



普通高等教育“十一五”国家级规划教材

高职高专计算机系列规划教材

C# 程序设计教程

(第3版)

C# Programming Tutorial (Third Edition)

刘甫迎 主编

刘光会 王蓉 刘焱 副主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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



第 1 章 绪论

第 2 章 C# 语言基础

C# 程序设计教程

(第 2 版)

C# Programming: Fundamentals and Examples

作者：[美] 詹姆斯·H·穆勒

译者：李永刚、王 强、王 磊、王 磊

清华大学出版社

http://www.tup.tsinghua.edu.cn

普通高等教育“十一五”国家级规划教材
高职高专计算机系列规划教材

C# 程序设计教程

(第3版)

刘甫迎 主编

刘光会 王 蓉 刘 焱 副主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

微软的 C# (VS 2012 版) 是从 C 和 C++ 演变而来的一种新的面向对象的程序设计语言, 用它结合 ASP.NET 或 Windows Phone 平台的开发应用代表了当前的编程方向。C# 可以编译成跨平台的代码, 它避免了 C 语言的指针和多继承, 简单易学且功能强, 其协变和逆变、LINQ 及 PLINQ 等新特点皆受到人们青睐。本书共 12 章和两个附录, 详述了 C# 的由来和发展, 数据类型、运算符与表达式, 结构化程序设计及算法, 数组、结构、枚举和集合, 面向对象程序设计 (类、委托、继承和接口), 可视化应用程序设计, 文件和流, 数据库操作, C/S 模式编程, 多线程和多窗体等。最后给出一个较完整的编程案例。

本书可作为高等学校及软件学院的教材, 也适合从事软件开发和应用的人员参考。

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。

图书在版编目 (CIP) 数据

C# 程序设计教程/刘甫迎主编. —3 版. —北京: 电子工业出版社, 2012.8

(普通高等教育“十一五”国家级规划教材·高职高专计算机系列规划教材)

ISBN 978-7-121-17325-7

I. ①C… II. ①刘… III. ①C 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 121430 号

策划编辑: 吕 迈

责任编辑: 吕 迈

印 刷:

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 22 字数: 563 千字

印 次: 2012 年 8 月第 1 次印刷

印 数: 4 000 册 定价: 37.80 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

/ 前 言 /

C#是从 C 和 C++演变而来的一种新的面向对象的编程语言。结合了 C#的强大功能和 Visual Studio.NET 平台的环境和类的支持后,形成了功能强大的开发工具——C#.NET。C#是目前主流语言中不多见的在 Internet 出现后推出的语言,故它的设计很适合在国际互联网上使用。C#结合 ASP.NET 或 Windows Phone 平台开发应用程序,代表了当前的编程方向。C#可以编译成跨平台的代码,它避免了 C 语言中的指针和多继承,简单易学且功能强。

目前,国内高校该类课程的教学内容一般理论过多,笔者认为主要应加强编程实践,提高动手能力。很有必要改变课程的教学内容和教学模式,用先进的教学理念和方法培养一流人才,本教材就是在此改革的思路下编写的,其主要特点如下。

(1)《C# 程序设计教程(第 3 版)》介绍了 VS2012(C#在其中)等的新特点,并以 C# 2010(C# 4)版平台为基础进行了内容更新和添加。所用的 Access 数据库也升级了版本。

(2)增加了泛型、泛型集合 List<T>、ICollection 接口、IEnumerable<T>接口、yield,以及泛型接口和委托中的协变、逆变的内容,见第 7.4 节。

(3)增加了 10.6 节“LINQ 语言集成查询以及 PLINQ”(Var、Lambda、查询表达式转换和 LINQ To SQL);第 6 章增加了对象初始化器;第 9 章加强了 C#的文件中流的概念。

(4)增加“C#的多线程、多窗体”,将原第 11 章 Web 服务放到笔者另外的 B/S 的书中。

(5)精炼了第 12 章 C#案例——人力资源管理系统的內容。

(6)区分各课程在该专业的不同技能培养目标。C#程序设计作为计算机软件类专业基础课,其目标重点是 C/S 模式编程,即对专业分解到该课程的结构化程序设计、面向对象程序设计、可视化编程、ADO.NET 与库表、.NET 平台等能力模块进行教学;而把 XML、正则表达式、WPF、Ajax、WCF、ListView 富数据控件等放笔者另外一部书(ASP.NET B/S)中。

(7)在选材上,继续强调理论“以必需、够用为度”,减少了个别章节的难度,且深入浅出。另外,书中例子使用了许多经典算法,弥补了有些读者未学《数据结构》的不足。

(8)增强实践性,各章均有不少经过运行验证的例题与实例(例如,10.7 节 C#应用实例——学生成绩管理系统)以及第 12 章较完整的案例(用 UML、CASE、MVC 等技术)。

(9) 辅助教学资源丰富, 打造成立体化精品教材。在其精品课程网站上, 有配套的 PPT、教学大纲、习题、实验指导书、专周课程设计、教师在线辅导、CAI、网络课程、试题库、考试系统、相关素材等, 这些资源生动、具体、形象、直观, 便于教与学。

本书由刘甫迎教授主编, 刘光会、王蓉、刘焱老师任副主编。刘甫迎编写第 1 章、7.4 节、第 12 章; 刘光会编写第 2 章至第 6 章和附录; 王蓉编写第 7 章、第 9 章、第 10 章; 刘焱编写第 8 章、6.2.4 节、10.6 节、第 11 章, 全书由刘甫迎统稿。其他参编人员有: 周绍敏、李朝蓉、党晋蓉、李琦、李飞跃、杨雅志、龚茗茗、谢春、饶斌、陈振梁、樊婷婷等。对电子工业出版社的编辑在本书出版过程中给予的帮助表示感谢。由于水平有限, 错误难免, 请斧正。

(Email: lifox@mail.cec.edu.cn)

刘甫迎
2012 年 4 月

目 录

CONTENTS

第 1 章 C#概述	1
1.1 C#简介	1
1.1.1 C#的发展	1
1.1.2 C#的特点	2
1.1.3 VS2012 等的新功能	4
1.1.4 Microsoft.NET 平台	7
1.1.5 C#的运行环境及安装	9
1.2 C#集成开发环境	12
1.2.1 C#的启动	12
1.2.2 C#集成开发环境	14
习题	19
第 2 章 简单的 C#程序设计	20
2.1 C#程序结构	20
2.1.1 第一个 C#程序	20
2.1.2 编译和执行程序	21
2.1.3 C#程序结构分析	22
2.1.4 标识符	24
2.2 输入/输出操作	25
2.2.1 Console.WriteLine()方法	25
2.2.2 Console.Write()方法	28
2.2.3 Console.ReadLine()方法	29
2.2.4 Console.Read()方法	29
习题	30
第 3 章 数据类型、运算符与表达式	31
3.1 数据类型	31
3.1.1 值类型	31
3.1.2 引用类型	34
3.1.3 类型转换	36
3.1.4 装箱和拆箱	38

3.2 常量与变量.....	39
3.2.1 常量.....	40
3.2.2 变量.....	40
3.3 运算符和表达式.....	42
3.3.1 运算符.....	42
3.3.2 表达式.....	50
习题.....	52
第4章 结构化程序设计	54
4.1 结构化程序设计的概念.....	54
4.1.1 结构化程序设计的概念及算法的概念.....	54
4.1.2 流程图.....	55
4.2 顺序结构.....	56
4.2.1 顺序结构的概念.....	56
4.2.2 顺序结构的实例.....	56
4.3 选择结构.....	57
4.3.1 选择结构的概念.....	57
4.3.2 条件语句.....	58
4.3.3 分支语句.....	63
4.3.4 选择结构的实例.....	66
4.4 循环结构.....	67
4.4.1 循环结构的概念.....	68
4.4.2 while 语句和 do-while 语句.....	68
4.4.3 for 语句和 foreach 语句.....	71
4.4.4 跳转语句.....	74
4.4.5 循环语句的算法——循环结构的实例.....	77
4.5 异常处理.....	79
4.5.1 异常处理的概念.....	80
4.5.2 异常的处理.....	81
习题.....	85
第5章 数组、结构和枚举.....	88
5.1 数组.....	88
5.1.1 数组的概念.....	88
5.1.2 一维数组.....	89
5.1.3 二维数组.....	93
5.1.4 “冒泡排序”算法——数组的实例.....	96
5.2 结构类型.....	98
5.2.1 结构的声明.....	98
5.2.2 结构成员的访问.....	99
5.2.3 顺序查找算法——结构的实例.....	100

5.3 枚举	102
5.3.1 枚举类型的定义	102
5.3.2 枚举成员的赋值	103
5.3.3 枚举成员的访问	105
习题	107
第6章 C#的面向对象程序设计	110
6.1 面向对象程序设计的基本概念	110
6.2 类和对象	113
6.2.1 类的声明	114
6.2.2 类的成员	115
6.2.3 对象的声明	117
6.2.4 对象初始化器	119
6.3 构造函数和析构函数	120
6.3.1 构造函数	120
6.3.2 析构函数	124
6.4 方法	125
6.4.1 方法的定义及调用	125
6.4.2 方法的参数类型	127
6.4.3 方法的重载	130
6.4.4 静态方法与非静态方法	131
6.4.5 运算符的重载	131
6.4.6 递归	134
6.5 属性	135
6.5.1 属性的声明	136
6.5.2 属性的访问	137
6.6 索引指示器	139
6.6.1 定义索引指示器	139
6.6.2 使用索引指示器访问对象	140
6.7 委托与事件	141
6.7.1 委托	141
6.7.2 事件	143
6.8 C#常用的基础类	145
6.8.1 Math 类与 Random 类	146
6.8.2 字符串	147
6.9 命名空间	150
6.9.1 命名空间的声明	150
6.9.2 命名空间的成员	151
6.9.3 命名空间的使用	152
习题	153

第 7 章 继承和接口	156
7.1 C#的继承机制	156
7.1.1 继承的基本知识	156
7.1.2 base 关键字	158
7.1.3 覆盖	160
7.2 多态性	161
7.2.1 多态性概述	161
7.2.2 虚方法	162
7.2.3 多态性的实现	162
7.3 接口	164
7.3.1 接口的定义	164
7.3.2 接口的成员	165
7.3.3 接口的实现	169
7.3.4 接口的应用实例	185
7.4 泛型、泛型接口和委托中的协变、逆变	187
7.4.1 泛型、泛型集合 List<T>、IEnumerable<T>接口及 yield	187
7.4.2 协变和逆变	191
7.4.3 泛型接口和委托中的协变和逆变	191
习题	195
第 8 章 可视化应用程序设计	196
8.1 Windows 窗体	196
8.1.1 Windows 窗体的基本属性	197
8.1.2 创建窗体	199
8.1.3 设置启动窗体属性	199
8.2 控件概述	200
8.2.1 控件的基本属性	200
8.2.2 控件共有的属性、事件和方法	201
8.3 命令按钮控件 (Button)	209
8.4 标签控件 (Label)	211
8.5 文本框控件 (TextBox)	213
8.6 单选按钮 (RadioButton) 和复选框 (CheckBox 和 CheckedListBox)	214
8.7 面板控件 (Panel) 和分组框 (GroupBox) 控件	219
8.8 图形框控件 (PictureBox)	221
8.9 列表框 (ListBox)、带复选框的列表框 (Checked ListBox) 和组合框 (ComboBox) 控件	223
8.9.1 列表框控件 (ListBox)	223
8.9.2 带复选框的列表框控件 (CheckedListBox)	225
8.9.3 组合框控件 (ComboBox)	226
8.10 滚动条控件 (ScrollBar)	228
8.11 定时器控件 (Timer)	230
8.12 超链接标签控件 (LinkLabel)	231

8.13 通用对话框控件.....	232
8.13.1 消息框.....	232
8.13.2 文件打开和保存对话框.....	233
8.13.3 颜色选择对话框.....	236
8.13.4 字体对话框.....	237
8.13.5 打印机对话框.....	238
8.13.6 对话框综合实例.....	240
习题.....	246
第 9 章 C#的文件和流.....	247
9.1 文件和流的概念.....	247
9.2 文件的读和写.....	252
9.2.1 读文件.....	252
9.2.2 写文件.....	253
9.3 文件操作实例——链表算法.....	254
习题.....	259
第 10 章 C#的数据库操作.....	260
10.1 Access 数据库.....	260
10.1.1 建立 Access 数据库表.....	260
10.1.2 输入表的数据.....	264
10.1.3 SQL 语言简介及常用的 SQL 命令.....	264
10.2 客户机/服务器 (C/S) 模式编程.....	268
10.3 ADO.NET 概念.....	269
10.4 ADO.NET 对象.....	270
10.4.1 Connection 对象.....	270
10.4.2 Command 对象.....	272
10.4.3 DataReader 对象.....	273
10.4.4 DataAdapter (数据适配器) 对象.....	274
10.4.5 DataSet 对象.....	274
10.5 数据绑定控件.....	276
10.6 LINQ 语言集成查询.....	278
10.6.1 LINQ 以及 PLINQ 的概念.....	278
10.6.2 LINQ 语法 (Var、扩展方法、Lambda、匿名类型和查询表达式转换).....	280
10.6.3 关系数据的集成查询 LINQ To SQL 应用.....	282
10.7 C#应用实例——学生成绩管理系统.....	283
习题.....	302
第 11 章 C#的多线程、多窗体 (MDI) 应用.....	303
11.1 C#的多线程编程.....	303
11.1.1 多线程简介.....	303
11.1.2 线程的优先级和执行状态.....	307

11.1.3 线程同步	308
11.2 C#的多窗体 (MDI) 应用程序	310
习题	314
第 12 章 C#案例——人力资源管理系统	315
12.1 系统概述	315
12.1.1 办公自动化系统 FoxOA	315
12.1.2 系统功能	315
12.2 系统分析与设计 (UML)	318
12.2.1 计算机辅助软件工程 (CASE) 工具——PD	318
12.2.2 本系统建模	318
12.3 主要关键技术	321
12.3.1 MVC 设计模式简介	321
12.3.2 MVC 设计模式的实现	321
12.4 数据库结构	322
12.5 用户使用说明书	323
12.5.1 系统平台	323
12.5.2 登录和进入系统	324
12.6 关键源程序	325
习题	333
附录 A 《C#程序设计课程》教学大纲	334
附录 B 《C#程序设计课程》实验指导书	337
参考文献	341

本章介绍 C# 的由来和发展、特点、VS 2012 等的新功能、Microsoft.NET 平台以及 C# 的运行环境，使读者能够初步了解 C#，并能够进行 C# 的安装、启动和熟悉 C# 的界面。

1.1 C#简介

1.1.1 C#的发展

美国微软公司的 Visual Studio.NET 已到 2012 版了，其代号为 Ark，方舟。Visual C#（简称 VC# 或 C#，本书统一称 C#）包含在其中，然而 C# 是如何发展而来的呢？

1995 年，SUN 公司正式推出了面向对象的开发语言 Java，并具有跨平台、跨语言的功能特点，Java 逐渐成了企业级应用系统开发的首选工具，而且使得越来越多的基于 C/C++ 的应用开发人员转向于从事基于 Java 的应用开发。

在 Java 势头很猛的软件开发领域可观前景的冲击下，作为世界上最大的软件公司微软立即做出了迎接挑战的反应。很快，微软也推出了基于 Java 语言的编译器 Visual J++。Visual J++ 在最短的时间里由 1.1 版本升级到了 6.0 版本。Visual J++ 6.0 集成在 Visual Studio 6.0 中，不但虚拟机（Java Virtual Machine, JVM）的运行速度大大加快，而且增加了许多新特性，同时支持调用 Windows API，这些特性使得 Visual J++ 成为强有力的 Windows 应用开发平台，并成为业界公认的优秀 Java 编译器。

Visual J++ 虽然具有强大的开发功能，但主要应用在 Windows 平台的系统开发中，SUN 公司认为 Visual J++ 违反了 Java 的许可协议，即违反了 Java 开发平台的中立性，因而，对微软提出了诉讼，这使得微软处于极为被动的局面。为了改变这种局面，微软另辟蹊径，决定推出其进军互联网的庞大 .NET 计划，和该计划中重要的开发语言——Visual C#。

微软的 .NET 是一项非常庞大的计划，也是微软发展的战略核心。Visual Studio .NET 则是微软 .NET 技术的开发平台，C# 就集成在 Visual Studio .NET 中。.NET 代表了一个集合、一个环境、一个编程的基本结构，作为一个平台来支持下一代的互联网（本书 1.1.4 将详细介绍 .NET 平台）。为了支持 .NET 平台，Visual Studio .NET 在原来的 Visual Studio 6.0 的

基础上进行了极大的修改和变更。在 Visual Studio .NET 测试版中 Visual J++ 消失了,取而代之的就是 C# 语言。

美国微软公司在 2000 年 6 月份举行的“职业开发人员技术大会”上正式发布了 C# 语言,其英文名为 Visual C-Sharp。微软公司对 C# 的定义是:“C# 是一种类型安全的、现代的、简单的,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++ 开发人员所熟悉。C# 的目的就是综合 Visual Basic 的高生产率和 C++ 的行动力。”

目前使用 C# 进行 C/S (客户/服务器) 结构编程或用 C# 与 ASP.NET 结合进行 B/S (浏览器/服务器) 结构编程的人员越来越多,用 C# 进行编程已成为今后程序设计的趋势之一,而且将逐步超越其他主流编程语言的地位。为什么会这样呢? 请看下节所讲述的 C# 的特点。

1.1.2 C# 的特点

作为微软新一代面向对象的语言产品,C# 语言自 C/C++ 演变而来,它是给那些愿意牺牲 C++ 一点底层功能,以获得更方便和更产品化的企业开发人员而创造的。C# 具有现代、简单、完全面向对象和类型安全等特点。

如果读者是 C/C++ 程序员,学习将会变得很容易。许多 C# 语句直接借用程序员所喜爱的语言,包括表达式和操作符。假如不仔细看,就会把它当成 C++。

关于 C# 最重要的一点:它是现代的编程语言。它在类、名字空间、方法重载和异常处理等领域简化了 C++。摒弃了 C++ 的复杂性,使它更易用、更少出错。

对 C# 的易用有贡献的是减少了 C++ 的一些特性,不再有宏、模板和多重继承。特别对企业开发者来说,上述功能只会产生更多的麻烦而不是效益。

使编程更方便的新功能是严格的类型安全、版本控制、垃圾收集 (garbage collect) 等。所有这些功能的目标都是瞄准了开发面向组件的软件。

1. 可避免指针等,语法更简单、易学

C# 中可避免复杂的令人头痛的 C++ 中流行的指针,禁止直接内存操作,不能使用 “::” 和 “→” 运算符,整型数据 0 和 1 也不再是布尔值,“==” 被用于比较操作而 “=” 被用作赋值操作,从而减少了运算符错误。C# 使用统一的类型系统,摒弃了 C++ 中多变的类型系统。

2. 支持跨平台

由于网络系统错综复杂,使用的硬件设备和软件系统各不相同,开发人员所设计的应用程序必须具有强大的跨平台性,C# 编写的应用程序就具有强大的跨平台性,这种跨平台性也包括了 C# 程序的客户端可以运行在不同类型的客户端上,比如 PDA、手机等非 PC 设备。

3. 面向对象且避免了多重继承

C# 支持所有关键的面向对象的概念,如封装、继承和多态性。完整的 C# 类模式构建

在 NGWS 运行时的虚拟对象系统（VOS, Virtual Object System）的上层。对象模式只是基础的一部分。

在 C#中,不存在全局函数、变量或者是常量。所有的东西都封装在类中,包括事例成员（通过类的事例——对象可以访问）或静态成员（通过数据类型）。这些使 C#代码更加易读且有助于减少潜在的命名冲突。

定义类中的方法默认是非虚拟的（它们不能被派生类改写）。主要特点是,这样会消除由于偶尔改写方法而导致另外一些原码出错。要改写方法,必须具有显式的虚拟标志。这种行为不但缩减了虚拟函数表,而且还确保正确版本的控制。

使用 C++编写类,用户可以使用访问权限给类成员设置不同的访问等级。C#同样支持 `private`, `protected` 和 `public` 三种访问权限,而且还增加了第四种: `internal`。

大多数情况,仅需从一个类派生出。这就是为什么 C#仅允许一个基类。如果程序员觉得需要多重继承,可以运用接口。

4. 现代快速应用开发（RAD）功能

支持快速应用开发（Rapid Application Development）是目前开发语言最为重要的功能之一,也正是 C/C++的致命伤。网络时代应用系统的开发必须按照网络时代的速度来进行,支持快速开发可以使得开发人员的开发效率倍增,从而使得他们可以从繁重的重复性劳动中解放出来。C#的 RAD 功能主要表现在如垃圾收集、委托等众多特性上。垃圾收集机制将减轻开发人员对内存的管理负担,而委托功能更是可以让程序员不经过内部类就调用函数。利用 C#的这些功能,可以使开发者通过较少的代码来实现更强大的应用程序,并且能够更好地避免错误的发生,从而缩短了应用系统的开发周期。许多用 C++很费力实现的功能,在 C#中不过是一部分的基本功能而已。

5. 语言的兼容、协作交互性

用 C#编写的程序能最大程度地实现与任何 .NET 的语言互相交换信息,为开发人员节省了大量的时间。C#与其他 .NET 语言有着最好的协作,这对开发人员非常重要。.NET 让各种语言可以真正地互相交流,开发者不必把一种语言强行改成另一种语言。全球从事编写软件的人当中,大约有 50%以 Visual Basic 作为基本的编程工具,在跨入 .NET 编程时代的时候,这些人能轻松地使用 Visual Basic.NET 开发 Web 应用程序,C#可与之很好地兼容、协作交互。

6. 与 XML 的天然融合

由于 XML 技术真正融入到了 .NET 和 C#之中,C#编程变成了真正意义上的网络编程,甚至可以说 .NET 和 C#是专为 XML 而设计的。使用 C#的程序员可以轻松用 C#内含的类使用 XML 技术。和其他编程语言相比,C#为程序员提供了更多的自由和更好的性能来使用 XML。

7. 对 C++的继承且类型安全

C#集成并保留了 C++强大的功能。例如,C#保留了类型安全检测和重载功能,还提供了一些新功能取代原来 C++中的预处理程序的部分功能,提高了语言的类型安全性。

类型安全可以选指针作为一个例子。在 C++ 中使用指针, 程序员能自由地把它强制转换为任何类型, 包括可以执行把一个 `int*` (整型指针) 强制转换成一个 `double*` (双精度指针) 这样的不安全操作。但只要内存支持这种操作, 它就可以执行, 这并不是程序员所想象的企业级编程语言的类型安全。

C# 实施最严格的类型安全, 以保护自己及垃圾收集器。因此必须遵守 C# 中一些相关变量的规则。

C# 取消了不安全的类型转换。不能把一个整型强制转换成一个引用类型 (如对象), 而当向下转换时, C# 验证这种转换是正确的, 也就是说, 派生类真的是从向下转换的那个类派生出来的。

8. 版本可控

在过去的几年中, 几乎所有的程序员都避免涉及众所周知的 “DLL 地狱”, 该问题起因于多个应用程序都安装了相同 DLL 名字的不同版本。有时, 老版本的应用程序可以很好地和新版本的 DLL 一起工作, 但是更多的时候它们会中断运行。

NGWS 运行时将对程序员所写的应用程序提供版本支持。C# 可以最好地支持版本控制。尽管 C# 不能确保正确的版本控制, 但是它可以为程序员保证版本控制成为可能。有了这种支持, 一个开发人员就可以确保当他的类库升级时, 仍保留着对已存在的客户应用程序的二进制兼容。

综上所述, 可以认为 C# 是派生于 C 语言和 C++ 语言的一种程序设计语言。它使程序员能够更快速、更容易地为微软 .NET 平台开发应用程序。但 C# 也有一些弱点, 例如: C# 程序设计和编译程序级的优化不能在非微软的平台上充分利用, 想在非 Windows 平台上展开 .NET, 再充分运用它们也是不现实的。

1.1.3 VS2012 等的新功能

包含 C# 语言的 Visual Studio.NET 已到 VS2012 (或曰 VS11 和 .NET 4.5) 版本了。

C# 1.0 于 2000 年亮相, 随着 C# 2.0 和 Visual Studio 2005 的问世, C# 等语言中新增了几个重要的功能, 其中包括泛型、迭代器和匿名方法等。随同 Microsoft Visual Studio 2008 发布的 C# 3.0 添加了更多功能, 例如扩展方法、lambda 表达式、自动属性, 对象初始化和集合初始化器以及最有名的语言集成查询 (Language Integrated Query, LINQ) 工具。C# 4.0 (C#2010) 提供了进一步的增强, 它改善了与其他语言 and 技术的互操作性。新增的功能包括命名和可选参数; `dynamic` 类型, 它告诉语言的 “运行时” 要实现一个对象的晚期绑定, 以及协变性和逆变性, 它们解决了泛型接口的定义方式所造成的一些问题。C# 4.0 利用了新版本的 .NET Framework, 版本也是 4.0。在这个版本中, .NET Framework 添加了许多新东西, 但最重要的就是构成 “任务并行库” (Task Parallel Library, TPL) 的类和类型。现在可以使用 TPL 构建具有良好伸缩性的应用程序, 从而快速和简单地利用多核处理器的强大能力。对 Web 服务和 Windows Communication Foundation (WCF) 的支持也得到了扩展; 现在可以遵循 REST 模型和较传统的 SOAP 方案构建服务。Microsoft Visual Studio.NET 2010 提供的开发环境使得这些强大的功能变得易于使用, Visual Studio 2010

新增的大量向导和增强措施也显著提高了开发人员的工作效率。

下面介绍 Visual C# 2010 中新的以及加强的主要功能与特性。这些新特性与新功能可有效改善 Microsoft Office 等的编程能力和对代码的分析、研究能力，并对测试驱动开发提供支持。

1. Visual C# 2010 中新的以及加强的主要功能与特性

(1) 动态支持 (Dynamic Support)。通过 Dynamic 这一新类型，Visual C# 2010 对后期绑定到动态类型提供了支持。这一附加功能带来了许多新应用，包括简化访问 COM API 如 Office Automation API，动态 API 如 IronPython，以及 HTML DOM。更多详情，请参考 Using Type dynamic (C# Programming Guide)，dynamic (C# Reference)。

(2) Office 编程能力。通过额外的命名参数和可选参数、动态类型、索引属性和可选 ref 修改器，访问 COM 接口，包括 Office Automation API 的能力被大幅加强。更多详情，请见 Named and Optional Arguments (C# Programming Guide)，Using Type dynamic (C# Programming Guide)，How to: Use Indexed Properties in COM Interop Programming (C# Programming Guide)，How to: Access Office Interop Objects by Using Visual C# 2010 Features (C# Programming Guide)。

(3) 类型等价支持 (Type Equivalence Support)。相比从主互操作程序集 (Primary Interop Assembly) 中导入类型信息，现在可以部署一款带有内嵌类型信息的应用。通过内嵌的类型信息，应用可在运行时间内使用类型而无须参考运行程序集。如果运行程序集发布了多种版本，那么包含内嵌类型信息的应用可以工作在多版本下而无须重新编译。更多详情，请见/link (C# Compiler Options)。示例请见 Walkthrough: Embedding Types from Managed Assemblies (C# and Visual Basic)。

(4) 协变与逆变 (Covariance and Contravariance)。相比泛型参数，协变能使用更具导出性的类型。而逆变则能使用更少导出性的类型。这就允许隐式变换定义类关键字来执行不同接口并通过不同的委托类型提供更灵活的运算方式。不同接口和委托可通过使用新的 in 和 out 语言关键字被创建。.NET Framework 也可对某些现有泛型接口和委托带来不同的支持，包括 IEnumerable<(Of<(T>))> 接口和 Func<(Of<(TResult>))> 与 Action<(Of<(T>))> 委托。更多详情，请见 Covariance and Contravariance (C# and Visual Basic)。

(5) 新的命令行选项：

/langversion 命令行选项可让编译器接受只在特定 C# 版本中有效的语法。

/appconfig 编译器选项可让 C# 应用指定程序集的应用配置文件位置。

Visual C# 集成了开发环境。以下部分对 Visual Studio 集成开发环境 (IDE) 的加强进行了描述。

(6) 调用层次结构 (Call Hierarchy)。调用层次结构能通过你的代码进行导航并显示以下信息：

- 到所选方式、属性和构造函数的所有调用。
- 接口成员的所有执行。
- 虚方法和抽象方法的所有覆盖。

这能让你更好地理解代码流动并评估代码变更的影响。更多详情，请见 Call Hierarchy。