



中国高等职业技术教育研究会推荐
高职高专计算机专业规划教材

Visual C#.NET 程序设计基础

■ 曾文权 周文琼 陶 南 卢淑萍 编 著
刘喜勋 主 审



西安电子科技大学出版社
<http://www.xduph.com>

□中国高等职业技术教育研究会推荐

高职高专计算机专业规划教材

Visual C#.NET程序设计基础

曾文权 周文琼 陶 南 卢淑萍 编著

刘喜勋 主审

西安电子科技大学出版社

2008

内 容 简 介

C# 是基于微软下一代平台 .NET 的面向对象程序设计语言, Visual C#.NET 是 C# 结合 Visual Studio.NET 集成开发平台进行软件开发的主要工具。本书采用了最新的 Visual Studio.NET 2005 作为开发平台, 坚持对理论知识的讲解把握度, 重在以应用结合紧密的实例、案例、项目为核心, 将知识点融汇在其中。

本书主要内容包括: .NET 的基本概念和知识、Visual Studio.NET 集成开发环境; 窗体及常用控件的基本属性; C# 语言的基本概念和高级特性; C# 完全面向对象的特性; Visual C#.NET 可视化编程; WinForms 控件及对话框; 程序调试、部署与打包; ADO.NET 数据库编程概述。本书每章末还配有一定的练习与作业, 方便学生巩固所学知识。本书更突出的是编写了面向学生、易于操作的上机内容。

本书主要面向高职高专学生, 也可作为初学编程人员的自学教程, 同时可作为相关技术人员的培训教材。

★ 本书配有电子教案, 需要者可登录出版社网站, 免费下载。

图书在版编目(CIP)数据

Visual C#.NET 程序设计基础 / 曾文权等编著.

—西安: 西安电子科技大学出版社, 2008.2

中国高等职业技术教育研究会推荐. 高职高专计算机专业规划教材

ISBN 978-7-5606-1975-0

I. V… II. 曾… 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 008138 号

策 划 云立实

责任编辑 王跃华 云立实

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com

E-mail: xdupfb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2008 年 2 月第 1 版 2008 年 2 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 28

字 数 661 千字

印 数 1~4000 册

定 价 39.00 元

ISBN 978-7-5606-1975-0/TP·1022

XDUP 2267001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜, 谨防盗版。

进入 21 世纪以来,高等职业教育呈现出快速发展的形势。高等职业教育的发展,丰富了高等教育的体系结构,突出了高等职业教育的类型特色,顺应了人民群众接受高等教育的强烈需求,为现代化建设培养了大量高素质技能型专门人才,对高等教育大众化作出了重要贡献。目前,高等职业教育在我国社会主义现代化建设事业中发挥着越来越重要的作用。

教育部 2006 年下发了《关于全面提高高等职业教育教学质量的若干意见》,其中提出了深化教育教学改革,重视内涵建设,促进“工学结合”人才培养模式改革,推进整体办学水平提升,形成结构合理、功能完善、质量优良、特色鲜明的高等职业教育体系的任务要求。

根据新的发展要求,高等职业院校积极与行业企业合作开发课程,根据技术领域和职业岗位群任职要求,参照相关职业资格标准,改革课程体系和教学内容,建立突出职业能力培养的课程标准,规范课程教学的基本要求,提高课程教学质量,不断更新教学内容,而实施具有工学结合特色的教材建设是推进高等职业教育改革发展的重要任务。

为配合教育部实施质量工程,解决当前高职高专精品教材不足的问题,西安电子科技大学出版社与中国高等职业技术教育研究会在前三轮联合策划、组织编写“计算机、通信电子、机电及汽车类专业”系列高职高专教材共 160 余种的基础上,又联合策划、组织编写了新一轮“计算机、通信、电子类”专业系列高职高专教材共 120 余种。这些教材的选题是在全国范围内近 30 所高职高专院校中,对教学计划和课程设置进行充分调研的基础上策划产生的。教材的编写采取在教育部精品专业或示范性专业的高职高专院校中公开招标的形式,以吸收尽可能多的优秀作者参与投标和编写。在此基础上,召开系列教材专家编委会,评审教材编写大纲,并对中标大纲提出修改、完善意见,确定主编、主审人选。该系列教材以满足职业岗位需求为目标,以培养学生的应用技能为着力点,在教材的编写中结合任务驱动、项目导向的教学方式,力求在新颖性、实用性、可读性三个方面有所突破,体现高职高专教材的特点。已出版的第一轮教材共 36 种,2001 年全部出齐,从使用情况看,比较适合高等职业院校的需要,普遍受到各学校的欢迎,一再重印,其中《互联网实用技术与网页制作》在短短两年多的时间里先后重印 6 次,并获教育部 2002 年普通高校优秀教材奖。第二轮教材共 60 余种,在 2004 年已全部出齐,有的教材出版一年多的时间里就重印 4 次,反映了市场对优秀专业教材的需求。前两轮教材中有十几种入选国家“十一五”规划教材。第三轮教材 2007 年 8 月之前全部出齐。本轮教材预计 2008 年全部出齐,相信也会成为系列精品教材。

教材建设是高职高专院校教学基本建设的一项重要工作。多年来,高职高专院校十分重视教材建设,组织教师参加教材编写,为高职高专教材从无到有,从有到优、到特而辛勤工作。但高职高专教材的建设起步时间不长,还需要与行业企业合作,通过共同努力,出版一大批符合培养高素质技能型专门人才要求的特色教材。

我们殷切希望广大从事高职高专教育的教师,面向市场,服务需求,为形成具有中国特色和高职教育特点的高职高专教材体系作出积极的贡献。

中国高等职业技术教育研究会会长

2007 年 6 月

高职高专计算机专业规划教材

编审专家委员会

主任: 温希东 (深圳职业技术学院副校长, 教授)

副主任: 徐人凤 (深圳职业技术学院电子与通信工程学院副院长, 高工)

刘中原 (上海第二工业大学计算机与信息学院副院长, 副教授)

李卓玲 (沈阳工程学院信息工程系主任, 教授)

委员: (按姓氏笔画排列)

丁桂芝 (天津职业大学电子信息工程学院院长, 教授)

马宏锋 (兰州工业高等专科学校计算机工程系副主任, 副教授)

王 军 (武汉交通职业学院信息系副主任, 副教授)

王 雷 (浙江机电职业技术学院计算机应用工程系主任, 高工)

王养森 (南京信息职业技术学院计算机科学与技术系主任, 高工)

王趾成 (石家庄职业技术学院计算机系主任, 高工)

汤 勇 (成都职业技术学院国际软件学院副院长, 副教授)

朱小平 (广东科学技术职业学院计算机学院副院长, 副教授)

齐志儒 (东北大学东软信息学院计算机系主任, 教授)

孙街亭 (安徽职业技术学院教务处处长, 副教授)

张 军 (石家庄职业技术学院计算机系, 高工)

李成大 (成都电子机械高等专科学校计算机工程系副主任, 副教授)

苏传芳 (安徽电子信息职业技术学院计算机科学系主任, 副教授)

苏国辉 (黎明职业大学计算机系副主任, 讲师)

汪临伟 (九江职业技术学院电气工程系主任, 副教授)

汪清明 (广东轻工职业技术学院计算机系副主任, 副教授)

杨文元 (漳州职业技术学院计算机工程系副主任, 副教授)

杨志茹 (株洲职业技术学院信息工程系副主任, 副教授)

胡昌杰 (湖北职业技术学院计算机科学与技术系副主任, 副教授)

聂 明 (南京信息职业技术学院软件学院院长, 副教授)

章忠宪 (漳州职业技术学院计算机工程系主任, 副教授)

睦碧霞 (常州信息职业技术学院软件学院院长, 副教授)

董 武 (安徽职业技术学院电气工程系副主任, 副教授)

蒋方纯 (深圳信息职业技术学院软件工程系主任, 副教授)

鲍有文 (北京联合大学信息学院副院长, 教授)

前言

C#是.NET平台上一种简单、现代、功能强大、安全而灵活的程序设计语言,它能够让开发人员在.NET平台上快速建立大量的应用程序。2000年7月在美国奥兰多举行的专业开发者大会上,Microsoft公司展示了最新的.NET体系,介绍了.NET的特征和构成,从此Microsoft公司将它的未来与.NET Framework 紧密联系在了一起。Visual C#.NET 作为基于此框架的主要开发工具,已经成为当今软件开发应用最广泛的开发工具之一。

本书从.NET 最新技术的基本概念介绍出发,结合高职高专学生的具体特点,由简单到复杂,由具体到抽象,逐步将学习者引入 Visual C#.NET 高级开发的“殿堂”。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,力求以实例、案例、项目为核心来融汇实用的知识点。

它山之石,可以攻玉。我们借鉴国内外先进的软件人才培养体系、培养理念和多年的高职教学经验来组织教学内容。坚持理论知识以“够用、适用”为度,不强求理论体系的完备,不强求理论学习的按部就班,不过分强调理论的指导作用。坚持通过大量的上机练习、代码阅读、代码改错、代码规范化等手段来提高学生的技能——动手能力和实际项目的开发能力。强调有效利用有限的学习时间,将“最需要”、“最能够在项目中用到”的知识传授给学生;强调将“最新软件开发平台和技术”介绍给学生,并使之掌握。

本书共分11章,各章内容如下:

第1章介绍了.NET平台的构成和 Visual Studio.NET 2005 集成开发环境。

第2章介绍了C#语言的基本概念和基本知识。

第3章讲解了C#语言面向对象的一些基本特性,包括类和对象、接口、继承与多态。

第4章介绍了C#语言中的一些高级概念,包括集合、异常、属性和索引器、事件与委托,为 Visual C#.NET 可视化程序设计打下基础。

第5、6章介绍了窗体编程的基本知识和一些常用控件的使用,主要包括: Visual C#.NET 中窗体容器和可视化程序设计的一些常用控件。

第7章讲解了开发窗体编程时常用的一些对话框。主要介绍了系统通用对话框和如何开发自定义对话框。

第8章介绍了窗体编程的一些高级界面设计控件。SDI和MDI界面设计、菜单设计、窗体工具栏和状态栏设计以及一些界面的增强功能。

第9章讲解了 Visual C#.NET 程序的调试、测试、打包和部署等方法。

第10章重点讲解了在 Visual C#.NET 中使用 ADO.NET 进行数据库访问的技术。

第11章分析了学生信息管理系统项目中所涉及到的 Visual C#.NET 开发技术，也是对前面10章所讲知识的应用与总结。

全书由曾文权主编。本书第1、10、11章由曾文权同志编写，第2、3章由卢淑萍同志编写，第4、5、6章由陶南同志编写，第7、8、9章由周文琼同志编写。

由于时间和编者水平有限，书中若有不妥之处敬请读者批评指正。

编者

2007年11月

目 录

第 1 章 .NET 平台与 VS.NET 2005 集成开发环境	1
1.1 .NET Framework 简介	2
1.1.1 .NET Framework 的体系结构	2
1.1.2 .NET Framework 的组件	3
1.1.3 .NET Framework 的类库的组织	4
1.2 C# 语言的特点	5
1.3 VS.NET 2005 集成开发环境的介绍	6
1.3.1 起始页	8
1.3.2 设计器视图与代码视图	9
1.3.3 解决方案资源管理器	11
1.3.4 属性窗口	11
1.3.5 工具箱	12
1.3.6 子窗口的操作	13
1.4 实例: 创建 .NET 应用程序	14
1.5 VS.NET 2005 帮助系统	17
1.5.1 动态帮助	17
1.5.2 智能感知	18
1.5.3 MSDN 帮助	19
本章小结	21
练习与作业	21
上机部分(一)	22
第 2 章 C# 基础知识	27
2.1 声明 C# 中的变量和常量	27
2.1.1 变量	27
2.1.2 C# 中的常量	32
2.2 C# 中数据类型的分类	33
2.2.1 值类型	34
2.2.2 引用类型	35
2.2.3 值类型和引用类型的区别	39
2.3 装箱和拆箱	40
2.3.1 装箱操作	40
2.3.2 拆箱操作	41
2.3.3 实例 1: 身体质量指数计算	42
2.4 C# 中的运算符与表达式	43
2.4.1 运算符的分类	43
2.4.2 关系运算符	44
2.4.3 运算符的优先级	47
2.5 C# 中的选择语句	49
2.5.1 if 语句	49
2.5.2 switch 语句	53
2.6 C# 中的循环语句	55
2.6.1 while 语句	55
2.6.2 do/while 语句	57
2.6.3 for 语句	58
2.6.4 foreach 语句	61
2.6.5 实例 2: 显示地址簿	62
2.7 数组、结构和枚举	65
2.7.1 数组	65
2.7.2 结构	70
2.7.3 枚举	71
2.8 综合案例: 冒泡排序	72
本章小结	74
练习与作业	74
上机部分(二)	77
第 3 章 在 C# 中实现 OOP	80
3.1 C# 中的对象和类	80
3.1.1 类的定义	80
3.1.2 类成员	81
3.2 访问修饰符	82
3.3 构造函数和析构函数	82
3.3.1 构造函数	82
3.3.2 带参数的构造函数	85
3.3.3 析构函数	87
3.4 方法	89
3.4.1 声明方法	89

3.4.2 调用方法	90	4.6 事件	155
3.5 方法重载	92	4.6.1 定义事件	156
3.5.1 基于不同数量的参数的方法重载	93	4.6.2 预订事件	156
3.5.2 基于不同类型的参数的方法重载	94	4.6.3 通知预订事件	157
3.5.3 静态方法	96	4.7 综合案例: 处理 SARS 紧急事件	159
3.6 实例 1: 计算个人所得税	97	本章小结	163
3.7 命名空间	101	练习与作业	163
3.8 C# 中的继承	102	上机部分(四)	165
3.8.1 继承与 base 关键字	102	第 5 章 WinForms 基础知识	169
3.8.2 C# 中方法的重写	107	5.1 Windows 窗体介绍	169
3.8.3 抽象类和抽象方法	109	5.1.1 用 C# 创建 Windows 应用程序	169
3.8.4 接口及其实现	112	5.2 Windows 窗体中常用的控件	172
3.9 综合案例: 继承的实现	117	5.2.1 标签	173
本章小结	119	5.2.2 文本框	173
练习与作业	120	5.2.3 按钮	174
上机部分(三)	121	5.2.4 实例 1: 制作登录界面	175
第 4 章 高级 C# 概念	126	5.2.5 列表框	176
4.1 集合	126	5.2.6 组合框	179
4.1.1 ICollection、IList 和 IDictionary 接口	127	5.3 C# 中的消息框	180
4.1.2 ArrayList 类	129	5.4 实例 2: 创建一个使用控件的应用程序	182
4.1.3 SortedList 类	131	5.5 窗体容器简介	184
4.2 异常	132	5.5.1 窗体属性	184
4.2.1 System.Exception	133	5.5.2 窗体的常用方法和事件	185
4.2.2 try 和 catch 块	135	5.5.3 建立窗体之间的链接	187
4.2.3 使用 throw 引发异常	137	5.6 实例 3: 创建多窗体的应用程序	189
4.2.4 自定义异常对象	139	5.7 综合案例: 学生信息管理系统界面制作	190
4.2.5 使用 finally	140	本章小结	198
4.3 属性	142	练习与作业	198
4.3.1 属性的类型	145	上机部分(五)	199
4.3.2 属性约束	147	第 6 章 WinForms 控件	202
4.4 索引器	147	6.1 单选按钮	202
4.4.1 索引器的创建	147	6.2 图片框	203
4.4.2 多参数索引器	150	6.3 选项卡控件	205
4.5 委托	151	6.4 滚动条	208
4.5.1 定义委托	152	6.5 进度条	209
4.5.2 实例化委托	152	6.6 其他高级控件(定时器、打印等控件)	211
4.5.3 使用委托	153	6.7 自定义控件	213
4.5.4 匿名方法	154		

6.7.1 用户控件	214	8.4.1 透明窗体、不规则窗体的实现	283
6.7.2 自定义控件	215	8.4.2 控件停靠与控件锚定	287
6.8 综合案例: 随机数游戏	217	8.4.3 可视化继承	288
本章小结	218	8.5 综合案例: 浏览器制作	292
练习与作业	219	本章小结	304
上机部分(六)	219	练习与作业	304
第7章 WinForms 应用程序对话框		上机部分(八)	205
设计	222	第9章 调试、测试与程序的打包、部署	
7.1 对话框特点与分类	222	部署	309
7.2 使用系统定义的对话框	224	9.1 调试简介	309
7.2.1 打开文件对话框	224	9.1.1 调试的必要性	309
7.2.2 保存文件对话框	228	9.1.2 调试过程	310
7.2.3 字体对话框	231	9.1.3 Visual Studio.NET 中的调试工具	313
7.2.4 颜色对话框	232	9.1.4 实例1: 在 VS.NET 2005 中	
7.2.5 其他对话框(MessageBox)	234	调试程序	316
7.2.6 实例1: 对话框应用	235	9.2 测试简介	317
7.3 创建自定义对话框	240	9.3 程序的打包和部署	327
7.3.1 自定义对话框介绍	240	9.3.1 帮助文件的制作	327
7.3.2 实例2: 自定义对话框	243	9.3.2 程序打包	329
7.4 综合案例: 写字板制作	245	9.3.3 程序部署	329
本章小结	257	9.4 综合案例: 简易计算器制作、测试、	
练习与作业	257	打包和部署	340
上机部分(七)	258	本章小结	349
第8章 WinForms 应用程序高级		练习与作业	350
界面设计	260	上机部分(九)	350
8.1 SDI 和 MDI 简介	260	第10章 使用 ADO.NET 进行数据库	
8.1.1 多文档界面(MDI)应用程序	262	访问	353
8.1.2 创建 MDI 应用程序	263	10.1 数据库概述	353
8.2 窗体菜单设计	264	10.2 ADO.NET 的基本组件	360
8.2.1 菜单概述及属性	264	10.3 使用 ADO.NET	363
8.2.2 MainMenu 菜单控件与 ContextMenu		10.3.1 使用 Connection 对象	363
菜单控件	265	10.3.2 使用 Command 对象	367
8.2.3 实例1: 菜单制作	273	10.3.3 使用 DataReader 对象	371
8.3 窗体工具栏和状态栏设计	276	10.3.4 使用 DataAdapter 对象	374
8.3.1 工具栏设计	276	10.3.5 使用 DataSet 对象	379
8.3.2 状态栏设计	278	10.4 综合案例: 图书管理系统——使用	
8.3.3 实例2: 制作带工具栏和状态栏的		DataGrid 显示和操作数据	383
应用程序	280	本章小结	396
8.4 实现用户界面增强功能	283	练习与作业	396

上机部分(十).....	397
第 11 章 综合案例: 学生信息管理	413
系统	413
11.1 需求概述.....	413
11.2 系统总体设计.....	414
11.2.1 功能模块设计.....	414
11.2.2 数据模块设计.....	415
11.3 系统的实现.....	417
11.3.1 系统登录模块.....	420

11.3.2 系统主模块.....	422
11.3.3 用户管理模块.....	425
11.3.4 学籍信息管理模块.....	428
11.3.5 课程班级管理模块.....	431
11.3.6 成绩信息管理模块.....	433
11.3.7 奖罚信息管理模块.....	435

11.3.2 系统主模块.....	422
11.3.3 用户管理模块.....	425
11.3.4 学籍信息管理模块.....	428
11.3.5 课程班级管理模块.....	431
11.3.6 成绩信息管理模块.....	433
11.3.7 奖罚信息管理模块.....	435

11.3.2 系统主模块.....	422
11.3.3 用户管理模块.....	425
11.3.4 学籍信息管理模块.....	428
11.3.5 课程班级管理模块.....	431
11.3.6 成绩信息管理模块.....	433
11.3.7 奖罚信息管理模块.....	435

11.3.2 系统主模块.....	422
11.3.3 用户管理模块.....	425
11.3.4 学籍信息管理模块.....	428
11.3.5 课程班级管理模块.....	431
11.3.6 成绩信息管理模块.....	433
11.3.7 奖罚信息管理模块.....	435

11.3.2 系统主模块.....	422
11.3.3 用户管理模块.....	425
11.3.4 学籍信息管理模块.....	428
11.3.5 课程班级管理模块.....	431
11.3.6 成绩信息管理模块.....	433
11.3.7 奖罚信息管理模块.....	435

第1章 .NET 平台与 VS.NET 2005 集成开发环境

学习目标

- 了解 .NET 平台的构成
- 熟悉 VS.NET 2005 集成开发环境(IDE)
- 编写一个基于 .NET 平台的应用程序

.NET 是微软推出的一种全新的开发平台。对 .NET 的需求源于 Internet 的高速发展。在 Internet 高速发展的背景下应用程序不再单一地是桌面运行的可执行文件，应用程序的模式已经发生了根本的变化，这些变化主要体现在以下方面：

- (1) 应用程序应随时、随地并且在任何设备中都可用。
- (2) 软件作为一种服务来提供，例如电子邮件服务。
- (3) 应用程序必须具有互操作性，应用程序之间协同工作、数据交换的需求也越来越高。

Microsoft .NET 就是在这样的大环境下诞生的，.NET 刺激了下一代计算技术的发展，加速了新一代 Internet 的来临，并为后一代应用程序的开发提供了便利。它为创建、部署以及管理安全、强大、高效的应用程序提供了前所未有的最大支持。

通过 .NET 平台，我们不仅可以开发出功能更强大的应用程序，而且还可以开发出丰富的 Web 应用和服务。

.NET 有一个庞大的技术体系，读者如何制定自己的学习步骤呢？图 1-1 为读者显示了 .NET 技术学习全景图。

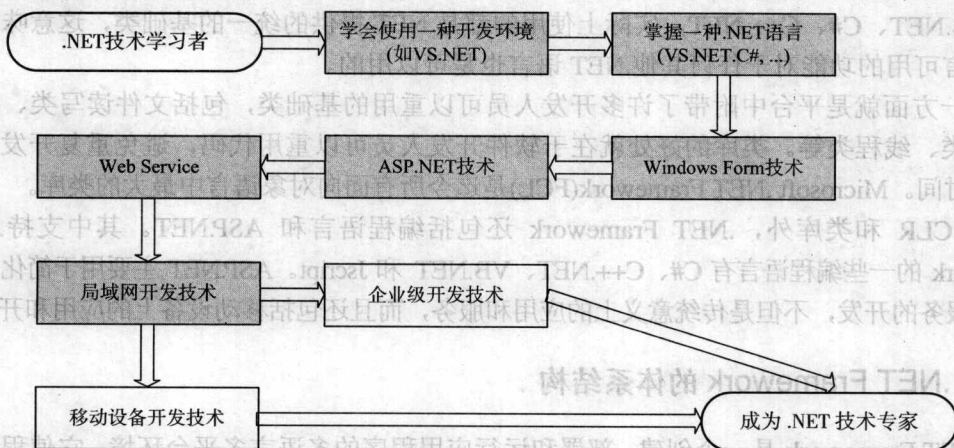


图 1-1 .NET 技术学习全景图

通过图 1-1 可以看出，.NET 平台中的程序设计技术主要分为以下四个方面：

Windows Forms 技术：可视化程序设计，主要用来开发 Windows 应用程序。

ASP.NET 技术：主要用来设计和开发网站。

Web Service 技术：主要用来开发基于互联网的 Web 服务。

移动设备开发技术：主要用来开发手机等移动设备的软件和服务。

学习需要循序渐进，在掌握上述四大核心开发技术之前，我们必须先了解 .NET 平台的构成，然后学会使用基于 .NET 平台的开发环境 VS.NET 和熟练掌握一种支持 .NET 的核心开发语言 C#。在此基础上，本书主要为读者介绍 Windows Forms 程序设计技术。

1.1 .NET Framework 简介

Framework 即框架，是开发人员对编程语言命令集的称呼。.NET Framework 的意义就在于只用统一的命令集即可支持任何的编程语言。

.NET Framework 是 .NET 平台的基础架构，我们通常所说的 .NET 平台实际上就是泛指 .NET Framework，它是 Microsoft 为开发应用程序而创建的一个富有革命性的新平台。具体讲，它就是一个软件包，里面定义了 .NET 应用程序的开发和运行环境，以及可以在 .NET 应用程序中使用的类库，类似于 Java 的虚拟机。它不仅便于开发 Internet 上运行的应用程序，而且也可用于开发运行于 Windows 桌面上的传统业务应用程序。

.NET Framework 包括最早期的 1.0 版，目前流行的 1.1 版和 2.0 版，还有正在测试的 3.0 版。读者可以从微软公司的网站或其他共享站点下载 .NET Framework 安装包。安装后，你的电脑就具备了开发 .NET 应用程序的运行环境和在 .NET 应用程序中使用的类库。

.NET Framework 包含以下两个主要组件：

➤ 公共语言运行时(Common Language Runtime, CLR)。

➤ 统一的类库集。

对于技术开发人员来说，.NET Framework 的两大特性是非常卓越的。首先是其所提供的公共语言运行时(CLR)对多语言的支持，任何 .NET 语言都可以使用 .NET 类库中的所有类，例如 VB.NET、C#、C++.NET，实际上使用的都是 .NET 提供的统一的基础类，这意味着对一种语言可用的功能对于任何其他 .NET 语言也是可以用的。

另一方面就是平台中附带了许多开发人员可以重用的基础类，包括文件读写类、数据库访问类、线程类等。类库的好处就在于软件开发人员可以重用代码，避免重复开发并缩短开发时间。Microsoft .NET Framework(FCL)是迄今所有面向对象语言中最大的类库。

除 CLR 和类库外，.NET Framework 还包括编程语言和 ASP.NET。其中支持 .NET Framework 的一些编程语言有 C#、C++.NET、VB.NET 和 Jscript。ASP.NET 主要用于简化 Web 应用和服务的开发，不但是传统意义上的应用和服务，而且还包括移动设备上的应用和开发。

1.1.1 .NET Framework 的体系结构

.NET Framework 是一个创建、部署和运行应用程序的多语言多平台环境。它使程序员能够开发用于 Windows、Internet、Pocket PC(个人掌上电脑)、Smart Phone(智能电话)等多种平台的应用程序。此外，它还提供了多种编程语言可供选择。图 1-2 显示了 .NET Framework 的体系结构。

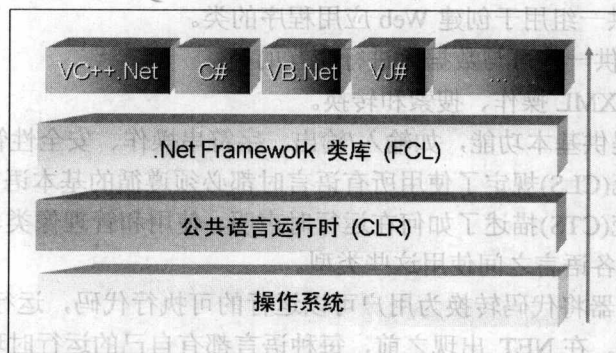


图 1-2 .NET Framework 的体系结构

程序员可以使用 .NET 支持的语言之一来开发应用程序。这些应用程序使用的基类库由 .NET Framework 类库(FCL)提供。例如，要显示一个对话框，可以编写如下代码：

```
MessageBox.Show(“我是一个对话框”)
```

其中的 MessageBox 就是类库中的一个类。Show 是这个类的用来显示对话框的方法。

从图 1-2 可以看出，只要支持 .NET Framework，则表明其支持 .NET 的开发，这与支持 JVM 就支持 Java 开发的机制是一样的，程序通过使 FCL 成为所有 .NET 语言的公共类库来实现。此外，.NET Framework 的公共语言运行时还管理 .NET 应用程序的执行。

C# 是 .NET 中的核心语言，也是使用频率最高的语言，在本书中，我们将主要介绍如何通过 C# 语言来进行 .NET 开发。

1.1.2 .NET Framework 的组件

.NET Framework 由框架类库(FCL)和公共语言运行时(CLR)这两个主要组件组成。图 1-3 显示了 .NET Framework 的组件及其要素。



图 1-3 .NET Framework 的组件

其中：

- Web Forms 提供一组网页设计的类，与 HTML 表单很相似。
- Web Services 包括一组设计 Web 服务的类，可以用来构造网络服务的应用程序。
- WinForms 提供一组 Windows 的窗体应用程序设计和开发的类。

- ASP.NET 提供一组用于创建 Web 应用程序的类。
- ADO.NET 提供一些可与数据库进行交互的类。
- XML 类启用 XML 操作、搜索和转换。
- 基本框架类提供基本功能，如输入/输出、字符串操作、安全性管理和网络通信等。
- 通用语言规范(CLS)规定了使用所有语言时都必须遵循的基本语言功能的子集。
- 公共类型系统(CTS)描述了如何在运行时声明、使用和管理像类、结构、枚举、接口等类型，从而便于在各语言之间使用这些类型。

通常，语言编译器将代码转换为用户可以运行的可执行代码，运行时环境为可执行代码提供操作系统服务。在 .NET 出现之前，每种语言都有自己的运行时环境。例如，C++ 带有一个名为 MSVCRT.DLL 的 DLL 文件，以负责 C++ 程序的执行，Visual Basic 也带有一个名为 MSVBVM60.DLL 的 DLL 文件。

.NET Framework 的主要目的之一是将各种运行时环境结合起来，使开发人员可以使用单一的运行时服务。因此，.NET Framework 提供了公共语言运行时(CLR)，CLR 为与 .NET Framework 配合使用的任何语言提供了诸如内存管理、安全性和错误处理等功能。.NET 中的所有代码都由 CLR 来管理，因此称为“托管代码”。托管代码包含关于代码的信息，例如在代码中定义的类、方法和变量。

当编译用兼容 .NET 语言编写代码时，输出的代码为微软中间语言(即 MSIL)的形式。MSIL 是由一组指示代码应该如何执行代码的特定指令组成。MSIL 不是为物理 CPU 开发的特定指令集。也就是说，它对计算机中的 CPU 完全不了解，同样计算机也不了解 MSIL。那么 .NET 应用程序如何执行呢？当第一次运行代码时，MSIL 代码将被转换为特定于 CPU 的代码，这个过程称为“即时”编译或 JIT。即时是指在需要时才编译 MSIL 代码。JIT 编译器的主要任务是将一般的 MSIL 代码转换为可由 CPU 直接执行的计算机代码。

总之，CLR 的作用除了对代码进行管理之外，同时要承担起即时编译的作用。图 1-4 描述了 .NET 支持语言、CLR、JIT、MSIL 和可执行代码之间的关系。

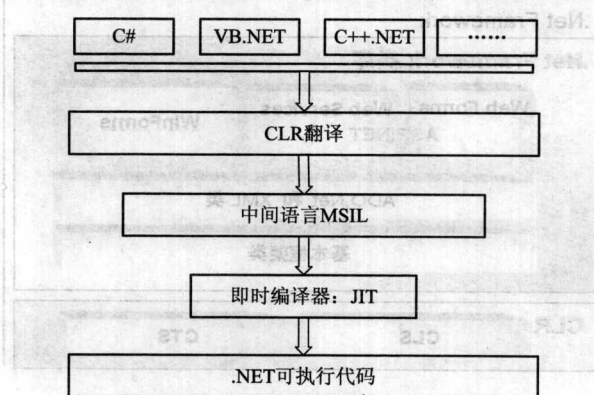


图 1-4 .NET 应用程序执行过程

1.1.3 .NET Framework 的类库的组织

.NET Framework 包含了庞大类库，类库中的成千上万个类可能名称相同或功能相似，

如何在逻辑上将这些类进行组织和分类呢? .NET Framework 中采用了命名空间(namespace)来将类库中具有相似功能的类在逻辑上进行分组。程序员通过命名空间可以快速查看和方便地使用该命名空间下的类。命名空间就好比容纳相似类物品的容器。例如,一所大学中的人员可以分别属于不同的院系、部门或班级。这些院系、部门或班级的名称就是命名空间的名字,它起到的作用就是方便管理教师和学生。命名空间有助于改善数据的构成,从而使每个人都可以轻松地获得想要的数据库。

在 .NET Framework 中,所有命名空间基本上都是由称为 System 的公共命名空间形成的。这就是 System 命名空间又称为根命名空间的原因。它实际上就像一个根,所有其他的命名空间都是从根命名空间延伸而形成的。System 命名空间包含了定义 .NET 中使用的公共数据类型(包括 Int32、Boolean 和 DateTime)等。此命名空间中包括的另一个重要数据类型为“Object”,Object 数据类型形成了所有其他 .NET 对象继承的基本对象。表 1-1 为 System 命名空间下的一些最常用的二级命名空间的名字及功能。

表 1-1 System 根命名空间下的部分二级命名空间

命名空间	说明
System. Drawing	处理图形和绘图,包括打印
System. Data	处理数据存取和管理,在定义 ADO.NET 技术中扮演重要角色
System. IO	管理对文件和流的同步和异步访问
System. Windows	处理基于窗体的窗口的创建
System. Reflection	包含从程序集读取元数据的类
System. Threading	包含用于多线程编程的类
System. Collections	包含定义各种对象集的接口和类

以上命名空间基本上是本书将要讲述的。

1.2 C# 语言的特点

C#语言是从 C/C++演变而来的。它是现代、简单、完全面向对象和类型安全的。如果您是 C/C++程序员,学习曲线将会很平坦。许多 C# 语句直接借用您所喜爱的语言,包括表达式和操作符。C# 是现代的编程语言。C# 减少了 C++的一些特性,不再有宏、模板和多重继承,使编程更方便,而且具有类型安全、版本控制、垃圾收集(garbage collect)等功能。所有的这些功能的目的都是开发面向组件的软件,因此也把 C# 称为面向组件的语言。

Visual C#(又称 Visual C Sharp,简称 C#)是美国微软公司推出的 Windows 操作系统下的应用程序开发平台 Visual Studio.NET(简称 VS.NET)家庭中的一员(其它成员包括 Visual C++,简称 VC++; Visual Basic .NET,简称 VB.NET; Visual J#,简称 VJ#等)。

C# 是专门为 .NET 应用而开发出的语言。它从根本上保证了 C# 与 .NET 框架的完美结合。在 .NET 运行库的支持下,.NET 框架的各种优点在 C# 中表现得淋漓尽致。读者可以先来看看 C# 的一些突出的特点,相信以后的学习过程中,你将会深深地体会到“#”

——“SHARP”(锋利)的真正含义。它具有如下特点:

➤ 简洁的语法。

- 精心的面向对象设计。
- 与 Web 的紧密结合。
- 完整的安全性 with 错误处理。
- 版本处理技术。
- 灵活性与兼容性。

学过 Java 的读者可能发现 C# 和它非常相似, 其实微软就是为了对抗 Sun 公司的 Java 语言而推出 C# 语言的。我们可以把它们俩看成是表兄弟关系, 它们的舅舅都是 C++。随着近五年的发展, C# 已经成为软件开发领域一门开发效率高, 功能强大, 深受广大程序员喜爱的流行开发语言。它不仅可以开发 Windows 桌面应用程序, 也可以开发 Web 应用程序。在当今和未来的开发岗位上, C# 程序员将是一个十分热门的工作岗位。

1.3 VS .NET 2005 集成开发环境的介绍

要做好 .NET 应用程序的开发, 读者要先熟练掌握 .NET 的开发工具——Visual Studio.NET (简称 VS.NET), 它是微软 .Net 平台上的一个功能强大的、集成多种开发语言的软件开发工具。通过该开发工具, 大多数 .Net 编程语言都可以实现快速开发。它作为一款集成开发环境, 将程序编辑、调试、测试、打包、部署等操作集成在同一个开发平台上, 大大提高了程序员的开发效率。

Visual Studio.NET 可用于多种类型的程序开发, 包括基于 Web 的应用程序、XML Web Service、基于 Windows 的应用程序和移动应用程序等。Visual Studio.NET 提供了一系列核心程序设计语言——VB .NET、VC++、VC# 和 VJ#。它允许这些编程语言共享工具, 甚至可以创建由不同编程语言混合在一起的应用程序, 从而结束了各种不同的编程语言只能各自使用独立的程序设计界面的局面。这些编程语言都可以利用 .NET Framework 的功能。

目前 VS.NET 开发平台的版本有 VS.NET 2003、VS.NET 2005 和 VS.NET 2008 测试版。其中 VS.NET 2005 是 Visual Studio.NET 系列中一个比较新的版本, 也是目前最成熟和使用最广泛的版本。本书将以 VS.NET 2005 为开发环境介绍 .NET 程序的开发。

下面先介绍在 VS.NET 2005 开发环境中配置 C# 的开发设置。初次运行 VS.NET 2005 将打开“选择默认环境设置”对话框, 如图 1-5 所示。在“选择默认环境设置”列表框中选择“Visual C# 开发设置”选项, 单击【启动 Visual Studio】按钮, 经过配置后打开 Visual Studio 2005 界面。

如果在使用的过程中想改变默认环境设置, 则可以执行“工具”菜单中的“导入和导出设置”命令, 即打开“导入和导出设置向导”对话框, 进行必要的设置, 从而使集成开发环境更适合 C# 程序的设计。

在如图 1-6 所示的“欢迎使用‘导入和导出设置向导’”对话框中, 选择“否; 仅重置设置, 从而改写我的当前设置”单选项, 如图 1-7 所示。单击【下一步】按钮。

在“选择一个默认设置集合”对话框中的“要重置为哪个设置集合”列表框中, 选择“Visual C# 开发设置”选项, 如图 1-8 所示。单击【完成】按钮, 系统重置后, 打开向导的“重置完成”对话框, 如图 1-9 所示。单击【关闭】按钮, 完成设置。