



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2版)

刘甫迎 刘光会 王蓉 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

普通高等教育国家“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2版)

刘甫迎 刘光会 王 蓉 编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

C#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,其结合 ASP.NET 平台开发应用程序设计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。

本书共 12 章和两个附录,详述了 C#的由来和发展, C#的数据类型、运算符与表达式,结构化程序设计及算法,数组、结构和枚举,可视化应用程序设计, C#的文件操作, C#的数据库操作, C/S 模式编程, Web Services 等,并加强了面向对象程序设计、继承和接口的内容。最后给出了一个完整的编程案例。

本书可作为高等学校及软件学院的教材,也适合于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,刘光会,王蓉编著. —2 版. —北京:电子工业出版社,2008.7

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 978-7-121-06490-6

I. C… II. ①刘… ②刘… ③王… 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 095541 号

策划编辑:吕 迈

责任编辑:吕 迈

印 刷:北京市顺义兴华印刷厂

装 订:三河市双峰印刷装订有限公司

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:23.75 字数:608 千字

印 次:2008 年 7 月第 1 次印刷

定 价:34.00 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 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、跨语言的代码。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比较偏重于编程语言理论的教学,教学模式大多是传统的模式,导致学生的编程设计和应用能力不够。笔者认为学习此课程主要应加强实践,提高动手、分析、解决问题的能力。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,其第 2 版的主要特点如下。

(1) 全书以 C#2005 版本为基础,并介绍了 C#2008 的新特点。

(2) 增加了重载、多态性和命名空间,加强了面向对象程序设计、继承和接口的内容(继承和接口独立写了一章,见第 7 章)。

(3) 增加了 SQL 命令。

(4) 加强了 ADO.NET 的内容。

(5) 增加了通用对话框。

(6) 继续把课程教学目标纳入该专业技能培养目标中,目标具体明确,学生学习兴趣大。例如对“计算机软件”等专业来说,C#程序设计课程是该专业的一门专业基础课,本书重点对本专业分解到该课程的结构化程序设计、面向对象程序设计、C/S 模式编程、可视化编程、.NET 平台及 Web Service 等能力模块进行教学。

(7) 在选材上,继续强调讲述的理论“以必需、够用为度”,减轻学生负担,且深入浅出。本书打破了不少教材将 .NET 平台放到书的前面讲授的惯例,将其与 Web Service 的内容放到书最后,甚至可放到专业实习或毕业设计阶段学习,便于学生理解。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) 继续强调实践和实用,突出案例。各章均有大量例子并在第 12 章给出一个综合实例(使用了 UML、CASE、MVC 等技术)。

本书由刘甫迎教授、刘光会、王蓉老师编著。刘甫迎编写第 1 章、第 8 章、第 11 章、第 12 章;刘光会编写第 2 章、第 3 章、第 4 章、第 5 章、第 6 章和附录;王蓉编写第 7 章、第 9 章、第 10 章。全书由刘甫迎统稿。在编写和出版的过程中,对电子工业出版社的编辑给予的帮助表示感谢。

由于水平有限,错误难免,请斧正。

(本书作者网址: www.LifoxSoft.cec.edu.cn)

刘甫迎

2008 年 3 月

目 录

第 1 章 C#概述	1
1.1 C#的由来及发展	1
1.1.1 C#的发展	1
1.1.2 C#的特点	2
1.1.3 C# 2008 的新功能	4
1.1.4 C#的运行环境	7
1.1.5 C#的安装	7
1.2 C#集成开发环境	10
1.2.1 C#的启动	10
1.2.2 C#集成开发环境	12
习题	17
第 2 章 简单的 C#程序设计	18
2.1 C#程序结构	18
2.1.1 第一个 C#程序	18
2.1.2 编译和执行程序	19
2.1.3 C#程序结构分析	20
2.1.4 标识符	21
2.2 输入/输出操作	23
2.2.1 Console.WriteLine()方法	23
2.2.2 Console.Write()方法	26
2.2.3 Console.ReadLine()方法	27
2.2.4 Console.Read()方法	27
习题	28
第 3 章 数据类型、运算符与表达式	29
3.1 数据类型	29
3.1.1 值类型	29
3.1.2 引用类型	32
3.1.3 类型转换	34
3.1.4 装箱和拆箱	36
3.2 常量与变量	37
3.2.1 常量	38
3.2.2 变量	38
3.3 运算符和表达式	40
3.3.1 运算符	40
3.3.2 表达式	48

习题	50
第4章 结构化程序设计	52
4.1 结构化程序设计的概念	52
4.1.1 结构化程序设计的概念及算法的概念	52
4.1.2 流程图	53
4.2 顺序结构	53
4.2.1 顺序结构的概念	54
4.2.2 顺序结构的实例	54
4.3 选择结构	55
4.3.1 选择结构的概念	55
4.3.2 条件语句	55
4.3.3 分支语句	61
4.3.4 选择结构的实例	64
4.4 循环结构	65
4.4.1 循环结构的概念	65
4.4.2 while 语句和 do-while 语句	66
4.4.3 for 语句和 foreach 语句	69
4.4.4 跳转语句	72
4.4.5 循环语句的算法——循环结构的实例	75
4.5 异常处理	78
4.5.1 异常处理的概念	78
4.5.2 异常的处理	79
习题	83
第5章 数组、结构和枚举	86
5.1 数组	86
5.1.1 数组的概念	86
5.1.2 一维数组	86
5.1.3 二维数组	91
5.1.4 “冒泡排序”算法——数组的实例	95
5.2 结构类型	96
5.2.1 结构的声明	96
5.2.2 结构成员的访问	97
5.2.3 顺序查找算法——结构的实例	98
5.3 枚举	100
5.3.1 枚举类型的定义	100
5.3.2 枚举成员的赋值	101
5.3.3 枚举成员的访问	103
习题	106
第6章 C#的面向对象程序设计	109
6.1 面向对象程序设计的基本概念	109

6.2	类和对象	112
6.2.1	类的声明	112
6.2.2	类的成员	113
6.2.3	对象的声明	116
6.3	构造函数和析构函数	118
6.3.1	构造函数	118
6.3.2	析构函数	121
6.4	方法	122
6.4.1	方法的定义及调用	123
6.4.2	方法的参数类型	125
6.4.3	方法的重载	127
6.4.4	静态方法与非静态方法	128
6.4.5	运算符的重载	129
6.4.6	递归	132
6.5	属性	133
6.5.1	属性的声明	134
6.5.2	属性的访问	135
6.6	索引指示器	137
6.6.1	定义索引指示器	137
6.6.2	使用索引指示器访问对象	138
6.7	委托与事件	139
6.7.1	委托	139
6.7.2	事件	142
6.8	C#常用的基础类	144
6.8.1	Math 类与 Random 类	144
6.8.2	字符串	146
6.9	命名空间	149
6.9.1	命名空间的声明	149
6.9.2	命名空间的成员	150
6.9.3	命名空间的使用	150
	习题	152
第 7 章	继承和接口	154
7.1	C#的继承机制	154
7.1.1	继承的基本知识	154
7.1.2	base 关键字	156
7.1.3	覆盖	157
7.2	多态性	159
7.2.1	多态性概述	159
7.2.2	虚方法	159
7.2.3	多态性的实现	160

7.3	接口	161
7.3.1	接口的定义	162
7.3.2	接口的成员	163
7.3.3	接口的实现	167
	习题	185
第8章	可视化应用程序设计	186
8.1	Windows 窗体	186
8.1.1	Windows 窗体的基本属性	186
8.1.2	创建窗体	188
8.1.3	设置启动窗体属性	189
8.2	控件概述	189
8.2.1	控件的基本属性	190
8.2.2	控件共有的属性、事件和方法	191
8.3	命令按钮控件 (Button)	199
8.4	标签控件 (Label)	201
8.5	文本框控件 (TextBox)	203
8.6	单选按钮 (RadioButton) 和复选框 (CheckBox 和 CheckedListBox)	204
8.7	面板控件 (Panel) 和分组框 (GroupBox) 控件	209
8.8	图形框控件 (PictureBox)	212
8.9	列表框 (ListBox)、带复选框的列表框 (CheckedListBox) 和组合框 (ComboBox) 控件	213
8.9.1	列表框控件 (ListBox)	213
8.9.2	带复选框的列表框控件 (CheckedListBox)	215
8.9.3	组合框控件 (ComboBox)	217
8.10	滚动条控件 (ScrollBar)	219
8.11	定时器控件 (Timer)	220
8.12	超链接标签控件 (LinkLabel)	221
8.13	通用对话框控件	223
8.13.1	消息框	223
8.13.2	文件打开和保存对话框	224
8.13.3	颜色选择对话框	226
8.13.4	字体对话框	227
8.13.5	打印机对话框	229
8.13.6	对话框综合实例	231
	习题	237
第9章	C#的文件操作	239
9.1	文件概念	239
9.2	文件的读和写	244
9.2.1	读文件	244
9.2.2	写文件	245
9.3	文件操作实例——链表算法	246

习题	251
第 10 章 C#的数据库操作	252
10.1 Access 数据库	252
10.1.1 建立 Access 数据库表	252
10.1.2 Access 数据库操作	256
10.1.3 SQL 语言简介及常用的 SQL 命令	260
10.2 客户机/服务器 (C/S) 模式编程	263
10.3 ADO.NET 概念	264
10.4 ADO.NET 对象	266
10.4.1 Connection 对象	266
10.4.2 Command 对象	267
10.4.3 DataReader 对象	269
10.4.4 DataAdapter (数据适配器) 对象	269
10.4.5 DataSet 对象	270
10.4.6 数据绑定控件	272
10.5 应用举例	273
习题	289
第 11 章 .NET 平台及 Web 服务	290
11.1 什么是 Microsoft.NET	290
11.2 Web Service 的概念	292
11.2.1 如何编写 Web Service	293
11.2.2 Web Service 测试	294
11.2.3 从客户端进行访问	295
11.2.4 Web Service 中的数据类型	296
11.3 Web Service 编程实例	297
11.3.1 返回 DataSet 的 Web Service 实例	297
11.3.2 在客户机上部署 Web Service	299
习题	304
第 12 章 C#案例——人事管理系统	305
12.1 系统概述	305
12.1.1 办公自动化系统 FoxOA	305
12.1.2 系统功能	306
12.1.3 系统的软件开发平台	307
12.2 系统分析与设计 (UML)	307
12.2.1 软件工程的发展	308
12.3 主要关键技术	329
12.3.1 MVC 设计模式简介	329
12.3.2 MVC 设计模式的实现	330
12.3.3 MVC 设计模式的扩展	332
12.3.4 MVC 设计模式的优点及不足之处	332

12.4	数据库结构.....	333
12.5	用户使用说明书.....	336
12.5.1	系统平台	336
12.5.2	登录和进入系统	337
12.5.3	人力资源部分使用说明	338
12.5.4	系统管理部分使用说明	346
12.6	关键源程序.....	351
	习题	359
附录 A	《C#程序设计课程》教学大纲.....	360
附录 B	《C#程序设计课程》实验指导书	363
参考文献		367

第 1 章 C#概述

本章介绍了 C# 的由来和发展、特点、C# 2008 的新功能以及 C# 的运行环境，让读者能够初步了解 C#，并能够进行 C# 的安装启动和熟悉 C# 的界面。

1.1 C#的由来及发展

1.1.1 C#的发展

美国的微软公司已推出 Visual C#（简称 VC# 或 C#，本书统一称 C#）2008 了，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，也是微软今后发展的战略核心。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（本书第 11 章将详细介绍 .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++ 就消失了，取而代之的就是 C# 语言。

美国的微软公司在 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# 进行编程必将成为今后程序设计的趋势，而

逐步取代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。所有的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 “=” 被用做赋值操作，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发生，从而缩短了应用系统的开发周。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，这点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。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成任何类型，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 C# 2008 的新功能

Visual Studio 2008 和.NET 3.5 发布了。.NET VS 2008 中新的 Visual Basic 和 C#编译器对这些语言做了显著的改进。两者都添加了对函数式编程概念的支持，允许编写更干净，更简洁，更具有表达性的代码。这些特性还促成了称之为 LINQ（语言级集成查询）的新编程模型，使得查询和操作数据成为.NET 中的一等编程概念。C# 2008 的新功能主要有：

- 自动属性，对象初始化和集合初始化器。
- 扩展方法。
- Lambda 表达式。
- 查询句法。
- 匿名类型。

【例 1-1】 C#语法新特型示例。

//说明:C#语法新特型示例

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace C3
{
    class Program
    {
        //新特性 1:简单属性，无须另外多写一个私有字段，比较适合于定义实体类
        public class GsunisMan
        {
            //传统写法，属性必须有一个私有字段
            private int age;
            public int Age
            {
                get { return age; }
                set { age = value; }
            }
        }
    }
}
```

```

    {
        get
        {
            return age;
        }
        set
        {
            age = value;
        }
    }

```

//3.5 写法,直接定义即可;

```

    public string Name { get; set; }
    public Int64 ID { get; private set; }
}

```

```

static void Main(string[] args)
{

```

//新特性 2:隐式类型,隐式类型化本地变量,但必须赋初值,编译器通过初值推导出变量类型

```
var i = 10; //相当于 int i=10;
```

```
var s = "成都小狐狸软件公司(lifox@mail.ccc.edu.cn)"; //s 为 String 类型;
```

```
var arrNums = new int[] { 3,6,9 };
```

//新特性 3:初始化数组的简写方式:new[],可以回忆一下 1.1、2.0 怎么定义。

```
int[] arrNums2 = new[] { 1, 2, 43 };//当然也可以写成:var arrNums2 = new[] { 1, 2, 43 };
```

///新特性 4:匿名类型,在 2.0 中有一个匿名方法,很容易混淆

```
var book1 = new { BookName = "《成都小狐狸软件 2008 纪念版》", BookNumber = 150,
    BookPublisher = "成都小狐狸软件公司" };

```

```

MessageBox.Show(string.Format("书名: {0}, 数量: {1}, 出版: {2}", book1.BookName,
    book1.BookNumber, book1.BookPublisher));

```

///新特性 5:对象构造者特性(实例化时,构造函数后面跟大括号,可直接初始化对象属性),在 1.1、2.0 中必须先定义,然后初始化

```
GsunisMan aMan = new GsunisMan() { Name = "李朝云", Age = 100 };
```

```

MessageBox.Show(string.Format("编号: {0}, 姓名: {1}, 年龄: {2}", aMan.ID,
    aMan.Name, aMan.Age));

```

///新特性 6:集合构造者特性。允许 List<>等自定义集合像数组一样用大括号赋初值,在以前只有数组才可以的

```
List<int> listNums = new List<int>() { 1, 3, 54, 6, 78 };
MessageBox.Show(listNums.Count.ToString());
```

//新特性 7.2,请首先看看下面的新特性 7.1;

```
string str = "123";
bool isNum = str.IsAllNumber();
MessageBox.Show(isNum.ToString());
```

//新特性 8:Linq,作用就不多说了,用法太多,自己找资料看;

//从一个集合中找出所有年龄大于 55 岁的人;

```
List<GsunisMan> men=new List<GsunisMan>();
men.Add(new GsunisMan { Name="刘氏",Age=65});
men.Add(new GsunisMan { Name = "周人", Age =60 });
men.Add(new GsunisMan { Name = "张珊", Age = 50 });
var man= from m in men where m.Age>55 select m; //使用 Linq,前面几行在准备数据,感觉
有些像 Sql 语句,呵呵
MessageBox.Show(man.First<GsunisMan>().Name); //在此仅显示第一个人姓名
```

//新特性 9:Lambda 表达式"=>", 一个简单的例子,用法太多,自己找资料看;

```
int[] numbers = { 5, 4, 1, 3, 9, 8, 6, 7, 2, 0 };
int oddNumbers = numbers.Count(n => n % 2 == 1); //找出奇数的个数
MessageBox.Show(oddNumbers.ToString());
```

```
static public class ExpandClass
```

```
{
//新特性 7.1:扩展方法。可在类的外部扩展已有的类,如下是扩展 string 类,为 String 类增加
一个 IsAllNumber 方法
```

!!!!此特性在.NET 2.0 下不可用!!!!

```
public static bool IsAllNumber(this string str)
{
foreach (var ch in str)
{
if (!char.IsNumber(ch))
{
return false;
}
}
return true;
}
```

```
    }  
    }  
    }  
}
```

1.1.4 C#的运行环境

C#是 Visual Studio.NET 的一部分。作为一个强大的集成开发工具，Visual Studio.NET 对系统环境有较高的要求。因此，在安装 C#之前要全面确定所使用计算机的软、硬件配置情况，看看是否能达到基本配置的要求，以便正确地安装并全面地使用其强大的功能。

1. 硬件要求

- 中央处理器 (CPU): 最低要求 600 兆赫 (MHz) Pentium 处理器，建议采用 1 千兆赫 (GHz) Pentium 处理器。
- 内存 (RAM): 最低配置为 192 兆字节 (MB)。当系统运行时 Visual Studio.NET 的 IDE 占用 90MB 内存空间，并且操作系统还要占用一定的内存空间。如果计算机系统的内存是 128MB，则内存空间就所剩无几了。因此，推荐配置为 512MB 内存。
- 硬盘: 不安装 MSDN: 安装驱动器上要有 2 GB 可用空间，系统驱动器上要有 1 GB 可用空间。安装 MSDN: 在完全安装 MSDN 的安装驱动器上要有 3.8 GB 的可用空间，在进行默认 MSDN 安装的安装驱动器上要有 2.8 GB 的可用空间。系统驱动器上要有 1 GB 可用空间。
- 显示器: 最低配置 800×600、256 色分辨率的显示器。推荐配置 1024×768 全彩分辨率以上的显示器，因高配置显示器可方便地显示开发环境提供的多个不同功能窗口等。

2. 软件要求

- 操作系统: Windows 2000 或 Windows XP 等。
- 后台数据库: Access 2000 或 SQL Server 2000 以上版本。

注意: 中文版的 Visual Studio.NET 不再支持在 Windows Me 和 Windows 95/98 等系统上安装。

1.1.5 C#的安装

C#是 Visual Studio.NET 的一部分，同其他的.NET 语言一样，都必须在.NET 框架环境下运行。因此，要建立一个完整的 C#开发平台，必须安装 Visual Studio.NET 和.NET Framework SDK (.NET 框架软件开发工具包)。

系统中无论安装的是 Windows 2000 还是 Windows XP，只要硬件配置满足 C#的要求，都可以安装 C#。Visual Studio.NET 2003 安装过程大致分为两部分，首先是按照.NET 框架的要求对 Windows 组件进行更新，然后再进行 Visual Studio.NET 和 C#的安装。

Visual Studio 2005 安装程序和以前的版本不同，安装步骤是：