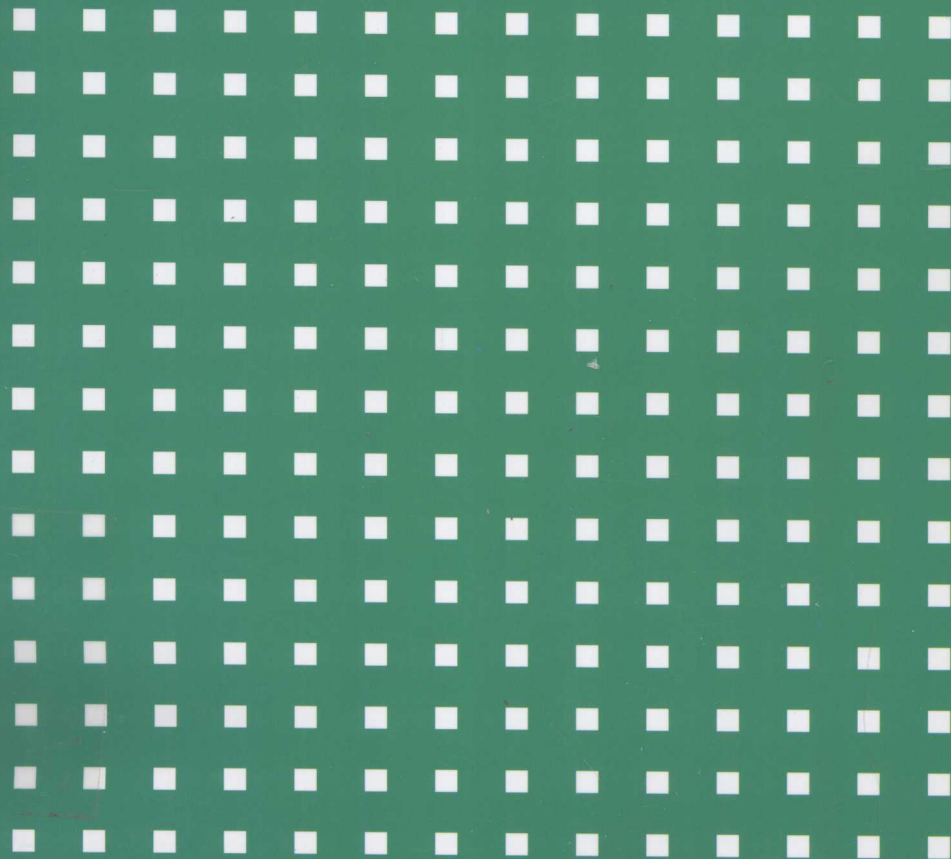


高等学校计算机专业教材精选·算法与程序设计

面向对象程序设计 实例教程

孙连英 刘畅 彭涛 编著



清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

面向对象程序设计 实例教程

孙连英 刘畅 彭涛 编著

清华大学出版社
北京

内 容 简 介

面向对象程序设计实例教程基于 Java 语言介绍面向对象程序设计理念,引入 UML 工具解释设计过程。应用丰富的案例阐述面向对象程序设计的基本理论和方法,详细介绍面向对象的基本特性、基本技术,知识点与应用实例相结合。全书分为三篇:第一篇,程序设计基础,介绍 Java 编程基础;第二篇,Java 面向对象,介绍面向对象的封装性、继承性和多态性;第三篇,高级面向对象设计,介绍 UI、多线程和网络编程等内容。全书的三篇从简单到复杂,阶梯式递进。读者可以根据需要选读。

本书介绍面向对象思想,突出理论联系实际。本书既可作为高等院校软件工程、计算机相关专业的本科学生的教材,也可作为研究生的参考资料。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

面向对象程序设计实例教程/孙连英,刘畅,彭涛编著. —北京:清华大学出版社,2014

高等学校计算机专业教材精选·算法与程序设计

ISBN 978-7-302-36352-1

I. ①面… II. ①孙… ②刘… ③彭… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 099134 号

责任编辑:张 玥 赵晓宁

封面设计:傅瑞学

责任校对:梁 毅

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:19

字 数:472 千字

版 次:2014 年 9 月第 1 版

印 次:2014 年 9 月第 1 次印刷

印 数:1~2000

定 价:34.50 元

产品编号:055849-01

前 言

背景

随着移动互联网和 Web 技术的不断发展,移动互联网逐渐成为人们日常交流、通信和娱乐的基本工具。截止到 2012 年 12 月底,新浪微博注册用户就已超过 5 亿,日活跃用户 4629 万,75% 的日活跃用户来自移动端。人们通过移动终端获取信息成为一种习惯和必要手段。国内外学者们相继对各种基于移动终端软件开发技术进行研究,主流的操作系统主要有 Android、IOS 和 Windows Phone 等,仅 2013 年第二季度 Android 市场占有率就达 70% 以上。Java 移动应用软件开发人员有较大的社会需求,并且可以和 C 语言很好地结合起来。

面向对象程序设计已经成为软件编程技术中一项非常关键的技术。Java 语言是一种完全的面向对象的语言,它吸收了其他语言的各种优点,设计简洁而优美,使用方便而高效。特别是跨平台性,使其在大型应用系统和嵌入式开发中都占有重要地位。本教材基于 Java 语言介绍面向对象的思想设计,运用 UML 建模,具有丰富的实例,体现面向对象程序设计的应用性。

Java 世界的巅峰永远都在技术人员的追求中不断升高,每个人都有不同的路,有效的参考教材是每一位程序员必不可少的工具。本书的编写主要从应用开发的逻辑路径组织内容,注重完成基本功能的思路和步骤分析说明,没有从试图穷尽所有的知识面角度来撰写,如果读者想就某一个具体的技术点(比如某个控件的特殊用法)进行突破需要参阅其他资料来完成。

本书特色

任何技术的目的都是为了应用。本书不仅结合实例详细讲解了 Java 的基础知识,同时也就 Java 的主要应用进行了实例讲解。全书共分 15 章,从基本概念的引入,到典型案例分析,使读者更形象的理解面向对象思想,掌握 Java 编程技术。

本书特点如下:

- (1) 由案例引入,从具体问题的分析入手,由浅入深。
- (2) 注重具体问题的分析、设计。案例中给出解决思路,有助于提高学生的分析问题、解决问题能力。
- (3) 突出软件开发的设计与实现过程,将面向对象分析与 Java 语言开发相结合,使学生掌握软件开发的基本技能。
- (4) 每章后配有实验,注重程序开发能力的锻炼,学生在练习中体会知识。

读者对象

本书可作为高等院校软件工程、计算机科学与技术等相关专业本科教材,也可作为相关学科的研究生参考教材,同时也可作为初学者学习 Java 开发、移动应用开发和 Java EE 开发的培训教材。

本书作者

本书第1、第10、第14和第15章由孙连英与李刚编写,第4~第8、第12和第13章由刘畅编写,第2、第3、第9和第11章由彭涛编写,全书由孙连英统稿。本书在编写过程中,得到了任运贵、张启秀、李琳、刘小安等的帮助,在此特表示感谢。

本书受到北京市教育委员会科技发展计划面上项目(SQKM201411417013)、(KM201211417002)、北京联合大学人才强校计划的资助。

本书实例中的程序源代码,读者可以从清华大学出版社网站上免费下载。

书中难免会有疏漏或不妥之处,敬请读者批评指正。

编者

2014年5月

目 录

第一篇	Java 编程基础	1
第 1 章	概述	1
1.1	面向对象程序设计	1
1.1.1	定义	1
1.1.2	基本概念	1
1.1.3	特点	2
1.1.4	面向对象编程语言	3
1.2	Java 语言概述	3
1.2.1	诞生及发展	3
1.2.2	Java 语言的特点	4
1.2.3	Java 开发平台	7
1.3	Java 开发环境	7
1.3.1	下载、安装 JDK	8
1.3.2	配置环境变量	9
1.4	Java 开发工具	11
1.5	学习创建第一个 Java 程序	12
1.5.1	制作 Java Application 应用程序	12
1.5.2	制作 Java Applet 小程序	16
	习题 1	17
第 2 章	编程基础	19
2.1	标识符	19
2.2	数据类型	20
2.2.1	基本数据类型	20
2.2.2	类型转换	21
2.3	变量和赋值	23
2.4	常量	23
2.5	运算符和表达式	23
2.5.1	基本赋值运算	24
2.5.2	算数运算	24
2.5.3	关系运算	26
2.5.4	布尔运算	27
2.5.5	位运算	27

2.5.6	条件运算	28
2.5.7	复合赋值运算符	28
2.6	字符串	28
2.6.1	输入字符串实例	29
2.6.2	字符串类型与数值型的转换	29
2.7	程序设计	30
	习题 2	30
	编程练习	32
第 3 章	控制语句	33
3.1	if 语句	33
3.1.1	if 语句的三种形式	33
3.1.2	分段显示实例(if)	35
3.2	switch 语句	36
3.2.1	分段显示实例(switch)	36
3.2.2	switch 语句详解	37
3.3	for 循环	37
3.3.1	for 循环语法	37
3.3.2	求和运算实例	38
3.4	while 循环	39
3.4.1	while 循环语句	39
3.4.2	while 循环实现输入控制	40
3.5	do··while 循环	40
3.5.1	do··while 语句	40
3.5.2	do··while 实现退出操作	41
3.6	break 与 continue	41
3.6.1	break 语句	41
3.6.2	continue 语句	42
3.7	综合实例:十进制与二进制转换	42
3.7.1	问题分析	42
3.7.2	算法设计	43
3.7.3	主程序	43
	习题 3	44
	编程练习	45
第 4 章	方法	47
4.1	定义方法	47
4.2	调用方法	47
4.3	方法实例	48

87	4.4	方法应用	51
87	4.4.1	单个参数的方法	51
88	4.4.2	多个参数的方法	52
88	4.4.3	递归方法	53
18	4.4.4	多个返回值的方法	54
18	4.5	方法重载	55
88	4.6	预定义方法	55
88		习题 4	58
18		编程练习	59
68			
	第 5 章	数组	60
88	5.1	创建数组	60
88	5.2	初始化一维数组	61
88	5.2.1	静态初始化	61
88	5.2.2	动态初始化	62
10	5.2.3	创建数组实例	63
88	5.3	数组名的使用	63
88	5.4	数组作为方法的参数和返回值	65
88	5.4.1	数组作为方法参数	65
88	5.4.2	数组作为返回类型	66
70	5.5	增强的 for 循环	67
80	5.6	有用的数组方法	68
101	5.7	多维数组	68
101	5.7.1	二维数组定义	68
101	5.7.2	初始化二维数组	69
101	5.7.3	二维数组实例	70
101	5.8	文本处理实例	71
801	5.8.1	问题分析和方法设计	71
801	5.8.2	主算法设计	73
	5.8.3	程序实现	74
811		习题 5	75
811		编程练习	76
811			
	第二篇	Java 面向对象	77
811			
	第 6 章	类和对象	77
811	6.1	类与对象的关系	77
881	6.2	类的设计与 UML 建模	77
881	6.3	类	78

16	6.3.1	汽车类实例	78
16	6.3.2	定义类	79
32	6.4	对象	80
80	6.4.1	创建对象	80
128	6.4.2	使用对象	81
80	6.5	成员变量与局部变量	81
36	6.6	构造方法	82
82	6.6.1	为什么需要构造方法	82
97	6.6.2	构造方法重载	84
	6.7	this 关键字	85
98	6.7.1	this 表示类的成员	86
98	6.7.2	this 调用构造方法	86
10	6.7.3	this 表示当前对象	88
18	6.8	static 修饰符	88
58	6.8.1	静态变量	89
63	6.8.2	静态方法	91
28	6.8.3	main 方法详解	92
28	6.8.4	静态成员特点	93
60	6.9	使用 String 类	93
86	6.10	对象参数实例	95
73	6.11	聚集类	97
20	6.11.1	聚集类实例	98
20	6.11.2	银行操作主界面	101
80	6.12	内部类	105
90	6.12.1	类中定义的内部类	105
107	6.12.2	内部类被外部引用	107
117	6.12.3	方法中定义的内部类	107
117		习题 6	108
87		编程练习	109
117			
	第 7 章	封装	112
87	7.1	类的抽象与封装	112
	7.2	Java 中的包	113
77	7.2.1	package 语句	113
	7.2.2	import 语句	115
77	7.2.3	Date、Calendar 与 DateFormat 类	116
77	7.2.4	Math 与 Random 类	118
77	7.3	类的成员的访问控制	122
87	7.3.1	包访问权限	122

161	7.3.2	public: 接口访问权限	122
161	7.3.3	private: 类内部访问权限	123
161	7.3.4	protected: 继承访问权限	123
161	7.4	类的访问权限	123
161	7.5	链表实例	123
	7.5.1	链表结点	124
861	7.5.2	链表类	124
861	7.5.3	测试类	128
861	习题 7	129
861	编程练习	130
871		130
	第 8 章 继承	131
161	8.1	继承的定义与实现	131
161	8.1.1	继承实例	131
161	8.1.2	继承的定义	131
161	8.1.3	继承的实现	132
161	8.1.4	继承的结果	135
161	8.2	方法重写	135
161	8.3	super 关键字	138
161	8.3.1	使用 super 访问父类成员	139
161	8.3.2	使用 super 调用父类构造方法	140
	8.3.3	构造子类对象	140
161	8.3.4	super 与 this	141
161	8.4	抽象类	141
161	8.4.1	Employee 抽象类	142
161	8.4.2	抽象类实例	142
161	8.4.3	抽象类的子类实例	143
161	8.5	接口	145
161	8.5.1	定义接口	145
161	8.5.2	接口实例	146
161	8.6	final 修饰符	147
161	8.6.1	final 修饰类	147
161	8.6.2	final 修饰方法	148
161	8.7	Object 类	148
161	8.7.1	toString() 方法	148
161	8.7.2	equals() 方法	149
161	8.8	类的关系	149
161	8.8.1	依赖	150
161	8.8.2	关联	150

8.8.3	聚合	151
8.8.4	组合	151
8.9	综合实例：组装计算机	152
	习题 8	154
	编程练习	154
第 9 章 多态 156		
9.1	多态实例	156
9.1.1	句柄引用对象实例	156
9.1.2	父类句柄引用子类对象	157
9.2	动态绑定	158
9.3	对象的多态性	160
9.3.1	向上转型	161
9.3.2	向下转型	161
9.3.3	instanceof 运算符	162
9.4	多态实例	162
9.4.1	接口作为参数实例	163
9.4.2	父类作为方法返回类型实例	165
9.4.3	面向接口编程	168
	习题 9	169
	编程练习	169
第三篇 高级面向对象设计 171		
第 10 章 图形用户界面设计 171		
10.1	Swing 和 AWT	171
10.1.1	AWT 组件	171
10.1.2	Swing 组件	172
10.1.3	容器类组件	173
10.2	创建基本 GUI 程序	174
10.2.1	使用 JFrame 类创建框架	174
10.2.2	在框架中添加组件	175
10.2.3	设置界面布局	176
10.2.4	事件处理	182
10.3	事件及监听器接口	184
10.3.1	事件	184
10.3.2	Windows 事件处理	185
10.3.3	键盘事件处理	188
10.3.4	鼠标事件处理	191

10.4	菜单	193
10.4.1	菜单的设计与实现	193
10.4.2	实现菜单项事件处理代码	194
习题 10		198
编程练习		199
第 11 章 异常处理		200
11.1	Java 异常处理	200
11.2	Exception 类	200
11.3	使用异常处理	201
11.3.1	try/catch/finally 块	201
11.3.2	catch 块的顺序	203
11.3.3	抛出异常	204
11.4	自定义异常	204
11.5	为异常编写文档	207
习题 11		208
编程练习		209
第 12 章 集合类		210
12.1	Iterable 接口	211
12.2	Collection 接口	211
12.3	List 接口	212
12.3.1	LinkedList 类	213
12.3.2	ArrayList 类	213
12.3.3	Vector 类	214
12.3.4	Stack 类	214
12.4	Set 接口	215
12.4.1	Set 接口常用方法	215
12.4.2	Set 接口实例	215
12.5	Map 接口	217
12.5.1	Map 常用方法	217
12.5.2	HashMap 管理网络名和密码	217
12.6	泛型	219
12.7	集合类实例: 书籍管理	220
习题 12		223
编程练习		223
第 13 章 输入输出流及文件处理		225
13.1	I/O 流	226

891	13.2	字节流	226
891	13.2.1	InputStream 类常用方法	227
891	13.2.2	OutputStream 类常用方法	227
897	13.2.3	文件数据流	228
891	13.2.4	实例: 输入信息保存到文件	228
	13.2.5	读取并显示文件	230
899	13.2.6	文件复制	231
899	13.3	字符流	232
899	13.3.1	InputStreamReader 和 OutputStreamWriter	232
899	13.3.2	字符流实例	233
899	13.4	文件输入输出	234
899	13.4.1	文件基本操作	234
899	13.4.2	文件随机读写	236
899	13.5	对象序列化	237
899	13.5.1	存储对象	237
899	13.5.2	Car 对象序列化实例	238
899	13.5.3	存储和读取序列化对象信息	238
		习题 13	239
899		编程练习	240
	第 14 章	多线程	241
899	14.1	线程概述	241
899	14.1.1	线程的概念	241
899	14.1.2	如何理解多线程	241
899	14.2	创建线程	242
899	14.2.1	通过继承 Thread 类来创建线程	242
899	14.2.2	通过实现 Runnable 接口的类来运行线程	243
899	14.3	线程的状态	244
899	14.3.1	线程的不同状态	244
899	14.3.2	线程的优先级及调度	246
899	14.3.3	线程状态的转换	247
899	14.5	制作动画	249
899	14.5.1	Java 多线程动画制作概述	249
899	14.5.2	动态文字显示	249
899	14.5.3	模拟赛跑	250
899	14.6	共享资源与线程同步	254
899	14.6.1	资源共享	254
899	14.6.2	线程同步	256
899	14.6.3	等待与通知	257

14.6.4 死锁·····	260
习题 14 ·····	260
编程练习·····	262
第 15 章 网络编程 ·····	263
15.1 Java 网络编程基础 ·····	263
15.1.1 IP 地址 ·····	263
15.1.2 端口号·····	263
15.2 InetAddress 类 ·····	264
15.3 使用 URL 类访问网络资源 ·····	265
15.4 Socket 套接字编程 ·····	268
15.4.1 使用 TCP 通信 ·····	268
15.4.2 使用 UDP 通信 ·····	270
15.5 访问远程数据库·····	272
15.6 访问 Web Service ·····	273
15.6.1 生成客户端类·····	273
15.6.2 访问股票行情查询 Web 服务 ·····	276
15.7 访问 Web Server 端应用程序 ·····	277
15.7.1 服务器端开发·····	277
15.7.2 在应用程序中访问·····	283
习题 15 ·····	286
编程练习·····	286
参考文献 ·····	287

第一篇 Java 编程基础

第 1 章 概 述

20 世纪 80 年代以来,面向对象的程序设计方法越来越引人注目。与传统的程序设计方法比较,面向对象的程序设计方法最显著的特点是它更接近于人们通常的思维规律,这也是它变得流行的主要原因。

本章首先讨论面向对象程序设计方法的一些基本概念,然后结合面向对象程序中的“佼佼者”——Java 来进行介绍。

1.1 面向对象程序设计

1.1.1 定义

面向对象(Object Oriented,OO)技术已经被人们谈论几十年了,其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。那么什么是面向对象呢?面向对象是一种新兴的程序设计方法,或者说它是一种新的程序设计范型,其基本思想是使用对象、类、继承、封装和消息等基本概念来进行程序设计。

在面向对象程序设计之前,程序设计语言是面向过程的,程序中数据和处理数据的过程分别存储于不同的地方,数据和过程之间没有逻辑或组织上的联系。这种程序设计方法在编写小型和中型程序时表现得相当好。不过,由于数据和过程分离,对于较大的程序,开发和维护就格外复杂。

吸取了面向过程程序设计的一切优点,同时又改变了程序中数据和处理数据过程分离的状况,人们提出了面向对象的程序设计方法。面向对象程序设计是程序设计方法的巨大变化,是当今最流行的程序设计方法,其本质是把数据和处理数据的过程抽象成一个具有特定身份和某些属性的实体——对象。

1.1.2 基本概念

1. 对象的概念

“对象(Object)”一词,早在 19 世纪就有现象学大师胡塞尔提出并定义。对象是人们要进行研究的任何事物,从最简单的整数到复杂的飞机等均可看作对象,它不仅能表示具体的事物,还能表示抽象的规则、计划或事件。对象是世界上的物体在人脑中的映像,是人的意识之所以为意识的反映,是作为一种概念而存在的东西,它还包括人的意愿。例如,当认识到一种新的物体,它叫汽车,于是在人的意识当中就形成了汽车的概念。这个概念会一直存

在于人的思维当中,并不会因为汽车的消失而消失。

对象具有状态,一个对象用数据值来描述它的状态。此外还有操作,用于改变对象的状态,对象及其操作就是对象的行为。对象实现了数据和操作的结合,使数据和操作封装于对象的统一体中。

2. 类

类是具有相同或相似性质的对象的抽象。因此,对象的抽象是类,类的具体化就是对象,也可以说类的实例是对象。类具有属性,它是对象的状态的抽象,用数据结构来描述类的属性。同样,类具有操作,它是对象的行为的抽象,用操作名和实现该操作的方法来描述。

1.1.3 特点

面向对象主要有三大特点(封装、继承和多态),下面对这三个特点进行说明。

1. 封装

组装计算机时,通常只需关心相应组件的接口规范,并不关心它的内部构造和工作原理。这里组装人员所关心的计算机组件只是一个“黑盒子”,只要符合规范就行,即计算机各组件对于组装人员来说是一个实体,是已经封装好的。

面向对象程序设计的核心就是封装。在面向对象程序设计中,封装是通过把一组数据与和数据有关的操作集合放在一起形成对象来实现的。对象通过特定的接口与外部发生联系,内部的具体细节则被隐藏起来,对外是不可见的。封装的目的就是防止非法访问,用户只能通过对象的接口利用对象提供的服务,看不到其中的具体实现细节。

在面向对象编程语言中,封装后得到的基本单元是类。类是数据及其相关操作的集合体,是对客观事物的抽象和描述,对象是类的实例。类所有的对象都具有相同的数据结构和操作代码。就像按照同一张汽车设计图纸,可以造出许多具体的小汽车。换成计算机语言,就是利用一个汽车类可以创建多个汽车对象。

类是数据及相关操作的封装体。类中的具体操纵细节被封装起来,用户在使用一个已经定义的对象时无须了解其内部的实际工作流程,只要知道如何通过对外接口使用它即可。只要保持类接口不变,该类的内部工作流程可以任意改动,相应程序都无须作任何修改。

面向对象编程语言中预定了大量的类,提供给程序员使用。同时,还允许程序员定义自己的类,以满足特定的需要。事实上,在使用面向对象编程语言时,程序员所需要做的就是定义类、创建对象、访问对象。访问对象也称为将消息发送给对象。程序发送消息给某个对象,该对象便知道此消息的对应目的,进而执行对应的程序代码。

2. 继承

继续前面汽车的例子,目前的汽车大多利用汽油作燃料,但也有部分汽车被改成双燃料汽车,除汽油外,还可使用液化石油气作燃料。这种汽车事实上只是在保留原有汽车的全部特性的基础上增加了部分功能,即这种汽车是原有汽车的继承。

面向对象程序设计同样存在继承机制,通过继承可以支持代码重用。继承是在扩展现有类的基础上定义新类的过程。当一个新定义的类是基于原有类时,新类将共享原有类的属性,并且还可以添加新的特性,其中新类被称为原有类的子类,原有类称为父类。继承提供了一种基于其他类来创建新类的方法。合理地使用继承,可以减少很多重复的劳动。

3. 多态

日常生活中人们说去运动,有可能跑步,有可能游泳,也可能踢足球,或者其他,这随个人的爱好而不同。在面向对象的程序设计中,通过多态性来支持这一思想。多态使得相同的消息被不同的对象接收时,可能产生不同的动作。

面向对象编程语言通过类继承过程中的方法覆盖和方法调用时的动态绑定来实现多态性,从而达到不同的对象按照自身的需求对同一消息进行正确处理的目的。

1.1.4 面向对象编程语言

较典型的面向对象编程语言很多,如 simula 67、Smalltalk、EIFFEL、C++、Java、C#等。虽然面向对象的编程语言很多,但是发展情况却不尽相同。在上面列出的几种语言里面,simula 67、Smalltalk、EIFFEL 诞生较早,但现在基本不使用了;使用较多的是 C++、Java、C#。本书关于面向对象的介绍将全部以 Java 语言来阐述。

1.2 Java 语言概述

1.2.1 诞生及发展

自从个人计算机(Personal Computer,PC)问世以来,计算机的应用领域迅速扩大,全球使用计算机的人口快速增长。计算机技术正在不断地改变着人们的生活和工作。1989年,由 James Gosling 所领导的 Oak 语言的研究开始;1990年由 Sun 公司中的 Patrick Naughton、Gosling 及 Mike Sheridan 三位开始领导 Green(绿)计划,此乃 Java 的培育之母。

1991年,Green 计划主要致力于智能型家电的研究,而 Oak 语言的编译器以及解释器也在此时完成,该计划旨在使家电计算机化,能够相互通信,从而达到以智能化方式控制家具环境的目的。要为家用电子设备编程,就需要一种编程语言。由于家用电子设备的软件要在不同的芯片上运行,当时的计算机语言(包括 C、C++)都不能满足要求。因此,计划负责人 James Gosling 设计了一种新的编程语言,并受办公室窗外一大片橡树的启发,将其命名为 Oak。后来,计划组成员发现 Oak 已经是另一种程序设计语言的名称了,因此将 Oak 改名为 Java。

1994年,Green 计划投入发展 WWW,就是 Java 语言开始大放光明的时候。这时,第一个可看 Java 的 WWW 浏览器——WebRunner 诞生,即 HotJava 的前身。最终 Sun 公司于 1995年在 SunWorld'95 上正式发布 Sun Java 以及 HotJava 浏览器。1995年 10月,Oracle 公司、Borland 公司获得 Java 许可证。Netscape 也与 Sun 合作,Navigator 开始支持 Java,Sun 和 Netscape 发布 JavaScript——一种基于 Java 语言的脚本语言,可供非编程人员使用。Sun、Netscape 和 SGI 宣布缔结软件联盟,以开发 Java 和 Internet 三维和多媒体交互性工具。1996年,Netscape Navigator 2.0 支持 Java 正式发布。1996年,Sun 推出 Java 开发工具包 JDK1.0 (Java Development Kit),为开发人员提供用来编制 Java 应用软件所需的工具。Sun 推出 Java 数据库连接 JDBC(JavaDataBaseConnectivity)数据库 API。1998年,JDK1.2 发布。1999年,Java 技术被分成 J2SE、J2EE 和 J2ME 三部分,Java Server Pages (JSP)技术也公之于众。2000年,JDK1.3 发布,Java 开发者成员上升到 150 万。2004 年是