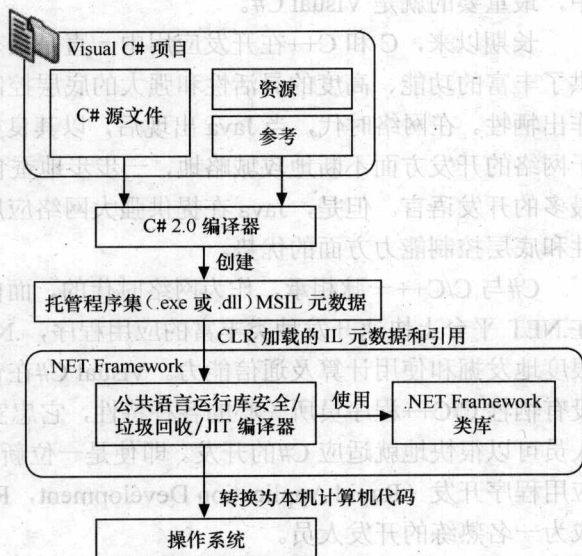


图 1-1 C#程序的编译与运行

下面是其中部分新特性。

(1) 分部类型

分部类型定义允许将单个类型（比如某个类）拆分为多个文件。使 C# 开发人员能够使用不完全类型，将包含大量源代码的类型分割到多个不同源文件中。不完全类型提供更方便的开发和代码维护，使开发人员能够分隔开设计器自动生成的和用户编写的类型代码部分，以便更有效地补充或修改由工具生成的代码。

(2) 可空类型

可空类型允许变量包含未定义的值。在使用数据库和其他可能包含未含有具体值的元素的数据结构时，可以使用可空类型。

(3) 迭代器

迭代器使得通过 `foreach` 循环来轮流访问集合的内容变得更加容易。

(4) 泛型

Visual C# 2005 中添加了一些泛型类型，使得程序员能够实现程度很高的代码重用，获得更高的集合类性能。

(5) 匿名方法

可以将代码块作为参数来传递。在本应使用委托的任何地方，都可以使用代码块来取代，不需要定义新的方法。

(6) 命名空间别名限定符

命名空间别名限定符 (`::`) 对访问命名空间成员提供了更多控制。`global ::` 别名允许访问可能被代码中的实体隐藏的根命名空间。

(7) 优化的代码段

通过提供可填写的模板，代码段加速了通用代码构造的输入。代码段存储为 XML 文件，可轻松对其进行编辑和自定义。

(8) 强大的重构工具

重构工具可自动重新构造源代码，例如，通过将局部变量升级为参数或将代码块转换为方法；使用提取方法 (Extract Method, EM)，可以通过从现有成员的代码块中提取选定的代码来创建新方法。

(9) 良好的编码风格

Visual C# 2005 对代码的编写风格非常讲究，能随时对正在编辑的代码块自动进行分段并缩进。这极好地纠正了部分编码人员不合适的风格，可产生一种潜移默化的影响。

(10) 友好的跟踪提示

在编写代码时，集成开发环境 (IDE) 会自动跟踪提示相关的类、结构体、枚举、变量等信息。这极大地提高了开发人员的输入速度，并保证了编码的正确性。也许开发人员只需敲几下空格和回车，按几下方向键就可以轻松完成一段代码。

(11) 新颖的类设计器

类设计器是一种新的、以图形方式显示类和类型的编辑器，并允许添加或修改方法。也可以从“类设计器”窗口中使用重构工具。

(12) 调试更加方便

在调试程序的时候，经常会遇到这样的情况：当调试一大段代码时，遇到了一个小小的

错误, 比如参数赋值有错误了, 这时候, 往往希望能够马上将这个错误改正过来, 并能够继续跟踪调试下去, 而不用结束整个调试过程去修改。

在 Visual Studio 2003 中, 遇到此类情况时, 就必须停止当前的调试, 修改完出错的地方, 再重新编译, 这样十分不方便。在 Visual Studio 2005 中, 提供了一个称为“edit and continue”的新功能, 当在调试时, 遇到了小的错误需要马上修改, 可以直接进行编辑修改; 然后继续往下调试, 不需要结束整个调试过程; 当修改完毕后, 调试器会在后台自动地进行编译, 并且执行的是修改后的新代码, 十分方便。

1.1.4 Visual Studio 2005 集成开发环境

Visual Studio .NET 集成开发环境是微软配合 .NET 战略而提供的一套完整的开发工具集, 用于生成桌面应用程序、ASP.NET Web 应用程序、XML Web Services 和移动应用程序。Visual C#、Visual Basic.NET、Visual C++ 和 Visual J# 全都使用相同的集成开发环境, 利用此集成开发环境可以共享工具, 且有助于创建混合语言解决方案。

1. Visual Studio 2005 的启动和退出

(1) 启动 Visual Studio 2005 集成开发环境

在 Windows 桌面, 单击“开始”→“所有程序”→“Microsoft Visual Studio 2005”→“Microsoft Visual Studio 2005”菜单命令, 启动 Visual Studio 2005 集成开发环境。默认情况下, 启动后显示的是 Microsoft Visual Studio 2005 集成开发环境“起始页”窗口, 如图 1-2 所示。

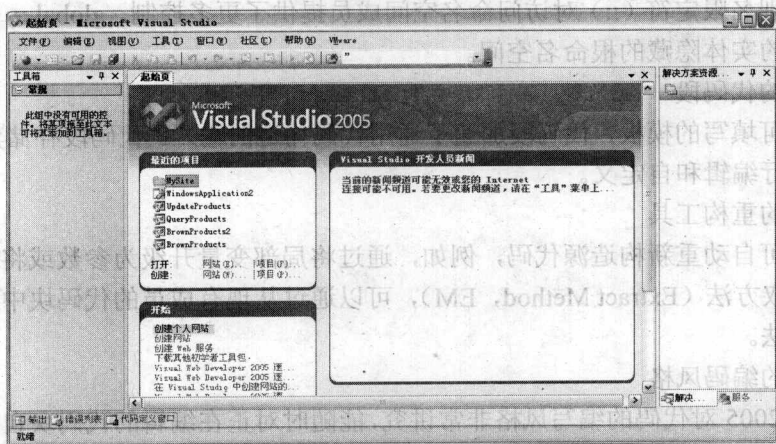


图 1-2 Microsoft Visual Studio 2005 集成开发环境“起始页”窗口

在 Microsoft Visual Studio 2005 集成开发环境“起始页”窗口中, 在“最近的项目”列表框中可以打开已创建的项目或网站, 也可以创建新的项目或网站; 能够通过“Visual Studio 开发人员指南”了解最新的 Visual Studio 开发动态和开发技术相关的知识性文章; 还可以从“开始”列表框中学习 Visual Studio 2005 项目(或网站)开发的基础知识、下载工具包等。

(2) 退出 Visual Studio 2005 集成开发环境

单击图 1-2 所示窗口中右上角的“关闭”按钮 , 或者单击“文件”→“退出”菜单命令, 都可以退出 Visual Studio 2005 集成开发环境。如果在退出时, 用户没有保存当前正在编辑的项目程序, 将会显示一个对话框, 提醒用户保存所编辑的程序。

2. Visual Studio 2005 集成开发环境界面

启动的 Visual Studio 2005 集成开发环境,是所有 Visual Studio 2005 开发工具(如 Visual C#、Visual Basic)所共有的公共环境,要进入 Visual C# 2005 特有的开发环境,首先需要创建(或打开)一个 Visual C# 2005 的应用程序。

在 Microsoft Visual Studio 2005 集成开发环境“起始页”窗口中,单击“文件”→“新建项目”菜单命令,调出“新建项目”对话框,如图 1-3 所示。

在“项目类型”列表中选中“Visual C#”选项,再在“模板”列表中,选中“Windows 应用程序”选项,在“名称”文本框中键入程序的名称,在“位置”文本框中,输入项目保存的位置(也可单击右边的“浏览”按钮以调出“项目位置”对话框来设置项目保存的位置)。最后单击“确定”按钮,创建 Visual C#应用程序并进入 Visual C#的集成开发环境,如图 1-4 所示。

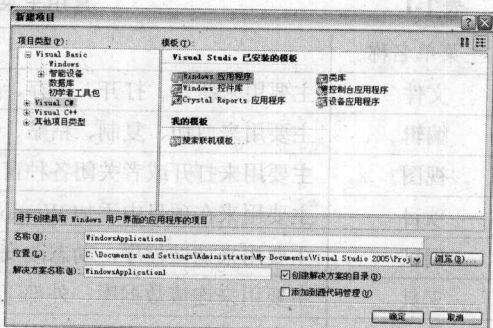


图 1-3 “新建项目”对话框

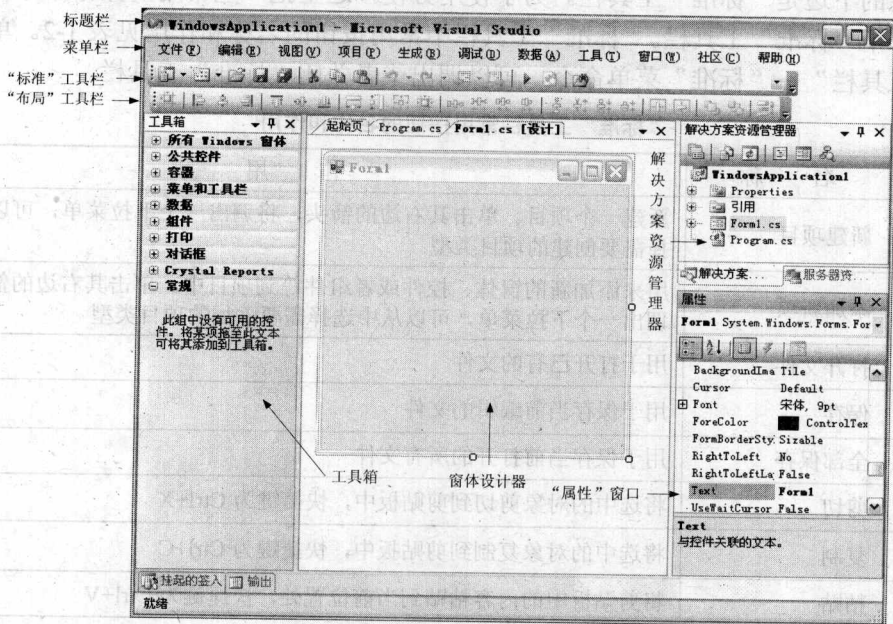


图 1-4 Visual Studio 2005 (C#) 集成开发环境

下面将依次介绍 Visual Studio 2005 (C#) 集成开发环境中的各个主要部分的名称和功能。

(1) 标题栏

标题栏位于窗口的最上方,其作用和风格与其他的 Windows 窗口基本一样。标题左边的文字是程序的名称和“Microsoft Visual Studio”文字,以及当前程序所处的状态,例如:“正在运行”、“正在调试”等。

(2) 菜单栏

菜单栏在标题栏的下边, Visual C# 2005 集成开发环境菜单的形式与其他 Windows 软件