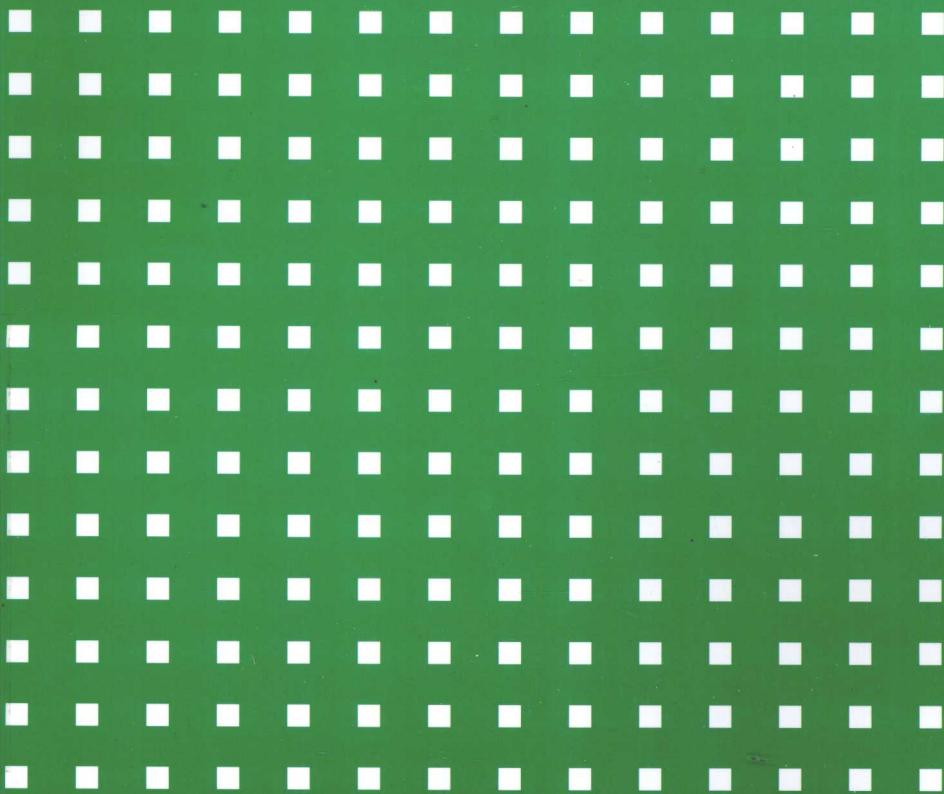


# Java 程序设计

## (第2版)

娄不夜 主编  
娄不夜 王利 编著



高等学校计算机专业教材精选·算法与程序设计

# Java 程序设计

## (第2版)

娄不夜 主 编  
娄不夜 王 利 编 著

清华大学出版社  
北京

## 内 容 简 介

为更好地适应 Java 技术的发展、满足 Java 程序设计教学的需要,本版在原书《面向对象的程序设计与 Java》的基础上进行了修订和补充。

本书从类和对象的概念入手,着力介绍 Java 面向对象的程序设计,旨在帮助读者建立面向对象的思维、掌握面向对象编程的基本方法和技能。

本书以“零”为起点,较为全面地介绍了 Java 语言的各种特性。全书共分 14 章,内容包括:类和对象、继承和嵌人类、接口和包、标注、例外处理、线程与同步、文件和输入输出流、泛型和集合框架、GUI 设计、applet 等。

本书立足基本理论和方法,注重实践与应用环节。对概念、原理和方法的描述力求准确、严谨,对示例代码力求精简、规范。本书每章的最后两部分都是小结和精选习题,便于读者复习、总结、巩固、练习与提高。

本书可作为普通高等院校计算机及相关专业的教材,也可作为读者自学 Java 语言和面向对象编程技术的参考书。

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

Java 程序设计/娄不夜主编. —2 版. —北京:清华大学出版社,2010.1

(高等学校计算机专业教材精选·算法与程序设计)

ISBN 978-7-302-20984-3

I. ①J… II. ①娄… ②王… 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(2009)第 163222 号

责任编辑:汪汉友

责任校对:白蕾

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者:北京嘉实印刷有限公司

装 订 者:三河市深源装订厂

经 销:全国新华书店

开 本:185×260 印 张:23.5

字 数:566 千字

版 次:2010 年 1 月第 2 版

印 次:2010 年 1 月第 1 次印刷

印 数:1~4000

定 价:36.00 元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。  
联系电话:010-62770177 转 3103 产品编号:031767-01

# 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了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,并力图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、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,相信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

为更好地适应 Java 技术的发展、满足 Java 程序设计教学的需要,本版在原书《面向对象的程序设计与 Java》的基础上进行了修订和补充。

本版的新内容如下。

- (1) assert 断言包含在 3.1 节(语句概述)中。
- (2) 格式化输出在 3.5.3 小节介绍。
- (3) 从控制台获取输入在 3.5.4 小节介绍。
- (4) 枚举类型在 4.8 节介绍。
- (5) 基本类型与包装类之间的自动转换在 5.4.2 小节介绍。
- (6) 嵌入类在 6.2 节介绍。
- (7) 标注在 7.3 节介绍。
- (8) import static 声明在 7.4.2 小节介绍。
- (9) RandomAccessFile 在 11.4.3 小节介绍。
- (10) 泛型和集合框架在第 12 章介绍。

这些新增内容对原书原有内容没有影响,相互之间也相对独立。教学时可根据需要有选择地讲授。

本书在编排时使用了一些符号和特殊处理,下面进行简单说明。

- (1) 代码左边的行号是为了引用和讲述方便而增加的,不是代码的一部分。
- (2) 在语言成分的语法格式描述中,有关符号的含义如下。

[ ]表示该项为可选项。

<>表示该项由程序员按规则指定或定义。

…表示该项可以出现多次。

- (3) 全书贯穿着用特殊字体表示的“注意”和“说明”。

“注意”是对当前内容的关键点和难点进行总结或提示。

“说明”是对当前内容的延伸,或与当前内容相关的其他内容进行说明。

本书提供相关的教学资源,包括示例程序的源代码、教学课件以及习题中所有选择题的答案。示例程序在 JDK 6.0 版本下编译和运行。欢迎读者从清华大学出版社网站下载和使用。

作者

2009 年 10 月

# 第 1 版前言

编写本书是为了提供一本零起点、全面介绍面向对象程序设计的教材。本书的读者不需要了解 C/C++ 或其他计算机语言及其程序设计的知识,但应该对程序、数据、二进制编码、文件、CPU、存储器、输入输出设备等概念有所理解。通过本书内容的学习,将为读者进一步学习各种 Java 技术、构建面向对象的软件系统以及网上应用系统打下坚实的基础。

Java 是一种完全面向对象的程序设计语言,适合于进行面向对象的软件系统开发和开发出真正的面向对象的软件系统。在 Java 中,除了数值、字符、布尔型几种基本类型数据外,其他所有类型的数据都是用对象来表示的。一个对象既包含了表示其状态的数据,也包含了描述其行为的方法。而对一个对象的数据的访问,则是通过向对象传递消息、调用对象的方法进行的。这就是所谓的对象处理方式。

从作者多年学习和教学的经验来看,学习 Java 编程的最大困难在于建立面向对象的思维方式、熟悉面向对象的编程风格。而类和对象的概念则是其中的基础。为此,本书在第 1 章 Java 程序初步就引入类和对象的概念,并通过一个简单的例子加以具体的说明。第 2 章和第 3 章在介绍数据与数据运算、Java 语句时进一步强化类和对象的概念,以及面向对象编程的思维方式。然后,以此为基础,在后续相关章节再由易到难、系统地介绍 Java 面向对象的编程方法和技能。

字符串和数组是几乎所有的计算机语言都支持的两种基本数据结构。与大多数计算机语言不同,Java 用对象来表示字符串和数组,并采用对象的方式处理。第 6 章、第 7 章分别介绍字符串和数组的使用,同时也为了进一步巩固面向对象的编程技能。

例外处理、线程与同步和输入输出是 Java 编程的几项基础技术,是开发各种 Java 应用软件所必不可少的。本书第 8 章~第 10 章也用较大的篇幅分别对它们进行了介绍。在这里,例外、线程、输入输出流同样是对象,对这些编程技术的掌握也都需要建立在对象和对象处理的概念之上。

图形用户界面(GUI)设计是许多读者感兴趣的课题。本书第 11 章、第 12 章分别介绍了基于 Swing 的图形用户界面和小应用程序的设计。同样地,图形用户界面和小应用程序设计中涉及的组件、事件、事件监听器、布局管理等也都表示为对象并采用对象的方式进行处理。

全书立足基本理论和方法,注重实践与应用。从应用的角度来介绍基础理论知识,通过例子来说明编程的方法和过程。对基本理论、原理、方法和技术的介绍力求概念明确、结构清晰、逻辑严谨。

本书每章的最后两部分是小结和习题。小结给出了该章的内容要点,便于读者复习与总结;习题主要有选择题和编程题两种形式,用以帮助读者检查学习效果,巩固已学知识,进一步提高能力。

感谢高林教授审阅了全书,并提出了许多宝贵的意见。同时也要感谢清华大学出版社老师为本书的出版所付出的辛勤工作。另外,作者在编写本书的过程中,参阅过大量的文献和网上资料,特别是 Sun Microsystems 公司 Java 网站(<http://java.sun.com/docs/books/>)上的内容,在此一并表示感谢。

由于作者学识水平有限,本书难免有错误和不妥之处,敬请广大读者批评指正。如果读者有好的建议和要求,请与作者联系,电子邮件地址是 [loubuye@163.com](mailto:loubuye@163.com)。

作者

2004 年 2 月

# 目 录

<b>第 1 章 Java 程序初步</b> .....	1
1.1 Java 语言特点 .....	1
1.1.1 面向对象 .....	1
1.1.2 平台无关 .....	4
1.2 一个简单的 Java 程序 .....	5
1.3 编译和运行 Java 程序 .....	7
1.3.1 Java 开发包简介 .....	7
1.3.2 编译和运行 Java 程序 .....	9
1.4 Java 词法结构 .....	10
1.4.1 空白符号 .....	11
1.4.2 注释 .....	11
1.4.3 词法符号 .....	12
小结 .....	14
习题 .....	14
<b>第 2 章 数据与数据运算</b> .....	17
2.1 基本数据类型 .....	17
2.1.1 分类及特性 .....	17
2.1.2 文字 .....	18
2.1.3 变量 .....	21
2.2 基本类型转换 .....	23
2.2.1 自动转换 .....	23
2.2.2 强制转换 .....	25
2.3 运算符 .....	27
2.3.1 算术运算符 .....	27
2.3.2 关系运算符 .....	30
2.3.3 逻辑运算符 .....	32
2.3.4 位逻辑运算符 .....	33
2.3.5 位移运算符 .....	35
2.3.6 三目条件运算符 .....	36
2.3.7 赋值运算符 .....	38
2.4 表达式 .....	39
2.4.1 表达式的结果和类型 .....	40
2.4.2 表达式的计算次序 .....	41



小结 .....	43
习题 .....	44
<b>第 3 章 Java 语句 .....</b>	<b>48</b>
3.1 语句概述 .....	48
3.2 选择语句 .....	50
3.2.1 if 语句 .....	50
3.2.2 if...else 语句 .....	51
3.2.3 switch 语句 .....	53
3.3 循环语句 .....	54
3.3.1 while 语句 .....	54
3.3.2 do...while 语句 .....	55
3.3.3 for 语句 .....	58
3.4 跳转语句 .....	60
3.4.1 return 语句 .....	60
3.4.2 break 语句 .....	60
3.4.3 continue 语句 .....	62
3.5 几个类及使用 .....	63
3.5.1 Math 类 .....	63
3.5.2 System 类 .....	64
3.5.3 格式化输出 .....	65
3.5.4 从控制台获取输入 .....	66
小结 .....	67
习题 .....	68
<b>第 4 章 类和对象 .....</b>	<b>72</b>
4.1 引用类型 .....	72
4.2 类的定义与对象的创建 .....	73
4.3 成员变量 .....	74
4.3.1 成员变量的定义 .....	74
4.3.2 成员变量的初始化 .....	76
4.3.3 对成员变量的访问 .....	79
4.4 方法 .....	81
4.4.1 方法定义 .....	81
4.4.2 方法调用 .....	83
4.4.3 构造方法 .....	84
4.4.4 方法重载 .....	86
4.4.5 类方法 .....	88
4.5 以对象为单位的信息传递 .....	89

4.5.1	用对象作为参数 .....	89
4.5.2	将对象作为返回值 .....	91
4.6	“has-a”关系 .....	91
4.7	对象清除 .....	93
4.8	枚举类型 .....	95
4.8.1	基本的枚举类型及使用 .....	95
4.8.2	枚举类型的方法 .....	97
4.8.3	带参数的枚举常量与构造方法 .....	98
小结	.....	99
习题	.....	100
<b>第 5 章</b>	<b>字符串和包装类 .....</b>	<b>108</b>
5.1	String 类 .....	108
5.1.1	构造方法 .....	109
5.1.2	提取与定位 .....	111
5.1.3	字符串比较 .....	112
5.1.4	其他若干实例方法 .....	115
5.1.5	类方法 valueOf .....	116
5.2	StringBuffer 类 .....	117
5.2.1	构造方法 .....	117
5.2.2	长度与容量 .....	117
5.2.3	基本方法 .....	119
5.3	字符串的特殊处理 .....	120
5.3.1	关于字符串文字 .....	120
5.3.2	关于运算符 + .....	121
5.4	包装类 .....	122
5.4.1	基本类型的包装类 .....	122
5.4.2	基本类型与包装类之间的自动转换 .....	125
小结	.....	126
习题	.....	127
<b>第 6 章</b>	<b>继承和嵌入类 .....</b>	<b>130</b>
6.1	继承 .....	130
6.1.1	extends 短语 .....	130
6.1.2	类成员 .....	131
6.1.3	“is-a”关系 .....	133
6.1.4	成员变量隐藏 .....	133
6.1.5	方法覆盖 .....	134
6.1.6	再论构造方法 .....	138

6.1.7	扩展抽象类	141
6.2	嵌人类	143
6.2.1	静态成员类	143
6.2.2	非静态成员类	144
6.2.3	局部类	145
6.2.4	匿名类	146
6.3	Object 类	148
	小结	150
	习题	151
<b>第 7 章</b>	<b>接口和包</b>	<b>156</b>
7.1	接口	156
7.1.1	接口定义	156
7.1.2	接口实现	158
7.1.3	接口类型	158
7.1.4	名字冲突处理	162
7.2	引用类型转换	163
7.2.1	自动赋值转换	163
7.2.2	强制转换	163
7.2.3	运算符 instanceof	164
7.3	标注	164
7.3.1	标注类型	165
7.3.2	标注	165
7.3.3	预定义标注类型	167
7.3.4	标注处理	169
7.4	包	170
7.4.1	包及其使用	171
7.4.2	import static 声明	174
7.4.3	访问控制	175
	小结	178
	习题	179
<b>第 8 章</b>	<b>数组</b>	<b>183</b>
8.1	数组类型与数组变量	183
8.1.1	数组类型	183
8.1.2	数组变量	183
8.2	数组创建	184
8.2.1	数组创建表达式	184
8.2.2	数组初始化块	185

8.3	数组访问 .....	187
8.3.1	对数组元素的访问 .....	187
8.3.2	增强的 for 语句 .....	188
8.3.3	对数组成员的访问 .....	188
8.4	二维数组 .....	189
8.5	数组应用举例 .....	190
	小结 .....	193
	习题 .....	194
<b>第 9 章</b>	<b>例外处理 .....</b>	<b>197</b>
9.1	例外分类 .....	197
9.2	引发例外 .....	198
9.3	声明抛出例外 .....	200
9.4	捕捉例外 .....	203
9.4.1	try 和 catch 子句 .....	204
9.4.2	多个 catch 子句 .....	205
9.4.3	未捕捉到的例外 .....	206
9.4.4	再引发例外 .....	207
9.4.5	finally 子句 .....	208
9.5	自定义例外类型 .....	209
9.6	构造方法与例外处理 .....	211
	小结 .....	212
	习题 .....	213
<b>第 10 章</b>	<b>线程与同步 .....</b>	<b>217</b>
10.1	线程创建 .....	217
10.1.1	通过实现 Runnable 接口创建线程 .....	217
10.1.2	通过扩展 Thread 类创建线程 .....	219
10.1.3	两种方法的共性 .....	220
10.2	线程控制 .....	221
10.2.1	线程状态 .....	222
10.2.2	线程优先级 .....	222
10.2.3	yield() .....	223
10.2.4	sleep(long millis) .....	224
10.2.5	interrupt() .....	225
10.2.6	join() .....	226
10.2.7	精灵线程与程序终止 .....	227
10.3	互斥与同步 .....	227
10.3.1	临界区与互斥 .....	227

10.3.2 线程同步.....	230
小结.....	236
习题.....	237
<b>第 11 章 文件与输入输出</b> .....	<b>240</b>
11.1 File 类.....	240
11.1.1 构造方法.....	240
11.1.2 实例方法.....	241
11.2 字节流.....	243
11.2.1 InputStream 和 OutputStream .....	244
11.2.2 FileInputStream 和 FileOutputStream .....	245
11.2.3 PipedInputStream 和 PipedOutputStream .....	247
11.2.4 BufferedInputStream 和 BufferedOutputStream .....	249
11.3 字符流.....	250
11.3.1 Reader 和 Writer .....	250
11.3.2 InputStreamReader 和 OutputStreamWriter .....	251
11.3.3 FileReader 和 FileWriter .....	252
11.3.4 BufferedReader 和 BufferedWriter .....	253
11.3.5 PrintStream 和 PrintWriter .....	253
11.4 高级流.....	255
11.4.1 DataInput 和 DataOutput .....	255
11.4.2 DataInputStream 和 DataOutputStream .....	256
11.4.3 RandomAccessFile .....	258
小结.....	258
习题.....	259
<b>第 12 章 泛型和集合框架</b> .....	<b>262</b>
12.1 泛型.....	262
12.1.1 泛型定义.....	262
12.1.2 约束类型参数.....	264
12.1.3 泛型与子类型化.....	264
12.1.4 通配符.....	265
12.1.5 约束通配符.....	266
12.1.6 泛型方法.....	267
12.1.7 类型参数的使用限制.....	268
12.2 集合框架概述.....	269
12.3 表.....	271
12.3.1 List 接口.....	271
12.3.2 数组表和链式表.....	273

12.3.3	向量与栈	275
12.4	集	277
12.4.1	散列集和链式散列集	277
12.4.2	树型集	279
12.5	映射	281
12.5.1	Map 接口	281
12.5.2	映射实现类	282
12.6	Collections 类	284
	小结	286
	习题	287
<b>第 13 章</b>	<b>GUI 设计</b>	<b>291</b>
13.1	GUI 程序概述	291
13.2	事件处理	293
13.2.1	委托模型	294
13.2.2	编程方法	294
13.2.3	常用的事件类和监听器接口	297
13.3	Swing 组件的一般功能	298
13.4	容器与布局	303
13.4.1	容器组件	303
13.4.2	流式布局管理器	306
13.4.3	边框式布局管理器	307
13.4.4	网格式布局管理器	309
13.4.5	框式布局管理器	310
13.5	几个常用的原子组件	314
13.5.1	标签	314
13.5.2	按钮	315
13.5.3	复选框	317
13.5.4	单选按钮	320
13.5.5	文本域	322
13.5.6	文本区	325
13.5.7	组合框	326
13.5.8	列表框	328
	小结	330
	习题	331
<b>第 14 章</b>	<b>Java applet</b>	<b>334</b>
14.1	applet 概述	334
14.2	HTML 的 APPLET 标记	337

14.3	Applet 类 .....	340
14.3.1	小应用程序生命周期.....	340
14.3.2	主要行为方法.....	341
14.4	绘制图形.....	345
14.4.1	GUI 绘制机制 .....	345
14.4.2	绘图支持类.....	348
小结	.....	355
习题	.....	356
<b>参考文献</b>	.....	<b>358</b>

# 第 1 章 Java 程序初步

Java 是 Sun Microsystems 公司(后简称 Sun 公司)开发的一种面向对象程序设计语言,是目前网络应用软件开发的主流工具。Java 是在 C++ 的基础上发展起来的,它继承了 C++ 中大量的语法成分,但抛弃了 C++ 中冗余和容易引起问题的成分。C++ 源自于 C,而且对业已存在的 C 代码具有兼容性。与 C++ 不同,Java 是一个纯面向对象的语言。

本章首先介绍 Java 语言的两个主要特点(面向对象和平台无关),然后通过例子进一步说明类与对象的概念、Java 程序的基本结构以及 Java 应用程序的编译和运行过程,最后介绍 Java 语言的词法结构。

## 1.1 Java 语言特点

利用 Java 语言可以开发面向对象的、平台无关的、安全的、多线程的、动态的、分布式的、健壮的、高性能的程序。其中面向对象和平台无关是 Java 两个最基本也是最主要的特点。

### 1.1.1 面向对象

客观世界是由客观世界的实体及实体之间的相互联系构成的,实体之间的相互联系产生实体的行为。当试图编写一个程序以解决客观世界的某个问题时,就必然要对客观世界的问题进行抽象。

传统的面向过程的程序设计侧重于实体行为的抽象,而把表示实体状态的属性置于一个被动、附属并相对分离的地位。面向对象的程序设计则把实体的属性和行为作为一个整体加以抽象。对象和类是这一抽象过程的产物,是面向对象程序设计的核心概念。

#### 1. 对象

对象(object)是对客观世界里的任何实体的抽象。被抽象的实体可以是具体的物品,也可以是某些概念,例如一部电话机、一名学生、一门课、一台计算机、一个命令按钮等。有些实体可由其他实体组成,例如一台计算机由 CPU、存储器、显示器等实体组成。

实体有属性和行为:属性表示实体的静态特征,所有属性的组合反映实体的状态;行为表示实体的动态特征,一个行为的过程可能会影响或改变实体的状态。客观世界里的任何实体往往都有丰富的属性和复杂的行为。抽象的目的是要从这些丰富和复杂的属性和行为中选择和提炼出为解决问题所需要的属性和方法,如图 1-1 所示。

对象是客观世界实体的软件模型,由数据和方法两部分组成。数据(变量、数组)对应于属性,用于表示对象的状态。方法是对行为的实现,用于表示对象所具有的操作或所能够提供的服务。对象的数据和方法属于对象。与非面向对象程序设计语言中的过程不同,对象的方法必须通过该对象来调用。对象方法的调用意味着该对象的一个行为的发生,它可能会改变这个对象的状态,它也可能需要调用另一个对象的某个方法。



对象是数据与方法的封装体。通过封装,对象可以对外界隐藏它的数据和方法的具体实现算法,而只把方法的格式信息(方法名、形参)和行为信息(功能)露在封装界面上,如图 1-2 所示。外界要使用一个对象,并不需要了解对象内部的实现细节,而只需知道对象封装界面上的信息,即该对象能够提供哪些服务。

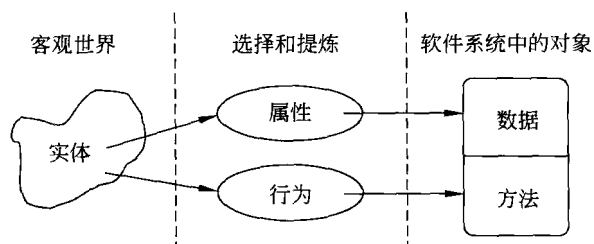


图 1-1 实体到对象的抽象

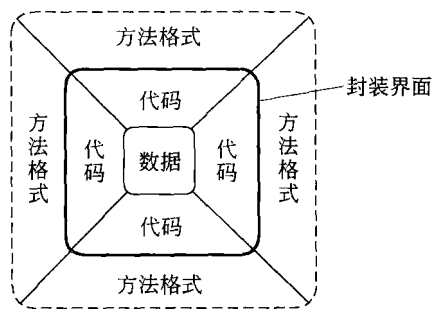


图 1-2 对象封装示意图

一般来说,外界不应该直接读取表示对象状态的数据,而应该通过调用对象的有关方法来获取对象的状态。这种用以返回对象状态的方法称为取值方法。按照惯例,取值方法一般取像 getWidth、getY 这样的名字。反过来,外界也不应该直接设置表示对象状态的数据,但可以通过调用对象的有关方法来设置对象的状态。这种用以设置对象状态的方法称为设值方法。按照惯例,设值方法一般取像 setWidth、setY 这样的名字。在对象设计中,纯粹的设值方法应尽量避免,对象状态的变化应该缘于对象的某种行为的发生,即真正的行为方法的执行。

## 2. 类

类(class)和对象关系密切,但并不是同一个概念。类是对一类相似对象的描述,这些对象具有相同的属性和行为、相同的变量(数据结构)和方法实现。类定义就是对这些变量和方法实现进行描述。类好比是一类对象的模板,有了类定义后,基于类就可以生成这类对象中任何一个对象。这些对象虽然采用相同的变量来表示状态,但它们在变量上的取值完全可以不同。这些对象一般有着不同的状态,且彼此间相对独立。

如图 1-3 所示,有一个矩形类 Rectangle,其中定义了两个变量 width 和 height,分别表示矩形的宽和高;定义的两个方法 getArea 和 getPerimeter,分别计算矩形的面积与周长。基于类创建的矩形对象虽然都用变量 width 和 height 来表示它们的状态,但每个矩形在变量上的取值是独立的。类中定义的方法指明了矩形对象可以向外界提供的服务。方法尽管定义在类中,但执行方法的主体是对象。同一个方法,如果由不同的对象去执行,一般会产生不同的结果。基于某个类生成的对象被称为这个类的实例。在 Java 中,除了数组对象,任何一个对象都是某个类的一个实例。

因为类的每个实例共用相同的方法,所以在程序运行时,不管基于这个类产生了多少个实例,类中定义的方法的代码在内存中只需要一个副本。另一方面,类的各个实例作为对象是相互独立的,每个实例有各自的表示其状态的数据。程序运行时,每当产生类的一个实例,系统就会为该实例建立相应的一组实例变量。也就是说,类中定义的实例变量在内存中可能有多个副本,每个副本属于某个实例。