

计 算 机 系 列 教 材

Java语言程序设计

主 编 赵海廷
副主编 胡 雯 阳小兰 沈宝华



WUHAN UNIVERSITY PRESS
武汉大学出版社

计 算 机 系 列 教 材

Java语言程序设计

主 编 赵海廷

副主编 胡 雯 阳小兰 沈宝华



WUHAN UNIVERSITY PRESS
武汉大学出版社

图书在版编目(CIP)数据

Java 语言程序设计/赵海廷主编. —武汉: 武汉大学出版社, 2005. 8
(计算机系列教材)
ISBN 7-307-04642-3

I. J… II. 赵… III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 075809 号

责任编辑: 杨华 黄金文 责任校对: 黄添生 版式设计: 支笛

出版发行: 武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件: wdp4@whu.edu.cn 网址: www.wdp.whu.edu.cn)

印刷: 湖北省孝感日报社印刷厂

开本: 787×980 1/16 印张: 24.5 字数: 502 千字

版次: 2005 年 8 月第 1 版 2005 年 8 月第 1 次印刷

ISBN 7-307-04642-3/TP·166 定价: 37.00 元

版权所有, 不得翻印; 凡购我社的图书, 如有缺页、倒页、脱页等质量问题, 请与当地图书销售部门联系调换。

序

近五年来,我国的教育事业快速发展,特别是民办高校、二级分校和高职高专发展之快、规模之大是前所未有的。在这种形势下,针对这类学校的专业培养目标和特点,探索新的教学方法,编写合适的教材成了当前刻不容缓的任务。

民办高校、二级分校和高职高专的目标是面向企业和社会培养多层次的应用型、实用型和技能型的人才,对于计算机专业来说,就要使培养的学生掌握实用技能,具有很强的动手能力以及从事开发和应用的能力。

为了满足这种需要,我们组织多所高校有丰富教学经验的教师联合编写了面向民办高校、二级分校和高职高专学生的计算机系列教材,分本科和专科两个层次。本系列教材的特点是:

1. 兼顾了系统性和先进性。教材既注重了知识的系统性,以便学生能够较系统地掌握一门课程,同时对于专业课,瞄准当前技术发展的动向,力求介绍当前最新的技术,以提高学生所学知识的可用性,在毕业后能够适应最新的开发环境。

2. 理论与实践结合。在阐明基本理论的基础上,注重了训练和实践,使学生学而能用。大部分教材编写了配套的上机和实训教程,阐述了实训方法、步骤,给出了大量的实例和习题,以保证实训和教学的效果,提高学生综合利用所学知识解决实际问题的能力和开发应用的能力。

3. 大部分教材制作了配套的多媒体课件,为教师教学提供了方便。

4. 教材结构合理,内容翔实,力求通俗易懂,重点突出,便于讲解和学习。

诚恳希望读者对本系列教材缺点和不足提出宝贵的意见。

编委会

2005年8月8日



前言

Java 语言自 1995 年由 SUN Microsystems 发布以来,在短短的几年间就席卷全球,它不仅是一门最为流行的程序设计语言,并形成一种专门的技术,已经促进了 20 世纪末和 21 世纪初网络科技和网络经济的超常发展。

Java 语言采用面向对象的编程技术,其功能强大且简单易学,深受广大编程人员的偏爱,是目前推广速度最快的程序设计语言。Java 语言伴随 Internet 问世,随着 Internet 的发展而迅速发展并走向成熟。Java 语言是精心设计的语言,它具有简单性、面向对象性、与平台无关性、安全性和健壮性等诸多特点,内置了多线程和网络支持功能,可以说它是网络世界的通用语言。Java 语言正像它的象征性图标暗示的那样,是一杯冒着香浓热气的咖啡,让人爱不释手又回味无穷。

面向对象编程技术为软件开发带来了一场深刻的变革。与传统的面向过程编程语言比较,面向对象编程语言提出了许多新概念、新方法,需要学习者了解和掌握;同时又提供了大量的类库和相应的方法,程序设计人员在熟悉和掌握基本概念和基础类库之后,编写程序时才能得心应手,而且要比采用面向过程编程语言编程更加轻松。

面向对象编程技术符合人们对客观世界的认知规律,是程序设计方法学的一场变革,已逐步取代传统的、面向过程的程序设计技术,成为当今软件开发领域的主流技术。面向对象编程技能已成为软件开发人员的最基本的素质之一。

本书采用笔者多年总结、实践的“难点分散和难度降解教学方法”,对 Java 语言的有关内容采取了难点分散和难度降解技术进行处理,使学习者学起来不至于产生难学的困惑,会有一种逐步攀登不断取得成功的成就感。

本书按 72 学时教学大纲编写,理论教学为 54 学时,实训教学为 18 学时。理论教学中,内容和学时的具体分配为:第一章面向对象程序设计概述 3 学时,第二章 Java 语言和 HTML 语言简介 4 学时,第三章 Java 语言的数据类型 2 学时,第四章 Java 语言的运算符和表达式 4 学时,第五章 Java 语言的基本语句 4 学时,第六章数组、字符串和向量 4 学时,第七章 Java 语言的方法 3 学时,第八章 Java 语言类的声明和对象的实例化 3 学时,第九章 Java 语言的继承和多态性 3 学时,第十章 Java 语言的接口、异常和包 3 学时,第十一章 Java 语言的数据流 3 学时,第十二章 Java 语言的多线程 3 学时,第十三章 Java 语言的基础类库与 AWT 组件 4 学时,第十四章 Java 语言的图形用户界面设计 4 学时,第十五章 Java 语言的网络编程 4 学时和第十六章 Java 语



言的数据库编程 3 学时。

上机实训是理论教学的实践和补充,与本教材配套的《Java 语言程序设计实训》教材给出了 20 个实训,具体内容为:实训一,网上搜索、下载与简单网页制作;实训二,JDK 开发工具及其应用;实训三,Java 语言的数据类型、运算符和表达式;实训四,Java 语言的标准输入输出;实训五,Java 语言选择结构程序设计;实训六,Java 语言循环结构程序设计;实训七,Java 语言的方法程序设计;实训八,数组、字符串和向量;实训九,Java 语言的类、对象和类的继承;实训十,Java 语言的接口;实训十一,Java 语言的包;实训十二,Java 的异常处理;实训十三,Java 语言的 GUI 标准组件;实训十四,Java 语言的事件处理;实训十五,Java 语言常用的基础类库;实训十六,Java 语言流式输入输出;实训十七,Java 语言的多线程;实训十八,Java 语言的网络编程;实训十九,Java 语言的数据库编程;实训二十,Java 语言的图形处理。实训一到实训十八是必做实训,每个实训要求在 1 学时内完成,最后两个实训可以针对具体情况选做。

本书的实例和《Java 语言程序设计实训》中的实训程序都在 J2SDK 环境下调试通过。

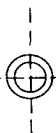
本书的第一章、第六章至第十五章及其附录由赵海廷编写,第二章、第三章由赵海廷和胡雯共同编写,第四章、第五章由赵海廷和阳小兰共同编写,第十六章由沈宝华编写,全书由赵海廷修改定稿。

在本书编写过程中,得到武汉大学王化文教授、有关领导和同仁的大力支持和帮助,在此深表谢意。

由于作者学识水平有限,书中难免有疏漏和错误,恳请同仁批评、斧正!

作 者

2005 年 5 月



目 录

第一章 面向对象程序设计概述	1
1.1 面向对象的编程思想和方法	1
1.1.1 面向对象方法是一种认知方法	1
1.1.2 面向对象语言与面向过程语言的比较	2
1.1.3 面向对象编程方法与软件的集成	3
1.2 面向对象程序设计的概念	3
1.2.1 对象(Object)	3
1.2.2 消息(Message)	4
1.2.3 类(Class)	5
1.2.4 继承(Inheritance)	6
1.2.5 多态性(Polymorphism)	6
1.3 对象的属性与相互关系	7
1.3.1 对象的属性	7
1.3.2 对象的相互关系	7
1.4 面向对象程序设计的特点	8
1.4.1 资源的可重用性	8
1.4.2 功能的可扩展性	9
1.4.3 系统的可管理性	9
1.5 面向对象系统的开发过程	10
1.5.1 面向对象系统的分析方法	10
1.5.2 面向对象系统的设计方法	10
1.5.3 面向对象系统的实现方法	10
1.6 Java 的开发与运行环境	11
1.6.1 J2SDK 的安装	11
1.6.2 J2SDK 的设置	11
1.6.3 开发与运行 Java 程序的步骤	12
第二章 Java 语言和 HTML 语言简介	14



2.1	Java 语言的 Application 程序	14
2.1.1	源代码的编辑	15
2.1.2	字节码文件的生成	16
2.1.3	字节码文件的解释执行	17
2.2	HTML 语言简介	19
2.2.1	HTML 的基本要素	20
2.2.2	HTML 文件的结构	22
2.2.3	在 HTML 文件中加入链接、图像和声音	22
2.2.4	Applet 标记	24
2.2.5	常用 HTML 标记	25
2.2.6	HTML 文件的编辑	25
2.3	Java 语言的 Applet 程序	26
2.3.1	源代码的编辑	26
2.3.2	代码的嵌入	27
2.3.3	Applet 程序的运行	27
2.4	Java 语言字符界面的输入输出	29
2.5	Java 语言图形界面的输入输出	31
2.5.1	Java Applet 图形界面输入输出	31
2.5.2	Java Application 图形界面输入输出	33
2.6	Java 语言的特点	34
第三章 Java 语言的数据类型		38
3.1	Java 语言的标识符	38
3.2	Java 语言的关键字	39
3.3	Java 语言的常量和变量	40
3.3.1	常量和符号常量	40
3.3.2	变 量	42
3.3.3	变量作用域	43
3.4	Java 语言的基本数据类型	43
3.4.1	整型	43
3.4.2	实型	44
3.4.3	字符型	45
3.4.4	布尔型	45
3.5	变量的初始化	45
3.6	简单程序举例	46

第四章 Java 语言的运算符和表达式	51
4.1 赋值、算术运算符及其表达式	51
4.1.1 赋值运算符及其表达式	51
4.1.2 算术运算符及其表达式	52
4.2 关系、逻辑运算符及其表达式	55
4.2.1 关系运算符及其表达式	55
4.2.2 逻辑运算符及其表达式	57
4.3 位运算符和表达式	58
4.3.1 计算机内数据的表示	59
4.3.2 位运算符及其表达式	60
4.4 其他运算符	64
4.5 Java 语言运算符的优先级和结合性	65
4.6 数据类型的转换	66
4.6.1 数据类型的自动转换	66
4.6.2 数据类型的强制转换	66
第五章 Java 语言的基本语句	68
5.1 Java 语言的三种基本结构	68
5.2 Java 语言的顺序结构语句	68
5.2.1 变量声明语句和表达式语句	69
5.2.2 复合语句和分程序	69
5.3 选择结构及其程序设计	71
5.3.1 条件运算符和条件表达式	71
5.3.2 if...else 语句	72
5.3.3 switch...case 语句	79
5.4 循环结构及其程序设计	81
5.4.1 while 语句	81
5.4.2 for 语句	83
5.4.3 do...while 语句	84
5.4.4 循环嵌套	85
5.5 转移控制语句	86
5.5.1 break 语句	87
5.5.2 continue 语句	88
第六章 数组、字符串和向量	94
6.1 一维数组	94



6.1.1	一维数组的声明	94
6.1.2	创建一维数组	95
6.1.3	一维数组的初始化	95
6.1.4	一维数组应用举例	96
6.2	多维数组	97
6.2.1	多维数组声明	97
6.2.2	创建多维数组	97
6.2.3	多维数组的初始化	98
6.2.4	多维数组应用举例	98
6.3	字符数组	100
6.3.1	字符数组声明	100
6.3.2	创建字符数组	100
6.3.3	字符数组初始化及其举例	100
6.4	不变字符串: String 类	103
6.4.1	String 类的构造方法及其应用举例	103
6.4.2	字符串的常用方法	105
6.4.3	字符串与子字符串操作	105
6.4.4	toString 方法及其他方法	107
6.5	可变字符串: StringBuffer 类	108
6.5.1	StringBuffer 类的构造方法	108
6.5.2	StringBuffer 类的常用方法	109
6.6	向量	110
6.6.1	向量的声明	111
6.6.2	Vector 类的常用方法	111
第七章	Java 语言的方法	114
7.1	return 语句	114
7.2	Java 语言方法的定义、返值和调用	115
7.2.1	Java 语言方法的修饰	115
7.2.2	Java 语言方法的定义	117
7.2.3	Java 语言方法的调用	118
7.2.4	Java 语言方法的返回值	120
7.3	方法之间的数据传递	122
7.3.1	方法间的数值传递	122
7.3.2	方法间的引用传递	123
7.4	Java 语言方法的递归调用	126

7.5	Java 语言方法的命令行参数	131
第八章 Java 语言类的声明和对象的实例化		
8.1	Java 语言的类	135
8.1.1	Java 语言类的声明	135
8.1.2	Java 语言类的修饰	136
8.1.3	Java 语言类的类体	137
8.1.4	Java 语言类的构造方法	139
8.2	Java 语言类的成员变量	140
8.2.1	Java 语言类成员变量的声明	141
8.2.2	Java 语言类成员变量的修饰	142
8.3	Java 语言类的成员方法	145
8.3.1	成员方法的设计	145
8.3.2	成员方法的声明和修饰	147
8.3.3	方法体	149
8.3.4	消息的传递	151
8.4	Java 语言对象的实例化	153
8.4.1	创建对象	153
8.4.2	使用对象	154
8.4.3	清除对象	154
第九章 Java 语言的继承和多态性		
9.1	继承与类的层次	156
9.1.1	继承的概念	156
9.1.2	类的层次	156
9.2	Java 语言的继承	157
9.2.1	父类和子类	158
9.2.2	成员变量的继承与隐藏	160
9.2.3	成员方法的继承与覆盖	162
9.3	Java 语言的多态性	165
9.3.1	多态性的概念	165
9.3.2	重载实现多态	165
9.3.3	覆盖实现多态	166
9.4	Java 语言构造方法的缺省、继承与重载	168
9.4.1	缺省的构造方法	168
9.4.2	构造方法的继承	170



9.4.3 构造方法的重载	173
---------------------	-----

第十章 Java 语言的接口、异常和包

180

10.1 Java 语言的接口	180
10.1.1 Java 语言接口的声明	180
10.1.2 Java 语言接口的继承关系	181
10.1.3 Java 语言接口的实现	182
10.2 Java 语言的异常及其处理	184
10.2.1 异常及其产生的原因	184
10.2.2 Java 的标准异常类	185
10.2.3 编译时对异常的检查 and 异常的层次	186
10.2.4 异常的处理技术	187
10.3 Java 语言的包	194
10.3.1 程序包的引用	195
10.3.2 声明用户的程序包	195
10.3.3 Java 语言的系统包	196
10.3.4 包应用举例	197

第十一章 Java 语言的数据流

204

11.1 Java 数据流的输入/输出	204
11.2 Java 输入/输出流类的层次结构	205
11.3 常用流类的常用方法	208
11.3.1 InputStream 和 OutputStream 抽象类	208
11.3.2 Reader、Writer 子类及其常用的方法	209
11.4 常用流类应用举例	210
11.4.1 文件输入/输出流的应用	210
11.4.2 缓冲输入/输出流的应用	211
11.4.3 数据输入/输出流的应用	213
11.4.4 字节数组输入/输出流的应用	215
11.5 File 类和 RandomAccessFile 类及其举例	217
11.5.1 File 类	217
11.5.2 随机存取文件类	218

第十二章 Java 语言的多线程

222

12.1 线程与多线程	222
12.1.1 线程的概念	222



12.1.2 线程的结构	223
12.2 线程的生命期	223
12.3 线程状态的转换方法	225
12.3.1 启动线程(start())方法	225
12.3.2 检查线程(isAlive())方法	225
12.3.3 挂起线程方法	226
12.4 线程的优先级及其设置	227
12.4.1 线程的优先级	227
12.4.2 优先级的设置	227
12.5 创建线程	228
12.5.1 继承 Thread 类创建线程	229
12.5.2 实现 Runnable 接口创建线程	230
12.6 同步问题	232
12.6.1 问题的提出	232
12.6.2 对象的锁定标志	234
12.6.3 同步方法	235
12.6.4 死锁问题	237
 第十三章 Java 语言的基础类库与 AWT 组件	239
13.1 基本类库	239
13.1.1 Object 类	239
13.1.2 数据类型类	240
13.1.3 Math 类	241
13.1.4 System 类	242
13.2 标签、按钮和滚动条的应用	246
13.2.1 标签	247
13.2.2 按钮	248
13.2.3 滚动条	250
13.3 单选按钮、复选框和选项框	251
13.3.1 单选按钮	251
13.3.2 复选框	253
13.3.3 选项框	255
13.4 文本框、文本区和列表	257
13.4.1 文本框	257
13.4.2 文本区	258
13.4.3 列表	260



13.5	Swing 介绍及其应用	262
13.5.1	Swing 按钮和标签的应用	263
13.5.2	Swing 边界和滑动条的应用	266
13.5.3	Swing 单选按钮、复选按钮和列表框的应用	270
第十四章	Java 语言的图形用户界面设计	278
14.1	组件的布局管理	279
14.1.1	顺序布局、边沿布局和卡片布局	279
14.1.2	网格布局、网格包布局和人为布局	281
14.1.3	事件处理综合举例	285
14.2	窗口和面板	290
14.2.1	可关闭窗口的创建	290
14.2.2	窗口事件适配器的介绍	291
14.2.3	向窗口加入组件和多重窗口	293
14.2.4	面板	295
14.3	菜单	297
14.3.1	菜单的有关概念	297
14.3.2	向窗口加入菜单	299
14.3.3	弹出式菜单	301
14.4	对话框	305
14.4.1	自定义对话框	305
14.4.2	文件对话框	307
第十五章	Java 语言的网络编程	310
15.1	网络编程简介	310
15.1.1	网络通信协议介绍	310
15.1.2	通信端口介绍	312
15.1.3	Java 语言网络功能的介绍	312
15.2	URL 类及其应用	313
15.2.1	URL 的概念、构造方法和 URL 的对象属性	313
15.2.2	创建 URL 对象及其应用	316
15.2.3	利用 URL 对象读取服务器文件	317
15.2.4	利用 URL 读取文件时的异常及其处理	318
15.3	URLConnection 类及其应用	320
15.3.1	URLConnection 类	320
15.3.2	AppletContext 接口类	320

15.3.3 从网络中获取图像应用举例	321
15.4 Socket 接口	323
15.4.1 Socket 接口的基本概念	323
15.4.2 利用 Socket 接口通信的步骤	323
15.4.3 Socket 类及其应用	324
15.4.4 ServerSocket 类及其应用	327
15.5 Datagram 数据报	330
15.5.1 DatagramPacket 类	331
15.5.2 DatagramSocket 类	332
15.5.3 Datagram 数据报应用举例	333
15.5.4 网络测试应用程序	342
第十六章 Java 语言的数据库编程	346
16.1 JDBC	346
16.1.1 JDBC 简介	346
16.1.2 JDBC 驱动程序	347
16.1.3 JDBC 与 ODBC	348
16.2 JDBC 中主要的接口和类	350
16.2.1 DriverManager 类	350
16.2.2 Connection 接口	351
16.2.3 Statement 接口	352
16.2.4 ResultSet 接口	354
16.3 结构化查询语言 SQL 简介	355
16.3.1 数据定义语句	355
16.3.2 数据操纵语句	356
16.4 使用 JDBC 访问数据库	357
16.4.1 创建数据源	357
16.4.2 连接 Access 数据库	360
16.4.3 查询数据库	360
附录 A Java 语言运算符的优先级和结合性表	366
附录 B ASCII 码表	367
附录 C Java 语言异常和错误信息	368
参考文献	373



第一章 面向对象程序设计概述



Java 是一种当今应用广泛的面向对象的程序设计语言。本章将介绍:面向对象方法是一种认知方法、面向对象语言与面向过程语言的比较、面向对象编程技术与软件的集成、面向对象程序设计语言的对象、消息、类、继承和多态性,对象的属性、对象的相互关系,面向对象程序设计语言系统资源的可重用性、功能的可扩展性和系统的可管理性,面向对象系统的分析方法、面向对象系统的设计方法和面向对象系统的实现方法,J2SDK 的安装与设置、开发与运行 Java 程序的步骤等。

1.1 面向对象的编程思想和方法

人们从不同角度看面向对象编程技术,认为它既是一种新的系统设计方法,也是一种新的程序设计方法,还是一种新的软件构造方法,等等。本节将从面向对象是一种认知方法学、面向对象与面向过程语言的比较、面向对象技术与软件的集成方面讨论面向对象的编程思想和方法。

1.1.1 面向对象方法是一种认知方法

人们认识事物的过程有如下两种方法。

其一是从一般到特殊的演绎方法。以学校为例,最初人们看到“学校”这样一个词,在对其进行分类的过程中,不断理解这个词的含义,进而知道学校有大学、中学和小学之分。再进一步分类,又知道大学可分为综合性大学、理、工、农、医和文科大学,等等。在不断分类到一定程度之后,用面向对象的方法,就是对各类对象进行状态描述和功能定义,以明确这一类对象所能完成的工作。然后就是让这一类对象运转起来,也就是使各类对象建立联系,使用继承和类比方法进行状态转换,完成它们应有的功能。对于学校这一系统,就是建立各教学保障单位与各教学单位的联系,教师与学生的联系,从而使这个教学系统正常运转起来。

其二是从特殊到一般的归纳方法。今天看到的一只黑猫,它是一个对象,明天又看到一只白猫,它也是一个对象,这两个对象除了颜色不同外,其他有关猫的特征完全一样。这样可以构造一个猫类,其中描述猫所具有的共同特征,例如,会叫,具有犬齿,嗅觉灵敏,毛具有一定的颜色,会捉老鼠等。而具体的黑猫与白猫都是猫类的一个实例。因此,面向对象方法很适合人们的认知方式。

面向对象方法提供了从一般到特殊的演绎手段,又提供了从特殊到一般的归纳方法。这种分类、归纳的方法在面向对象程序设计技术中是非常有用的,是一种很好的认知方法。

1.1.2 面向对象语言与面向过程语言的比较

目前,程序设计语言可分为两大类:面向过程的程序设计语言和面向对象的程序设计语言。在算法语言发展过程中,面向过程的结构化程序设计语言起到过重要的作用。面向过程的结构化程序设计语言强调的是功能的调用和程序的模块化,它将要解决问题的过程看做是一个处理过程。而面向对象程序设计语言则综合了功能抽象和数据抽象,它将要解决问题的过程看做是为分类演绎的过程。

它们之间可以从如下四个方面进行比较:

在面向过程的结构化程序设计中,模块是对功能的抽象,每个模块就是一个处理单位,它具有输入、输出。而在面向对象的程序设计中,对象是包括数据和操作方法的整体,是数据和功能的抽象和统一。也可以这样说:对象包含了模块。

在面向过程的结构化程序设计中,过程是一个独立的实体,显式地为它的使用者所见,而且对于相同的输入参数,在每次的过程调用时,其输出的结果是相同的。而在面向对象的程序设计中,方法隶属于对象,它不是独立存在的实体,而是对象功能的体现。从对象的实现机制来看,对象是一台“自动机”,其中私有状态表示了对象的状态,该状态只能由对象的操作来改变它。Smalltalk 把操作称做方法(method),C++语言等称之为函数。每次改变状态,只能由其他对象向该对象发送消息,对象响应消息后,按照消息的模式找到与之匹配的方法,并执行该方法。应该注意,发送消息和过程调用的意义是不同的。发送消息只是触发自动机,同样的输入参数可能因为自动机状态的不同其输出结果也不同,即,同一条消息的多次发送可能产生不同的输出结果。

类型是数据的抽象,即定义了一组具有共同特征的数据以及定义可以操作这些数据的一组函数。但是,类所定义的数据集(包括数据和方法)比面向过程语言的数据定义的数据集要复杂得多。

例如在C语言中,“int i;”定义变量*i*是一个整型的变量,而类需要先进行类的类型说明,然后才能创建类的实例。在类中,还要规定一些创建实例的方法,在C++中称为构造函数,在Java等面向对象程序设计语言中称为构造方法。在实例变量不再使用时,还需要使用析构函数或自动回收机来释放该对象所占用的内存空间。

在面向过程的结构化程序设计方法中,其核心是逐步细化。这种自顶向下的设计方法是通过不断地在控制结构中增加细节来开发系统程序,它产生的模块往往为了满足特定的需要,其可重用性较差。面向对象循环设计语言系统设计的方法以数据为核心来开发模块,同时一体化地考虑所有操作功能,抓住了程序设计中最不易变化的部分——数据,因此对象常具有良好的可重用性,而且,类还引入了类的继承机