



普通高等学校计算机科学与技术应用型规划教材

C#程序设计语言

C # CHENGXU
SHEJI YUYAN

杜松江 卢东方 张 佳 编著
李华贵 主审



北京邮电大学出版社
www.buptpress.com

普通高等学校计算机科学与技术应用型规划教材

C# 程序设计语言

杜松江 卢东方 张 佳 编 著
李华贵 主 审

北京邮电大学出版社
· 北 京 ·

内 容 简 介

本书的编写目的在于培养学生 C# 语言编程的基本能力,主要包括 Visual Studio 2008 开发环境的介绍、C# 语言基础、程序控制语句、类和类成员的设计、继承与多态、异常处理、字符串、数组与集合、泛型、委托与事件、文件和流等。

本书注重 C# 语言基本语法的讲解,内容深入浅出,通俗易懂,可操作性强。本书的全部代码均运行通过,代码可从北京邮电大学出版社网站下载。本书可作为高等院校、独立院校及高职高专计算机、信息类专业和其他相关专业的教材,也可供广大程序设计爱好者参考。

图书在版编目(CIP)数据

C# 程序设计语言/杜松江,卢东方,张佳编著.--北京:北京邮电大学出版社,2011.4

ISBN 978-7-5635-2570-6

I. ①C… II. ①杜…②卢…③张… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 045102 号

书 名: C# 程序设计语言

作 者: 杜松江 卢东方 张 佳

责任编辑: 王丹丹 田雨佳

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京源海印刷有限责任公司

开 本: 787 mm×1 092 mm 1/16

印 张: 18

字 数: 444 千字

印 数: 1—3 000 册

版 次: 2011 年 4 月第 1 版 2011 年 4 月第 1 次印刷

ISBN 978-7-5635-2570-6

定 价: 32.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

.NET 是微软公司推出的一个非常优秀的平台,近年来已经迅速普及。在该平台上,可以开发出各种安全、可靠的应用程序:控制台应用程序、Windows 应用程序、ASP.NET 网站应用程序、Windows 服务、Web 服务、智能设备应用程序等。.NET 平台上支持的语言较多,有 C#、VB.NET、J# 等。C# 语言自 C/C++ 演变而来,它具有简洁、功能强大、类型安全、完全面向对象等优点,使其成为微软 .NET 战略中一个极其重要的组成部分。C# 语言通常被作为初学者进入 .NET 平台的一种语言,随着 .NET 技术的普及,C# 语言已成为开发基于 .NET 的企业级应用程序的首选语言。编写本书的目的就是为了能够让读者认识 .NET 平台并掌握 C# 语言。

本书通过简明扼要、深入浅出的语言,并结合大量的实例,讲解了 C# 语言的相关知识及各种热点技术,内容主要包括 Visual Studio 2008 开发环境的介绍、C# 语言基础、程序控制语句、类和类成员的设计、继承与多态、异常处理、字符串、数组与集合、泛型、委托与事件、文件和流等。通过学习本书内容,读者能够比较全面地了解 and 掌握 C# 语言语法的相关知识,从而为下一步学习基于 C# 的 Windows 应用程序设计和 Web 应用程序设计奠定坚实的基础。

本书内容由浅入深、循序渐进、实例丰富、可操作性强,既可作为高等院校、独立院校及高职高专计算机、信息类专业和其他相关专业的 C# 程序设计语言的教材,也可供广大程序设计爱好者参考。本书的主要特点如下。

1. 内容全面。本书力求内容系统完整、重点突出,尤其注重培养学生程序设计的思想和实际应用的能力。

2. 实用性强。理论讲述以实例为主,通过实例说明理论。在阐述 C# 语言基本知识的基础上,以控制台应用程序的设计为主,用图书馆管理等相关实例为主线,阐述 C# 程序设计的方法与技巧。

3. 可操作性强。每章均有一定量的习题,帮助读者掌握和巩固所学的知识。每个例子配有完整的代码,便于实践环节的展开。

4. 本书提供所有代码源程序下载,并提供电子教案供教师参考,读者可从北京邮电大学出版社网站免费下载。

本书第 1、2、3、6、7 章由杜松江编写,第 4、5 章由张佳编写,第 8、9、10、11 章由卢东方编写,李华贵教授任主审。全书由杜松江负责统稿。本书编写过程中,参考了大量权威书籍和资料,同时也融入了编者多年在教学和实际项目开发中积累的大量经验。本书在编写过程中得到了各方面的大力支持,在此一并表示感谢。

鉴于编者水平有限,时间仓促,不足与错误之处在所难免,敬请读者指正。

编 者

目 录

第 1 章 C# 程序设计初步	1
1.1 .NET 平台与 .NET 框架简介	1
1.1.1 .NET 平台	1
1.1.2 .NET 框架	1
1.1.3 编译 .NET 程序	2
1.2 面向对象的理解	3
1.2.1 传统的面向过程	3
1.2.2 什么是面向对象	4
1.2.3 类和对象	5
1.2.4 面向对象的三大特性	5
1.3 C# 语言的特点	6
1.4 Visual Studio 2008	7
1.4.1 安装 Visual Studio 2008	7
1.4.2 介绍 Visual Studio 2008 界面	11
1.5 应用程序开发	12
1.5.1 开发和运行控制台应用程序	13
1.5.2 C# 程序的基本结构	14
1.5.3 控制台输入/输出	15
1.6 获取帮助	15
习题	16
第 2 章 C# 语言基础	17
2.1 关键字、标识符和编码规则	17
2.1.1 关键字	17
2.1.2 标识符	18
2.1.3 编码规则与约定	18
2.2 值类型与引用类型的区别	18
2.3 值类型分类	20

2.3.1 简单类型	20
2.3.2 枚举类型	21
2.3.3 结构类型	22
2.4 变量	25
2.5 常量	26
2.6 运算符与表达式	27
2.6.1 赋值和相等运算符	27
2.6.2 算术运算符	28
2.6.3 自增和自减运算符	29
2.6.4 关系运算符	29
2.6.5 逻辑运算符	30
2.6.6 复合赋值运算符	30
2.6.7 条件运算符	30
2.7 数据类型转换	31
2.7.1 隐式转换	31
2.7.2 显式转换	32
2.8 装箱与拆箱	34
习题	34
第3章 C#控制语句	36
3.1 选择语句	36
3.1.1 if 语句	36
3.1.2 switch 语句	40
3.2 循环语句	41
3.2.1 while 语句	42
3.2.2 do-while 语句	43
3.2.3 for 语句	44
3.2.4 循环的嵌套	45
3.3 跳转语句	46
3.3.1 break 语句	46
3.3.2 continue 语句	47
3.3.3 return 语句	48
3.3.4 goto 语句	48
习题	49
第4章 类与类成员	50
4.1 类和对象	50
4.1.1 类的声明	50

4.1.2 对象	51
4.1.3 类的成员	53
4.2 字段	54
4.2.1 实例字段	54
4.2.2 静态字段	57
4.2.3 只读字段	59
4.3 常量	59
4.4 方法	61
4.4.1 方法声明	61
4.4.2 方法体	62
4.4.3 实例方法与静态方法	67
4.4.4 方法调用	67
4.4.5 参数传递	68
4.4.6 方法重载	77
4.4.7 方法递归	79
4.5 构造函数与析构函数	80
4.5.1 实例构造函数	80
4.5.2 this 关键字	82
4.5.3 默认构造函数	84
4.5.4 静态构造函数	85
4.5.5 析构函数	86
4.6 属性	87
4.6.1 常规属性	87
4.6.2 自动属性	90
4.6.3 只读与只写属性	91
4.7 索引器	92
4.8 命名空间	95
4.8.1 为什么要使用命名空间	95
4.8.2 创建和使用命名空间	95
4.8.3 using 指令	97
4.9 分部类	98
习题	99
第5章 继承与多态	101
5.1 继承	101
5.1.1 类继承	101
5.1.2 访问继承的成员	102
5.1.3 Object 类	104

5.1.4 派生类的构造函数	106
5.2 隐藏与重写	110
5.2.1 隐藏基类的成员	111
5.2.2 访问基类的成员	115
5.2.3 重写基类的成员	116
5.3 引用类型转换	118
5.3.1 派生类与基类	118
5.3.2 is 运算符	119
5.3.3 as 运算符	121
5.4 多态	122
5.4.1 方法绑定	122
5.4.2 多态的实现	127
5.5 抽象类	130
5.5.1 抽象类	130
5.5.2 抽象方法	130
5.6 接口	133
5.6.1 声明接口	134
5.6.2 实现接口	135
5.6.3 实现多个接口	137
5.6.4 接口实现多态	141
5.6.5 显示实现接口	142
5.7 密封类和静态类	144
5.7.1 密封类	144
5.7.2 静态类	145
5.8 访问修饰符进阶	147
5.9 方法进阶	150
5.9.1 扩展方法	150
5.9.2 外部方法	152
习题	153
第6章 异常处理	156
6.1 异常介绍	156
6.2 引发异常	158
6.3 异常的捕捉和处理	159
6.3.1 try 语句结构	159
6.3.2 多个 catch 语句块	161
6.4 整型溢出检查	163
习题	164

第 7 章 字符串	165
7.1 String 类	165
7.1.1 创建字符串	165
7.1.2 String 类的属性	166
7.1.3 String 类的方法	167
7.2 StringBuilder 类	176
7.2.1 创建字符串变量	177
7.2.2 StringBuilder 类的属性	178
7.2.3 StringBuilder 类的方法	178
7.3 正则表达式	179
习题	181
第 8 章 数组与集合	182
8.1 声明和创建数组	182
8.1.1 声明和创建一维数组	182
8.1.2 创建数组	182
8.1.3 数组初始化	184
8.1.4 多维数组	188
8.1.5 交错数组	191
8.2 数组与方法	194
8.2.1 将数组传递给方法	194
8.2.2 参数数组	196
8.2.3 返回数组	198
8.3 Array 类	199
8.3.1 数组的复制	199
8.3.2 数组排序	200
8.3.3 在数组中查找	201
8.4 foreach 语句	203
8.5 常用集合类	205
8.5.1 ArrayList 类	205
8.5.2 Queue 类	207
8.5.3 Stack 类	208
8.5.4 Hashtable 类	209
习题	211
第 9 章 泛型	212
9.1 泛型简介	212

9.2 泛型集合	213
9.3 泛型类和接口	215
9.3.1 泛型类	215
9.3.2 泛型接口	216
9.4 泛型方法	218
习题	223
第 10 章 委托与事件	224
10.1 委托	224
10.1.1 定义和使用委托	224
10.1.2 组合委托	228
10.1.3 匿名方法	228
10.1.4 泛型委托	232
10.2 事件	233
10.2.1 委托的发布和订阅	234
10.2.2 事件的发布和订阅	236
10.2.3 使用 EventHandler 类	241
习题	247
第 11 章 文件和流	248
11.1 文件和流简介	248
11.2 I/O 枚举	249
11.2.1 FileAccess 枚举	249
11.2.2 FileAttributes 枚举	249
11.2.3 FileMode 枚举	250
11.3 文件存储管理	250
11.3.1 Directory 和 DirectoryInfo 类	250
11.3.2 File 和 FileInfo 类	257
11.3.3 字节流	268
11.3.4 Stream 抽象类	268
11.3.5 BufferedStream 类	269
11.3.6 FileStream 类	271
11.3.7 MemoryStream 类	271
习题	274
参考文献	275

第1章 C#程序设计初步

.NET 是一种面向网络、支持各种用户终端的开发平台环境。C# 语言是微软公司推出的 .NET 平台下的一种新型语言,集多种语言的特点与优势,是 .NET 应用程序开发的首选编程语言。

本章介绍 .NET 平台、面向对象的含义、C# 语言的特点、Visual Studio 2008 开发环境及控制台应用程序的开发步骤、如何获取帮助等内容。

1.1 .NET 平台与 .NET 框架简介

1.1.1 .NET 平台

2000 年 6 月 22 日,微软公司正式推出 Microsoft .NET。比尔·盖茨在推出 .NET 时曾经说过:“未来 5 年,我们的目标就是超越今天各自为营的 Web 站点,把 Internet 建成一个可以互相交换组件的地方。”微软已经感觉到只靠销售软件包没有什么前途,计划今后逐步转向在网络上使用“服务”的这种商务模型。这样,首先要解决的就是网络上用来开发并执行所有“服务”的平台,这就是 Microsoft .NET。

目前,随着各种不同终端设备的发展,利用不同语言开发的、基于不同系统平台、相对分散独立的各种信息的交互变得备受关注。.NET 的最终目标就是让用户可以在任何地点、任何时间、通过任何现有的设备来得到自己需要的信息,真正达到网络互联的“3A”: Anywhere、Anytime、Anydevice。

在 .NET 平台上,可以实现使用多种语言开发应用程序,可以简单地重复使用已有的功能模块或数据。开发的应用程序可以运行在不同的设备和操作系统上。

.NET 平台包括 .NET 框架和 .NET 开发工具等组成部分。.NET 框架是整个开发平台的基础,包括公共语言运行库和 .NET 类库。.NET 开发工具包括 Visual Studio .NET 集成开发环境和 .NET 编程语言。

1.1.2 .NET 框架

.NET 框架(.NET Framework)是 .NET 开发平台的基础。.NET 框架提供了一个跨语言的、统一的、面向对象的开发和运行环境。.NET 框架的目的是便于开发人员更容易地建立 Web 应用程序和 Web Service,使得 Internet 上的各个应用程序之间可以使用 Web Service 进行沟通。

.NET 框架的基本结构如图 1-1 所示。在图 1-1 中,操作系统处于整个 .NET 框架的最底层。公共语言运行库(Common Language Run,CLR)是 .NET 框架的基础。公共语言运行库是一个在执行时管理代码的代理,提供核心服务(如内存管理、线程管理和远程处理等),而且还强制实施严格的类型安全以及可确保安全性和可靠性的其他形式的代码准确性。

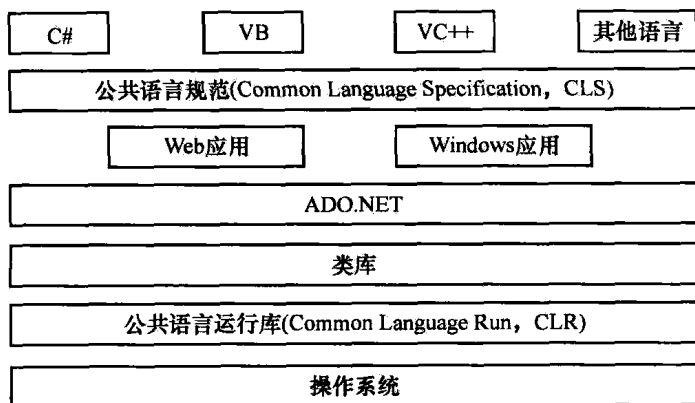


图 1-1 .NET 框架结构

.NET 类库是一个综合性的可重用类型的集合,是使用系统功能及编写 .NET 程序的基础。.NET 类库通过命名空间对所有类及其内容进行管理。

ADO.NET 是基于 .NET Framework 的新一代数据访问技术,ADO.NET 的主要目的是可以在 .NET Framework 中更容易地创建分布式的数据共享的应用程序。ADO.NET 建立在 XML 的基础之上,为数据访问提供了许多工具,为应用程序间交换数据提供了很好的条件。

Web 应用和 Windows 应用属于两种不同的 .NET 平台下应用程序开发模式,即所谓的基于 Windows 的应用程序开发和基于 Web 的应用程序开发。

公共语言规范(Common Language Specification,CLS)是支持 .NET 平台上各种语言间可互操作的一组规则。所有 .NET 语言都应该遵循此规则才能创建与其他语言可互操作的应用程序。

1.1.3 编译.NET 程序

.NET 采用特殊的方式编译和执行各种应用程序。编译时,内置的语言编译器首先将应用程序编译为微软中间语言(Microsoft Intermediate Language,MSIL)。MSIL 由 .NET 框架中的组件 CLR 管理和执行。在进行第二步编译时,.NET 框架采用了一种名为即时编译(Just In Time,JIT)的技术。JIT 将 MSIL 代码转换为可以直接由 CPU 执行的机器代码,整个编译过程如图 1-2 所示。一旦编译成功,在下一次被调用时也无须再次编译。

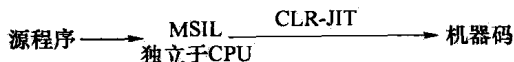


图 1-2 .NET 程序编译过程

需要强调的是,在.NET 框架支持的语言中,各种语言在第一步编译时都被编译成 MSIL 代码。而 MSIL 代码是不存在语言差别的,它是独立于任何一种硬件平台和操作系统的。因此,语言之间可以实现相互调用和代码共享,开发人员可以任意选用自己熟悉的.NET 编程语言。

1.2 面向对象的理解

面向对象(Object Oriented, OO)是一种应用程序设计和开发的思想,已经成为了软件开发方法的主流。面向对象的概念和应用已超越了程序设计和软件开发,扩展到很宽的范围,如数据库系统、交互式界面等领域。

起初,面向对象是专指在程序设计中采用抽象、封装、继承、多态等设计方法。而现在,面向对象的思想已经涉及软件开发的各个方面,如面向对象的分析(Object Oriented Analysis, OOA)、面向对象的设计(Object Oriented Design, OOD),以及我们经常说的面向对象的编程(Object Oriented Programming, OOP)。使用面向对象进行应用程序开发能够很好地对现实中的物体进行抽象,这样就在一定程度上丰富了应用程序的结构。

1.2.1 传统的面向过程

在面向对象编程出现之前,面向过程(结构化编程)的编程方式一直占据主流的位置。所谓面向过程编程,就是分析出解决问题所需要的所有步骤,然后用函数或过程将这些步骤一步一步地实现和调用。面向过程讲究的是自顶向下、逐步细化的编程思想。FORTRAN、Pascal、C 等都是面向过程的语言。

比如求两个整数的和,面向过程的一段 C 语言代码如下:

```
void main()
{
    int s = 0;           //定义一个整型值,用来存放和
    InputElem(x,y);      //由用户输入 2 个整型值
    s = Sum(x,y);        //Sum 方法计算和之后将返回值赋值给 s
    printSum(s);         //将计算结果打印出来
}
```

从上述代码中可以看出,当执行一个主函数 main 时,按照程序逻辑调用不同的函数,来达到运算的目的。用结构化方法开发的软件,尤其是面向大型软件的开发时,其稳定性、可修改性和可重用性都比较差,这是因为结构化方法的本质是功能分解。具体而言,面向过程编程存在以下问题:

(1)面向过程编程最大的一个缺点就是对数据的安全性保护不够。在面向过程编程中,数据与操作数据的函数是分开的。把数据和代码作为分离的实体,反映了计算机的观点,因为在计算机内部数据和程序是分开存放的。但有时数据是全局的,这意味着对数据的访问和操作是不能控制,也是不能预测的,如多个函数或语句访问相同的全局数据,可

以想象这种方式编制出来的软件潜在的危险性有多大。使数据和操作保持一致,是程序员的一个沉重负担,在多人分工合作开发一个大型软件系统的过程中,如果负责设计数据结构的人中途改变了某个数据的结构而又没有及时通知所有人员,则会发生许多不该发生的错误。

(2) 可维护性差。结构化方法是围绕实现处理功能的“过程”来构造系统的。然而,用户需求的变化大部分是针对功能的,因此这种变化对于基于过程的设计来说是灾难性的。用这种方法设计出来的系统结构常常是不稳定的,用户需求的变化往往造成系统结构的较大变化,从而需要花费很大代价才能实现这种变化。

(3) 可重用性差。软件代码的重用性很差,即使重用,也是简单的复制,代码数量急剧增加,而不能直接继承和应用。

1.2.2 什么是面向对象

面向对象把程序中各个功能模块按照分类进行归纳和整理,然后将整理结果制作成一个“类”。面向对象技术是一种以对象为基础,以事件或消息来驱动对象执行程序处理的编程技术,具有抽象、封装、继承及多态等特性。面向对象程序设计方法认为,客观世界是由各种各样的实体组成的,这些实体就是面向对象方法中的对象。Visual Basic、C++和C#等都是面向对象的语言。

与面向过程相比,它具有以下优点:

(1) 与人类习惯的思维方法一致

面向对象的开发方法与传统的面向过程的方法有本质不同,这种方法的基本原理是,使用现实世界的概念抽象地思考问题,从而自然地解决问题。它强调模拟现实世界中的概念而不强调算法,它鼓励开发者在软件开发的绝大部分过程中都用应用领域的概念去思考。在面向对象的开发方法中,计算机的观点是不重要的,现实世界的模型才是最重要的。面向对象的软件开发过程从始至终都围绕着建立问题领域的对象模型来进行:对问题领域进行自然的分解,确定需要使用的对象和类,建立适当的类等级,在对象之间传递消息实现必要的联系,从而按照人们习惯的思维方式建立起问题领域的模型,模拟客观世界。

比如,在图书馆管理系统中,我们可以从现实世界中得到这些实体:图书、读者、图书馆管理员等,那么这些实体就是我们在图书馆管理系统中要构造的对象。这个系统的运行过程就是这些对象进行交互的过程。

(2) 数据和行为分离

面向对象的软件技术以对象为核心。对象是对现实世界实体的正确抽象,它是由描述对象内部状态的数据,以及可以对这些数据施加的操作(行为),封装在一起所构成的统一体。对象内部状态的数据具有访问权限的控制,有些数据是能够公开访问的,有些数据则是限制访问的,这样就能够确保数据的正确性和安全性。

(3) 稳定性好

因为面向对象的软件系统的结构是根据问题领域的模型建立起来的,而不是基于对

系统应完成的功能的分解,所以当对系统的功能需求变化时,并不会引起软件结构的整体变化,往往仅需要作一些局部性的修改。由于现实世界中的实体是相对稳定的,因此以对象为中心构成的软件系统也是比较稳定的。

(4) 可重用性好

重用是提高编程效率的一个重要方法。面向对象的软件技术在利用可重用的软件成分构造新的软件系统时,体现出较大的灵活性。面向对象的软件技术所实现的可重用性是自然和准确的,在软件重用技术中它是最成功的一个。

(5) 可维护性好

面向对象的软件技术符合人们习惯的思维方式,因此用这种方法所建立的软件系统容易被维护人员理解。每个对象都明确自己具有什么职责,软件哪个地方出现了问题,就只用修改相应对象的代码,而不会担心“牵一发而动全身”,提高了软件的可维护性。此外,一个设计良好的面向对象系统是易于扩充和修改的,因此能够适应不断增加的新需求,以上这些都是从长远考虑的软件质量指标。

1.2.3 类和对象

面向对象程序设计从所处理的数据入手,并以数据为中心,把现实世界的问题抽象为“类”的概念。类是对现实世界中一系列具有相同性质的事物的总称,是对具有共同数据和行为的一类事物的抽象描述。其中,共同数据被描述为类中的数据成员,共同行为被描述为类中的方法成员。例如,读者(Reader)是对各种读者(如老师读者、学生读者等)的一个抽象,读者的证号、姓名、年龄等都是读者类的数据成员,而借书和还书等是读者类的方法成员。

面向对象的程序设计方法最基本的思想就是把所有要进行研究的事物,都称为对象。一切皆是对象,任何物理实体、抽象的规则、计划或者事件都可以是对象。例如,一个人、一本书、一张椅子,乃至一首歌曲、一个构想,都可以作为一个对象。对象是类的一个实例。实例化的对象具有唯一性,用以区别同类的其他对象。例如,读者是一个类,具体到某一个读者(如某同学)就是一个对象,并且是唯一的。

1.2.4 面向对象的三大特性

1. 封装

封装是把数据和对数据的操作封藏成一个有机整体,即把字段、属性以及方法、事件等一起封藏在一个公共结构中,创建一种称为类或结构的新“数据类型”。

在面向对象程序设计中,把数据和对数据的操作封装后,就可以为其指定属性和方法来供用户使用。对于外部使用者而言,无须知道对象内部的具体实现细节,而只需要关心如何使用该对象,从而保证了私密的内容不会被用户察觉,不会被外界随意改变,也使对象成了相对独立的功能模块,避免数据被程序直接访问的概念称为“数据隐藏”。

例如,一部手机由屏幕、内部线路、电池等部分组成。在整个“手机”对象里面,手机内部线路对一般的用户是不可见的,因为用户不知道怎样拆装内部线路,也不需要去了解工

作原理,但是用户知道怎样将电池安装上去,怎样拨打电话。封装能够让用户更加关注“手机”本身的使用,而不去关注“手机”内部是怎样实现的。

2. 继承

继承是可以让某个类型对象获得另一个类型对象的共有特性的一种手段,是类的层次结构之间共享数据和方法的一种机制。继承可以描述为一种树状的层次关系。例如,

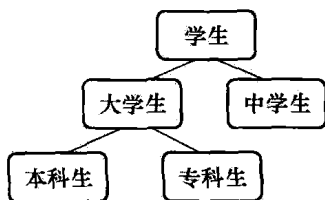


图 1-3 “学生”继承示例图

本科生和专科生都属于大学生,而大学生又属于一种学生类型,如图 1-3 所示。这种分类的原则是,每一个子类都具有父类的公共特性。子类在继承父类成员的同时,也可以定义属于自己的成员内容。

在面向对象中,继承的概念很好地支持了代码的重用性,也就是说,通过继承生成的新类将具有原来那个类的特性,以及它本身一些新的特性。

3. 多态

方法在处理不同对象的时候会得到不同的结果,这个就是类的多态。利用多态,可以在具有继承关系的多个类中,定义名称相同但操作不同的多个方法。在程序运行时,根据对象的实际类型调用相应的方法。例如,所有动物都具有“吃”这个方法,但是在动物的子类中,不同的动物在“吃”的时候,却有各自不同的“吃法”,即不同的实现方法。应用多态,可使程序具有良好的可扩充性。

1.3 C# 语言的特点

.NET 平台支持多种语言,如 C# (读作 C Sharp)、VB.NET、Visual# C++ .NET 等。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++ 的使用者所熟悉。C# 的目的就是综合 Visual Basic 的高生产率和 C++ 的行动力。”C# 主要具有以下特点:

(1) C# 语言拥有 C/C++ 的强大功能以及 Visual Basic 简易使用的特性,而且看起来与 Java 有着惊人的相似。因集众家之长,使其不仅安全,而且易于掌握和使用。

(2) C# 语言不仅有实时的编译器,而且含有比一般语言更丰富的数据类型。无论是从输出格式,还是网络 I/O,都有一整套标准的类和数据类型。

(3) C# 语言以提供脚本语言所无法提供的强大功能,使用 C# 编写 ASP.NET 应用程序将成为 Web 开发的最佳选择。

(4) C# 语言包含的内置特性,使任何组件可以轻松转化为 XML 网络服务,通过 Internet 被任何操作系统上运行的任何程序调用。

(5) C# 语言能够消除很多常见的 C++ 编程错误,如变量由环境自动初始化、变量的类型安全等。

除了这些特点,C#语言也支持.NET平台本身具有的垃圾回收、语言自由、跨平台等特性。

1.4 Visual Studio 2008

Visual Studio 2008 是 Visual Studio 的新版本,它是一套用来开发、调试、部署各种应用程序的产品的最重要的开发环境。Visual Studio 2008 引入了 250 多个新特性,整合了对象、关系型数据、XML 的访问方式,语言更加简洁。

Visual Studio 2008 开发环境可以为使用者完成以下工作:

- 对 AJAX 的支持
- 引入了语言集成查询 LINQ
- 开发 Windows 应用程序
- 开发 Web 应用程序
- 对网页 Java script 脚本进行调试
- 开发 Web Services 应用程序
- 调试 WIN API COM 接口

1.4.1 安装 Visual Studio 2008

安装 Visual Studio 2008,首先要注意计算机的配置要求。

(1) 支持的操作系统

- Microsoft Windows XP
- Microsoft Windows Server 2003
- Windows Vista
- Microsoft Windows 7

(2) 硬件要求

- 最低要求:1.6 GHz CPU、384 MB RAM、1 024×768 显示器、5400 RPM 硬盘。
- 建议配置:2.2 GHz 或速度更快的 CPU、1 024 MB 或更大容量的 RAM、1 280×1 024 显示器、7200 RPM 或更高转速的硬盘。

需要注意的是,对于硬盘,完全安装的空间要求至少 4G。为了完全支持 Web 应用程序的发布,还要注意必须安装 Internet 信息服务 (IIS)。

Visual Studio 2008 安装文件可以在微软的官方网站上下载。下面以在 Windows XP SP3 系统中安装带有 MSDN 的 Visual Studio 2008 中文版为例,介绍整个安装的过程。启动安装文件中的 Setup.exe 执行程序,如图 1-4 所示。选择“安装 Visual Studio 2008”后,开始加载安装组件,如图 1-5 所示。

等待片刻之后,进入安装程序的起始页,选择“我已阅读并接受许可条款”,并输入用户名,如图 1-6 所示,单击“下一步”。