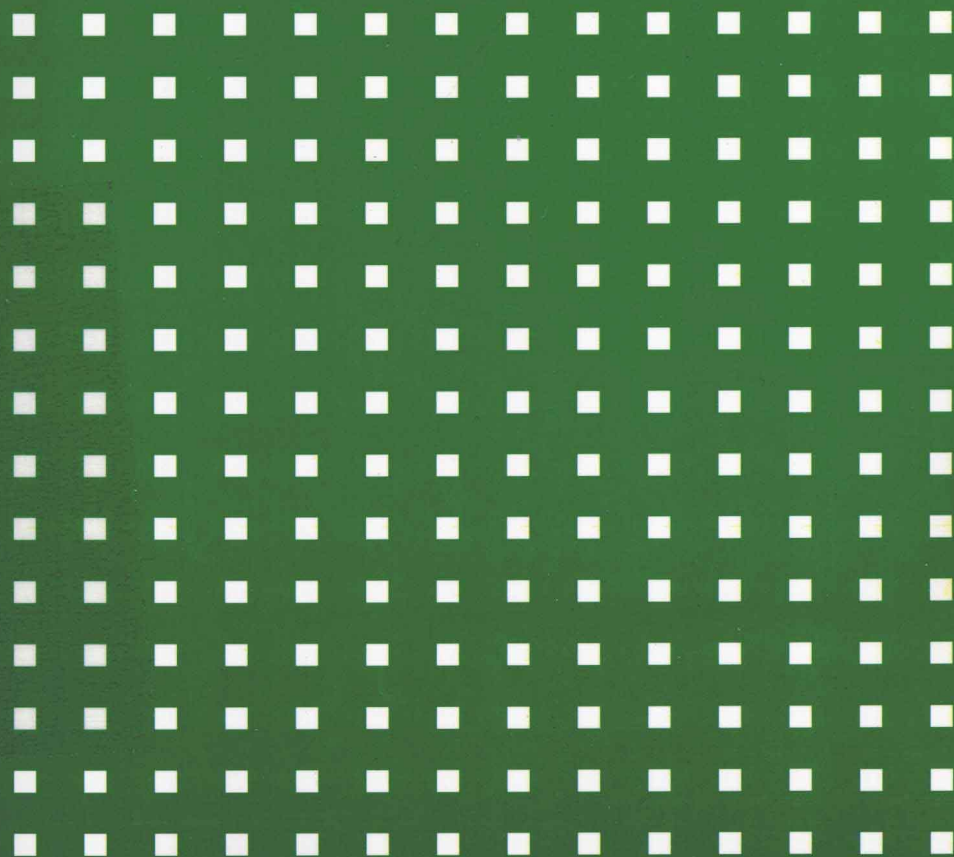


新编Java程序设计教程

陈轶 姚晓昆 杨宇仙 主编



高等学校计算机专业教材精选·算法与程序设计

新编Java程序设计教程

陈轶 姚晓昆 杨宇仙 主编

清华大学出版社
北京

内 容 简 介

这是一本针对 Java 语言程序设计的学习书。本书分成 13 章,涵盖了当前 Java SE 中的基础知识、面向对象程序设计思路和 Java 的常见的应用接口。结合最新的 JDK 7.0,本书涵盖内容较为全面,具体包括 Java 的基本数据类型和基本运算、Java 控制语句、数组、类和对象、继承和多态性、异常处理、字符串处理、Java 的图形界面编程、事件处理、Java 的文件处理、Java 的网络编程的实现、Java 的数据库编程以及 Java 解析处理 XML 的常见应用接口等内容。

为了让读者更好地学习 Java 语言,本书将每章分成 Java 语言知识点详细介绍和应用实例说明两部分。知识点部分对学习过程中需要注意的知识点和一些常见的问题做了归纳和总结,能帮助读者对关键知识点快速地了解 and 掌握。应用实例通过一个简单扑克牌游戏展开,充分结合相应的 Java 知识点内容,采用面向对象程序设计思路,由浅入深,由简入繁,通过详细的实例代码说明应用开发的实现,立体地指导读者深入理解和掌握 Java 语言,通过实现理论和实践相结合的理念帮助读者较好地学习和掌握理论知识。

本书可以作为普通高校“面向对象程序设计”或“Java 程序设计”课程的主讲教材,也可供计算机应用开发人员学习和参考。

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

新编 Java 程序设计教程 / 陈轶,姚晓昆,杨宇仙主编. —北京: 清华大学出版社, 2012. 6

高等学校计算机专业教材精选·算法与程序设计

ISBN 978-7-302-27959-4

I. ①新… II. ①陈… ②姚… ③杨… 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(2012)第 011276 号

责任编辑: 汪汉友 徐跃进

封面设计: 傅瑞学

责任校对: 时翠兰

责任印制: 李红英

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 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 33.25

字 数: 807 千字

版 次: 2012 年 6 月第 1 版

印 次: 2012 年 6 月第 1 次印刷

印 数: 1~2500

定 价: 49.50 元

产品编号: 035512-01

出版说明

我国高等学校计算机教育近年来迅猛发展,应用所学计算机知识解决实际问题,已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在,很多高等学校都在积极探索符合自身特点的教学模式,涌现出一大批非常优秀的精品课程。

为了适应社会的需求,满足计算机教育的发展需要,清华大学出版社在进行大量调查研究的基础上,组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材,把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生,使学生通过本套教材,学习计算机科学与技术方面的基本理论和基本知识,接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各高校相应课程的主讲,有一定经验积累,且编写思路清晰,有独特的教学思路和指导思想,其教学经验具有推广价值。本套教材中不乏各类精品课配套教材,并努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则,书中的许多必须掌握的理论都将结合实例来讲,同时注重培养学生分析问题、解决问题的能力,满足社会用人要求。

(4) 易教易用,合理适当。本套教材编写时注意结合教学实际的课时数,把握教材的篇幅。同时,对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答,学生上机实验指导、教学网站等辅助教学资源,方便教学。

随着本套教材陆续出版,我们相信它能够得到广大读者的认可和支持,为我国计算机教材建设及计算机教学水平的提高,为计算机教育事业的发展做出应有的贡献。

清华大学出版社

前 言

Java 语言是一门不断创新的、流行的计算机编程语言。“Java 程序设计”课程已成为各大高校计算机专业及相关专业的专业基础课程。因此,有必要编写一本面向 Java 语言开发发展现况和满足计算机教学要求的教材。本书旨在帮助广大读者了解和掌握 Java 语言的特点,以便解决“学”和“实用”脱节的问题。

笔者结合从事 Java 课程教学活动中积累的经验,从实用性、科学性以及当前的计算机技术出发编写了本书。本书分为四大部分:Java 语言基础、面向对象程序设计、图形用户界面编程和高级编程初步。

第 1 部分: Java 语言基础

从第 1 章到第 3 章,目的是帮助读者了解 Java 的语言基础并初步了解面向对象概念,具体内容涉及面向对象设计的基本概念、面向对象的基本特性、Java 的发展、Java 语言的特点、Java 语言的开发平台、Java 的数据类型和基本操作、Java 的基本输入输出处理、控制语句、方法、数组以及字符串。通过本章学习,读者可以了解用 Java 开发程序的基本过程。

第 2 部分: 面向对象程序设计

从第 4 章到第 6 章,详细介绍面向对象的基本概念。对 Java 语言实现面向对象的四大特性(抽象性、封装性、继承性和多态性)展开介绍。目的是帮助读者在掌握 Java 基础语法和一些应用接口的基础上,开发面向对象的应用。主要内容包括类的定义、类的访问控制和修饰符、特性形式的类(如内部类、抽象类)、对象的创建和对象的调用、this 和 super 关键字、包、Java 语言实现继承、重载和覆盖、继承机制导致的多态性和接口实现多态性。此外,还对集合中的列表、集、映射和泛型概念展开介绍。从这部分开始,本书引入了扑克牌游戏的应用实例,结合面向对象概念,说明 Java 语言实现面向对象编程的具体过程。

第 3 部分: 图形用户界面编程

介绍这部分知识内容的目的是让读者开发良好交互性的应用,充分调用读者学习和掌握 Java 的积极性。第 7 章介绍了图形用户界面编程、事件处理和 Java 2D 的图形处理。本书涉及常用的 GUI 组件、图形界面的布局管理、事件处理,使得 GUI 编程更加符合 HCI (Human Computer Interface),具有良好的人机交互性。

第 4 部分: 高级编程初步

从第 8 章到第 13 章涉及 Java 语言高级编程内容,是 Java 语言开发高级应用实例的支撑技术。在这一部分,对运用 Java 语言实现异常处理、多线程编程、文件处理、网络编程、访问数据库与 XML 数据处理等方面做了初步介绍,为进一步学习这些技术提供基础和学习方向。这部分介绍了异常处理、断言、多线程基本概念、线程的实现方法、线程的调度、线

程的生命周期、文件流、JDBC API、Java 开发数据库应用基本步骤、基于 TCP/IP 协议的网络编程的实现、XML 文件结构、常见解析和处理 XML 的 API 等多方面知识点。并从一个具体扑克牌游戏开发的应用实例出发,分别介绍这些概念和对应的 API 在这个应用实例中的作用。启发读者综合地应用 Java 语言知识开发具有一定规模的实际应用。

为了让读者方便练习,本书提供了相应章节的源代码,可以在清华大学出版社网站上下载。

参加本书编写的有陈轶、姚晓昆、杨宇仙、肖建、熊宇虹和唐祎玲。其中,第 1 章由唐祎玲和肖建合写,第 2 章和第 3 章由陈轶和熊宇虹合写,第 11 章由陈轶和唐祎玲合写,第 8 章和第 12 章由陈轶和姚晓昆合写。第 7 章和第 13 章由陈轶和杨宇仙合写。第 4、5、6、9 和 10 章由陈轶编写。最后由陈轶统稿。本书在编写过程中得到了江顺亮教授和邱桃荣教授的大力支持,他们对本教材提供了宝贵的意见和建议,特此表示深深的感谢。特别感谢姚力文教授,他一直鼓励和关心编者们的编撰书稿的进度。本书由林仲达教授主审,他对本书编写提出许多有益的建议,在此表示衷心的感谢。

由于编者水平所限,书中难免存在错误和不足之处,恳请广大读者对本书提出宝贵意见和建议。

编 者

2012 年 3 月

目 录

第 1 章 Java 语言概述	1
1.1 面向对象程序设计	1
1.1.1 面向对象基本概念	1
1.1.2 面向对象基本特性	2
1.2 Java 语言	3
1.2.1 Java 语言的发展	3
1.2.2 Java 语言的特点	4
1.2.3 Java 语言的开发平台	5
1.2.4 Java 程序的开发	6
1.3 本章小结	11
1.4 习题	12
第 2 章 Java 语言基础	13
2.1 标识符和保留字	13
2.1.1 标识符	13
2.1.2 保留字	13
2.2 变量和数组	14
2.2.1 直接量和变量	14
2.2.2 基本数据类型	15
2.2.3 枚举类型	17
2.2.4 数组	18
2.3 运算符	20
2.4 表达式和语句	22
2.5 Java 的标准输入和输出操作	24
2.6 流程控制语句	26
2.6.1 条件控制语句	27
2.6.2 循环语句	30
2.6.3 转向语句	33
2.7 本章小结	36
2.8 习题	36
第 3 章 字符串	38
3.1 字符和字符串	38
3.2 String 类(java.lang.String)	38

3.2.1	String 的常用构造方法	38
3.2.2	String 常用的方法	40
3.3	StringBuffer 类(java. lang. StringBuffer)	46
3.3.1	StringBuffer 类的构造方法	46
3.3.2	StringBuffer 的常用方法	47
3.4	StringBuilder 类(java. lang. StringBuilder)	52
3.4.1	StringBuilder 类的构造方法	52
3.4.2	StringBuilder 类的常用方法	53
3.5	应用实例——扑克游戏	55
3.6	本章小结	57
3.7	习题	57
第 4 章	类和对象	58
4.1	面向对象编程基础	58
4.1.1	对象和类的基本概念	58
4.1.2	类的声明	59
4.1.3	类的访问控制	61
4.1.4	类的非访问修饰符	62
4.1.5	构造方法	65
4.1.6	类的成员	71
4.1.7	对象的实现	78
4.2	包	80
4.2.1	包的创建	80
4.2.2	包的应用	82
4.3	this 关键字	83
4.3.1	用 this 关键字访问对象的成员	83
4.3.2	用 this 关键字调用构造方法	85
4.4	super 关键字	86
4.5	finalize() 方法	89
4.5.1	finalize() 方法的特点	89
4.5.2	finalize() 方法的用途	91
4.6	嵌套类	92
4.6.1	静态嵌套类	92
4.6.2	内部类	94
4.7	抽象类	98
4.7.1	抽象方法	98
4.7.2	抽象类	98
4.8	应用实例——扑克牌游戏	100
4.9	本章小结	107

4.10 习题.....	108
第5章 继承和多态.....	109
5.1 继承	109
5.1.1 父类和子类.....	110
5.1.2 继承的实现.....	110
5.2 接口	114
5.2.1 接口的格式定义.....	115
5.2.2 接口的实现.....	116
5.2.3 接口和抽象类的区别.....	119
5.3 多态性	120
5.3.1 动态绑定.....	121
5.3.2 方法的重载.....	122
5.3.3 方法的覆盖和隐藏.....	123
5.3.4 数据成员的隐藏.....	131
5.3.5 运行多态性的实现.....	135
5.4 应用实例——模拟纸牌游戏	137
5.5 本章小结	145
5.6 习题	145
第6章 集合和泛型.....	147
6.1 Java 的集合框架	147
6.1.1 访问 Collection 集合	148
6.1.2 集合与数组的转换.....	148
6.2 Set 集	150
6.2.1 java.util.HashSet<E>	150
6.2.2 java.util.TreeSet<E>	152
6.2.3 java.util.LinkedHashSet<E>	154
6.3 列表和队列	156
6.3.1 java.util.ArrayList<E>	157
6.3.2 java.util.LinkedList<E>	159
6.4 java.util.Map<K,V>	163
6.5 泛型	166
6.5.1 定义泛型类.....	166
6.5.2 类型变量的限制.....	168
6.5.3 泛型实例创建的类型推断.....	168
6.6 集合的算法	169
6.7 应用实例——纸牌游戏	172
6.8 本章小结	181

6.9 习题	181
第7章 Java 的 GUI 编程和事件处理	182
7.1 Java 基础类概述	182
7.2 Swing 包的 GUI 组件	184
7.2.1 框架窗口	184
7.2.2 标签和文本处理组件	186
7.2.3 按钮和选择交互处理的组件	190
7.2.4 面板	195
7.2.5 工具条	197
7.2.6 菜单的制作	198
7.2.7 多文档界面制作	203
7.2.8 树组件	206
7.2.9 表格组件	208
7.3 布局管理器	210
7.3.1 流布局管理器	210
7.3.2 边界布局管理方式	211
7.3.3 卡片布局管理器	214
7.3.4 网格布局管理器	216
7.3.5 网格包布局管理器	217
7.3.6 手工布局处理	221
7.4 事件处理概述	222
7.4.1 事件处理模式	222
7.4.2 事件处理类型	223
7.4.3 常用的事件处理	223
7.4.4 事件适配器	230
7.4.5 Action 动作	232
7.5 Java 图形处理和 Java 2D	235
7.5.1 Java 的颜色处理	235
7.5.2 Java 的字体处理	235
7.5.3 Java 2D 和图形处理	238
7.6 应用实例——纸牌游戏界面设计	247
7.7 本章小结	257
7.8 习题	257
第8章 异常和错误处理	258
8.1 Java 的错误处理	258
8.2 异常和异常类	258
8.2.1 异常的定义	259

8.2.2	异常的分类	259
8.3	异常处理机制	262
8.3.1	自定义异常类	262
8.3.2	异常的抛出	263
8.3.3	异常的捕捉	265
8.3.4	try-with-resource 语句	267
8.3.5	链式异常	268
8.4	断言处理机制	270
8.4.1	断言的语法	270
8.4.2	应用断言	271
8.4.3	捕捉断言错误	274
8.5	Java 标准日志处理	275
8.5.1	Java 日志框架	275
8.5.2	创建日志对象	276
8.5.3	日志的级别	276
8.5.4	记录日志	277
8.5.5	日志处理器	278
8.5.6	日志的格式化	279
8.6	应用实例——扑克牌游戏的异常处理	280
8.7	本章小结	284
8.8	习题	284
第9章	多线程和并发处理	286
9.1	Java 线程	286
9.1.1	Thread 类	287
9.1.2	Runnable 接口	290
9.1.3	线程生命周期	292
9.1.4	线程组	295
9.2	线程的同步	297
9.2.1	竞争条件	297
9.2.2	同步方法和同步语句	300
9.3	显式锁和条件对象	304
9.4	阻塞队列	308
9.5	传输队列	311
9.6	线程池	314
9.7	Fork/Join 模式	316
9.8	应用实例——扑克牌游戏的线程处理	319
9.9	本章小结	323
9.10	习题	323

第 10 章 文件和流	325
10.1 文件相关类.....	325
10.2 Java 的 I/O 流	328
10.2.1 字符流类.....	329
10.2.2 字节流类.....	332
10.2.3 对象流和对象的序列化.....	335
10.3 随机访问文件.....	339
10.4 文件的输入输出处理.....	341
10.4.1 NIO 2.0 框架的路径.....	341
10.4.2 java.nio.file.Files 实现文件处理	345
10.5 应用实例——扑克牌游戏的文件处理.....	359
10.6 本章小结.....	368
10.7 习题.....	369
 第 11 章 数据库编程	 370
11.1 JDBC 的概述	370
11.2 SQL 结构化的查询语言	372
11.2.1 常用的 SQL 语句	372
11.2.2 SQL 的存储过程	374
11.3 JDBC 访问数据库	374
11.3.1 导入 JDBC 类和接口	374
11.3.2 装载驱动程序.....	375
11.3.3 连接数据库.....	376
11.3.4 创建语句对象和执行语句.....	379
11.3.5 处理结果.....	390
11.3.6 关闭连接.....	402
11.4 事务处理.....	410
11.5 应用实例——扑克牌游戏的数据库处理.....	411
11.5.1 数据库设计.....	411
11.5.2 扑克牌游戏的数据库实现.....	414
11.6 本章小结.....	429
11.7 习题.....	429
 第 12 章 网络编程	 431
12.1 Java 网络编程基础	431
12.1.1 TCP/IP 协议	431
12.1.2 套接字和端口.....	432
12.1.3 Java 网络编程包	432
12.2 通过 URL 访问网络资源	433

12.2.1	URL 类	433
12.2.2	JEditorPane 显示 HTML 文件	435
12.3	面向连接的 TCP 协议编程	438
12.3.1	java.net.InetAddress 类	438
12.3.2	java.net.Socket 类	440
12.3.3	java.net.ServerSocket 类	443
12.3.4	TCP 协议编程实现	445
12.4	面向非连接的 UDP 协议编程	456
12.4.1	java.net.DatagramPacket 类	456
12.4.2	java.net.DatagramSocket 类	457
12.4.3	java.net.MulticastSocket 类	458
12.4.4	UDP 协议通信编程	459
12.5	应用实例——双人扑克牌游戏	467
12.6	本章小结	481
12.7	习题	482
第 13 章	Java 与 XML	483
13.1	XML 概述	483
13.1.1	XML 文件结构	483
13.1.2	XML 的相关技术	486
13.2	处理 XML 的常见 Java API	486
13.2.1	JAXP API	486
13.2.2	DOM API 处理 XML	487
13.2.3	SAX API	497
13.2.4	StAX API	501
13.3	应用实例——扑克游戏的配置处理	512
13.4	本章小结	516
13.5	习题	516

第 1 章 Java 语言概述

面向对象程序设计方法是目前使用广泛的主程序设计方法之一。与 C++ 语言一样, Java 语言也是一种典型的面向对象程序设计语言, 具有抽象性、封装性、继承性、多态性四大特征。本章介绍面向对象程序设计的基本概念和主要特性, 并从这些最基本的概念出发, 对 Java 语言的发展情况、基本特点等概念依次进行说明, 为后续的 Java 语言的学习提供基础。

1.1 面向对象程序设计

观察过去的几十年程序设计语言的发展过程, 可以发现, 程序设计语言对抽象机制的支持程度不断提高。从最早的机器语言到汇编语言, 到高级语言, 直到面向对象语言。汇编语言出现后, 程序员就避免了直接使用 0-1 串来进行编码, 而是利用符号来表示机器指令, 从而更方便地编写程序; 当程序规模继续增长的时候, 出现了 FORTRAN、C、PASCAL 等面向过程的高级语言, 这些高级语言使得编写复杂的程序变得较为容易, 程序员可以更好地处理较为复杂的问题。但是如果软件系统达到一定规模, 即使应用结构化程序设计方法, 也无法解决所产生的问题。因此, 为了降低编程的复杂性, 从而产生了面向对象的程序语言, 面向对象程序设计也随之产生。面向对象程序设计把数据和属于它们的操作放入对象中, 这样做可以解决面向过程程序设计中不可避免的许多问题, 它是更贴近真实世界的模型组织程序。

“对象”的概念并不是面向对象程序设计语言独有的概念。最早出现在 20 世纪 50 年代初关于人工智能的早期著作中。在 20 世纪 60 年代由挪威奥斯陆计算机中心 Ole-Johan Dahl 和 Kristen Nygaard 开始将“对象”这一个概念和面向对象方法引入到程序设计语言 Simula67 语言, 主要用于环境模拟。麻省理工 PDP-1 系统中也出现了“对象和实例”的概念。20 世纪 70 年代, 在施乐 PARC 研究所发明的 Smalltalk 语言中, 对“对象”的概念进一步进行改进, 提出了动态创建和销毁对象实例, 对象和实例可以使用消息, 并引入了继承的思想。这些标志着面向对象的概念正走向成熟。到 20 世纪的 80 年代, 面向对象设计方法已经成为程序设计语言的主导方法。采用面向对象的程序设计语言, 如 C++, 得到广泛应用。其他面向过程的语言如 FORTRAN、PASCAL 也将面向对象的概念引入到自身中。在这一时期, 最值得注意的是, 事件处理方式开始加入面向对象的概念。可以说, 20 世纪 80 年代面向对象程序设计语言开始占有绝对的主导地位。在 20 世纪 90 年代, 面向对象概念不断完善。1995 年, Java 语言问世, 这一典型的面向对象程序设计语言从一开始就得到人们的关注, 直至现今它已成为主流的编程语言之一。

1.1.1 面向对象基本概念

面向对象的基本概念中包含了对象、类和消息这些概念。

1. 对象(Object)

面向对象程序设计中的对象描述了现实世界中的具体物理实体在计算机中的映射,它可以是有形的,如一个人、一支笔,也可以是无形的,如一项计划等。可以说,对象是离散的、可识别的实体。每个对象都有自己的状态和行为,对象的状态可以通过某些具体的数据来描述(通常称为属性),对象的行为也就是该对象所具有的功能可以通过某些方法的集合来描述。因此对象是由一组属性和一组操作所构成的一个封闭整体。对象在计算机内存中的映像称为实例。

2. 类(Class)

在现实世界中人们常常习惯将具有一定共性的事物归为一类,从而得到一个抽象的概念,这是人类认识客观世界的常用方法。在面向对象程序设计思想中“类”就是对一组对象的抽象,即具有相同属性和操作的一组对象的集合。在面向对象的程序设计中,类是程序的基本单元,对象是类的实例。例如,陈某的诺基亚 6210C 型号的手机可以视为对象,而“手机”可视之为这个手机所属的类。

可以说,类描述对象的“基本原型”,是描述性的类别或模板,即对一组对象的抽象。它定义一组对象所能拥有的共同特征(属性和能完成的操作),用以说明该组对象的能力与性质。

3. 消息

对象能和其他对象通过信息传递机制进行通信,是传递消息的方式相互联系的通信实体。消息一般由三部分组成,即消息的接收对象名、消息操作名和必要的参数。消息传送与传统的函数调用的主要差别有以下几点:

(1) 函数调用可带或不带参数,但消息至少带一个参数(即接收该消息的对象)。

(2) 消息操作名类似于函数名,但它们有本质的不同。函数名代表一段可执行的代码,而消息名的具体功能选定还取决于接收消息的对象本身。

(3) 函数调用是过程式的,而消息传送是说明式的,具体如何做由对象根据收到的消息自行确定。

假设,用 TVSet 表示电视机类,Controller 表示遥控器类。分别创建它们的对象如下所示:

```
TVSet tv=new TVSet();           //创建电视机对象
Controller controller=new Controller(); //创建一个遥控器对象
```

controller.control(tv)可以视为一个消息,表示遥控器控制电视机。其中,controller 是消息的接收对象,control 是消息的操作名,而对象 tv 表示必要的参数。

1.1.2 面向对象基本特性

所有的面向对象程序设计语言都提供以下特性来帮助程序员设计面向对象的模型,这些特性主要为抽象性、封装性、继承性和多态性。

1. 抽象性

抽象性是关注应用最本质问题而且用来简化复杂客观问题的途径。抽象性实际上就是对对象实体的内在的、本质属性的抽象。在面向对象中,类就是将一组对象静态属相(数据)

和动态属性(方法)的抽象。

2. 封装性

所谓封装,是将代码及其处理的数据绑定在一起的一种编程机制,它保证了程序的数据不受外界干扰且不被误用。封装的目的在于将使用者与设计者分离,使用者不必知道操作实现的细节,只需用设计者提供的方法来访问对象。但值得注意的是,封装性并不能单纯地理解为信息隐藏。信息隐藏表示信息将无从获取,信息从公共接口中消失,得到了保护。但是,封装后的数据可以通过设计设置系列方法和获取系列方法来修改信息,信息并没有因此而隐藏。

Java 封装的基本单元是类,一个类当中包含了该类所描述的所有对象的共同的数据(成员变量)和方法(成员方法)。对类的使用者而言,无须了解这些类的方法或成员变量是如何形成的,只要关注如何应用它们来解决现有的问题即可。

3. 继承性

继承是指一个类(子类)享有另一个类(父类)的全部属性和方法。它是面向对象程序设计中的一个重要机制,通过这项机制,程序员在定义新类时就只需要描述其在已有类中没有的属性和方法,而其他属性和方法的定义可以通过从已有类中继承而得到,大大减少了代码的编写量。在面向对象程序开发中利用继承性有助于开发快速原型。继承性是实现软件重用性的最有效机制,促进了系统的可扩充性。

4. 多态性

多态性通俗地说,是指一个名字具有多种语义,即指同一消息为不同对象所接收时,可以导致不同的操作。可以将多态性理解为“同一接口,不同实现方法”,即在父类中定义的属性或行为可以在被其子类继承后具有不同的数据类型或实现不同的操作。在面向对象程序设计中使用多态性可以很好地降低程序的复杂度。

1.2 Java 语言

Java 从产生之初就是以面向对象的思想来设计的,因此它是一种典型的具有面向对象特点的语言。随着 Internet 的发展和广泛使用,Java 语言因为具有很好的网上移植性和安全性而被迅速地推广和使用。

1.2.1 Java 语言的发展

1991 年,Sun 公司的 James Gosling 等人所组成的小组在 Sun Microsystems 公司研究一个项目:使用 E-mail 给电冰箱、电视机等家用电器发送信息,对它们进行控制,与它们进行信息交流。开始,这些开发人员准备采用 C++,但觉得 C++ 太复杂,安全性差,最后他们基于 C++ 开发出一种新的语言 Oak(Java 的前身,意为橡树)。随后由于 Internet 的迅猛发展以及 WWW 万维网的快速增长,Oak 项目组成员认为 Oak 非常适用于 Internet,于是他们用 Oak 开发了 HotJava 浏览器,促进了 Oak 的研究和开发。1995 年,Oak 改名为 Java(一个盛产咖啡的小岛的名字)语言,并逐渐成为 Internet 上受欢迎的开发和编程语言。

从 Java 语言产生至今,其开发平台从 JDK 1.0 发展到 JDK 2.0、JDK 5.0,目前流行的是 JDK 6.0,而且 JDK 7.0 已经正式推出,日益成为主流技术。Java 语言的每一个版本的

推出,都是对自身不足的克服和完善。它日益成为网络应用的一支强有力的技术力量。Internet 使得 Java 成为网络上最流行的编程语言。由于 Java 语言所具有的安全性、跨平台、多线程等特点,使得 Java 语言已经被广泛地应用在各种领域,如网络远程教学、安全的金融应用平台、无线应用平台、远程传感器的监控等。

1.2.2 Java 语言的特点

Java 语言的迅速发展和广泛使用归功于它的设计思想。根据 Sun 公司在 Java 语言白皮书中描述的:Java 是一种简单的、面向对象的、分布式的、解释型的、健壮的、安全的、平台无关的、可移植的、高效的、多线程的、动态的语言。

1. 简单性

Java 语言参考了 C++ 语言,具有类似于 C++ 的风格和语法。但是,Java 语言在 C++ 语言的基础上进行了简化和改进,Java 摒弃了 C++ 中容易引发程序错误的地方,如多重继承、运算符重载、指针和内存管理等。为了方便编程,Java 语言提供了极为丰富的类库和接口,使得程序员可以利用它们编写复杂的应用。此外,Java 语言还引入了多线程、自动垃圾收集等特性,方便用户迅速掌握 Java。这些都使得学习和应用 Java 语言开发程序更容易。

2. 面向对象

Java 语言是一门典型的面向对象程序设计语言,它的代码设计完全是面向对象的,支持抽象性、封装性、继承性和多态性这些面向对象概念的四大特征。Java 语言充分体现了程序的模块化、重用性、可用性。

3. 分布式

Java 支持网络编程,它的 java.net 包中包含了基于 TCP/IP 协议的库,对 TCP 协议、UDP 协议的分布式通信可以很容易实现。Java 可以实现 HTTP 协议、FTP、SMTP、POP3、Telnet 等协议的应用。因此,使用 Java 应用程序通过 URL 存取文件就和访问本地文件一样简单方便。Java 的分布性为实现在分布环境尤其是 Internet 下实现动态内容提供了技术途径。

4. 健壮性

Java 的健壮性体现在以下几个方面:首先,Java 是一种强类型语言,它在编译和运行时要进行代码检查和校验。Java 非常重视早期错误检查,这些检查能够帮助及早发现开发早期出现的错误。其次,Java 丢弃了许多其他程序开发语言容易引起错误的一些概念,如 C/C++ 支持的指针。然后,Java 的数组并非采用指针实现,从而避免了数组越界的可能,在程序中由不采用指针来访问内存单元,从而也避免了许多错误发生的可能。而且,Java 通过自动垃圾收集器避免了许多由于内存管理而造成的错误。另外,Java 还具有实时异常处理的功能,强制程序员编写处理异常情况的代码,使得在出现意外情况时程序能够继续正常执行错误处理代码并结束。

5. 平台无关

Java 编译器将 Java 源程序编译成一种与体系结构无关的字节码文件。只要在运行平台上配置 Java 虚拟机(JVM),就可以在任意平台上运行 Java 字节码文件。无论计算机运行的是 UNIX 操作系统,还是 Windows 7,只要安装了 Java 虚拟机(Java Virtual Machine, JVM),就可以运行编译好的类。因此,Java 程序往往被称为“一次编译,到处运行”。