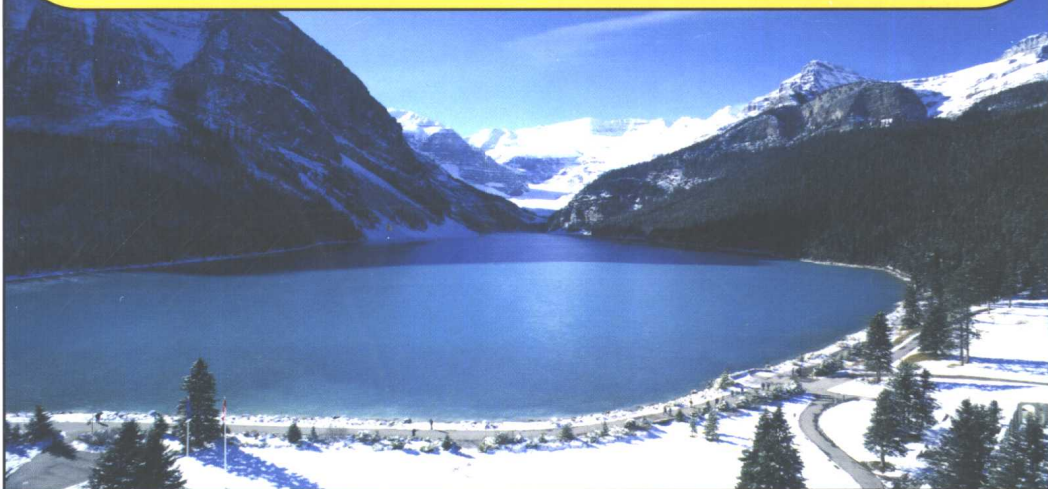


高等院校信息管理与信息系统专业系列教材



Java

程序开发教程

张基温 朱嘉钢 张景莉 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

高等院校信息管理与信息系统专业系列教材

Java 程序开发教程

张基温 朱嘉钢 张景莉 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

该书以程序设计的初学者为对象,以面向对象的程序开发方法为主线,介绍了 Java 语言和 Java 平台的基础知识并将面向对象的思想贯穿在全书之中。

本书例题丰富,内容循序渐进,通俗易懂,每章都配有一定数量的习题或思考题,适合作为本、专科学学生程序设计课程的入门教材,也可作为计算机爱好者学习面向对象程序设计的自学教材。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Java 程序开发教程

作 者: 张基温 朱嘉钢 张景莉 编著

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

[http:// www. tup. tsinghua. edu. cn](http://www.tup.tsinghua.edu.cn)

责任编辑: 徐跃进

印 刷 者: 北京顺义振华印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 18.25 **字数:** 418 千字

版 次: 2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书 号: ISBN 7-302-05747-8/TP·3395

印 数: 0001~6000

定 价: 24.00 元

出版说明

20世纪三四十年代,长期摸索前进的古老的计算技术与刚走向成熟的电子技术相结合。这一结合,不仅孕育了新一代计算工具——电子计算机,还产生了当时谁也没有料到的巨大效应。电子计算机——这种当初为计算而开发出来的工具,很快就超出计算的范畴,成为“信息处理机”的代名词;人类开始能够高效率地开发并利用信息;信息对人类社会的作用得以有效地发挥,并逐步超过材料和能源成为人类社会的重要支柱;信息产业急剧增长,信息经济高速发展,社会生产力达到了新的高度;人们的信息化意识不断加强,人类在信息资源方面开始更加激烈的竞争,社会发展走上信息化轨道。

科学技术是第一生产力,教育是基础。为了加速社会信息化的过程,以培养信息资源开发人才为目标的信息管理与信息系统专业应运而生。

对于与信息有关的学科从纵向来看,信息管理与信息系统处于信息学、信息技术、信息管理、信息经济、信息社会学这个层次的中间。它以信息学和信息技术为基础,并与信息经济和信息社会学相关联。从横向来看,它处在管理学、信息科学与技术、系统科学等有关学科领域的交叉点上。它对技术有极高的要求,又要求对组织的深刻理解和对行为的合理组织,反映了科学与人本融合的特点。这种交叉和融合正是信息管理与信息系统的最重要特征,是别的学科或专业难以取代和涵盖的。

我国的信息管理与信息系统专业创建于20世纪70年代末。在不到20年的时间里,已发展到150多个点,成为培养信息化人才的主要摇篮。其发展速度之快,影响之深远,已令世人和学术界刮目相看。

然而,作为一个新的学科,这个专业的课程体系、教学内容以及教学方法都需要经历一个逐步完善、逐步成熟的过程。特别是教材的建设更需要经过长期的实践和探索。没有这样一个过程,具有专业特点,符合中国实际的教材是不可能产生的。近20年来,大家一直在课程体系的完善和建设并具有自己专业特点的教材方面不断进行探讨。1991年全国10所财经类院校的经济信息管理专业负责人汇聚在太原召开第一次教学研讨会。以后,1993年在大连,1995年在武汉,1997年在烟台,又有更多的院校参加了这一研讨。在讨论中,各校同仁一致认为,教材建设是当务之急,它不仅直接体现和落实培养目标,同时也是学科建设的根本所在。目前一些课程缺乏专业特点,简单搬用其他专业教材的状况亟待改变。在武汉会议上,这一共识得到了与会的国家教委有关部门负责同志的赞许,清华大学出版社也对此给予了热情的支持。会议确定了首批计划编写八九本教材,由张基温教授主编,由清华大学出版社出版,并聘请了魏晴宇、陈禹两位教授作为顾问。

经过两年多的工作,在全国许多高等院校的同仁共同努力下,其中7本已完成初稿。我们希望这批教材的问世,能够起到抛砖引玉的作用,对各校信息管理与信息系统专业的建设和发展有所裨益。

近20年来的实践使我们对信息管理与信息系统专业的重要性和困难有了切身的体

会。一方面,席卷全球的信息化大潮把信息管理推到了时代发展的前沿,信息、信息管理、信息系统已经成为全社会关注的热点,这为信息管理与信息系统专业的建设创造了良好的外部条件,提供了难得的机遇。另一方面,信息技术的迅速发展与普及,多种社会经济因素的互相渗透和影响,前所未有的许多新问题、新情况的出现,又给这个专业的发展带来了很大的困难。我们深感责任之重大和任务之艰巨。在这套教材问世之时,我们再次表示这样一个心愿:希望与全国的同行共勉,为祖国信息化建设的宏伟事业多添一块砖,多加一块瓦,多出一份力,培养出更多的优秀人才。

由于上述种种原因,这套教材当然不会是完整的,也不会是完美的。它必然要不断补充、不断修改、不断完善。因此,对于它的任何修改意见,都是我们非常盼望的。希望能够在这套教材出版后,收到更多的意见和建议,使之逐步走向成熟。

全国高等院校计算机基础教育研究会
财经信息-管理专业委员会
信息管理与信息系统专业教材编委会
1997年9月

前 言

Java 是 20 世纪 90 年代初问世的一种计算机语言,在短短的几年内已风靡全球,从嵌入式系统到网络编程,都得到了空前广泛的应用。Java 集当时十余年间计算机界的几乎所有重要新技术于一身,具有许多独特的优点。其中最主要的有两个:一是面向对象——Java 是一种纯面向对象的语言,具有简洁而完整的支持面向对象方法的语言机制;二是平台兼容性好——Java 是一种计算机世界的“国际语言”,可以运行于不同平台。

面向对象的方法是目前主流的软件开发方法。本书以 Java 为语言,描述面向对象的程序开发方法,为程序开发的初学者提供一本通俗的入门教材。本书主要有两个特点:第一,本书是介绍 Java 程序开发的入门教材,以高中文化知识基础为教学起点,不要求读者在学习本书之前学过任何计算机语言或专门知识;第二,本书一开始(第 1 章)即引入对象和类等面向对象的概念,力图使面向对象的方法在学习者心中“先入为主”。

本书的第 1 章至第 3 章介绍 Java 的语言基础,并使读者对面向对象的方法有一个渐进的认识;第 4 章着重介绍继承、多态性等面向对象方法的重要概念和 Java 的支持机制,同时使读者对类、对象、方法的定义和调用等概念有较为系统和深入的认识;第 5 章和第 6 章介绍 Java 资源中常用的包和类,使读者了解 Java 资源的大体组织,为日后使用 Java 平台打下初步基础;第 7 章介绍 Applet。本书的一大特点是例题丰富,每一章都配有一定数量的习题或思考题。

建议本书的教学学时数,本科为 100 学时,其中授课 60 学时,课内上机 20 学时,另外再增加课外上机 20 学时。对于非计算机专业,可以根据需要对本书后 3 章的内容进行删选;专科可在此基础上酌情增减学时。

编者

2002 年 5 月

目 录

第 1 章 Java 入门	1
1.1 Java 简介	1
1.1.1 Java 的崛起	1
1.1.2 面向对象的程序设计语言	2
1.1.3 Java 平台和虚拟机	7
1.1.4 两类 Java 程序	8
1.1.5 Java 程序开发环境——SDK	12
1.2 Java 程序开发过程	12
1.2.1 Java 应用程序开发过程举例	12
1.2.2 Java Applet 开发过程举例	16
习题	18
第 2 章 方法设计	20
2.1 方法使用基础	20
2.1.1 方法的定义	20
2.1.2 方法的调用与参数传递	22
2.2 运算符与表达式	25
2.2.1 赋值运算	26
2.2.2 算术运算	26
2.2.3 关系运算	29
2.2.4 逻辑运算	30
2.2.5 位运算符	32
2.2.6 三元运算	35
2.2.7 其他运算符	36
2.3 流程控制语句	37
2.3.1 循环结构	37
2.3.2 选择语句	47
2.4 算法设计举例	54
2.4.1 穷举	54
2.4.2 迭代	55

2.4.3 递归	56
习题	58
第3章 数据设计	61
3.1 数据描述	61
3.1.1 标识符与关键字	61
3.1.2 数据类型	62
3.1.3 数据类型转换	64
3.1.4 字面量	67
3.1.5 变量的作用域与生存期	68
3.2 数组	72
3.2.1 一维数组	72
3.2.2 多维数组	77
3.2.3 数组的应用	81
3.3 向量	85
习题	87
第4章 Java 程序结构	88
4.1 类层次结构	88
4.1.1 继承及其描述	88
4.1.2 类的继承规则	89
4.1.3 变量的隐藏	90
4.2 对象的使用与对象间的通信	91
4.2.1 对象的创建	91
4.2.2 对象的使用	97
4.2.3 对象间的通信(消息)与实例方法的调用	104
4.3 多态性	107
4.3.1 方法的重构与重载	107
4.3.2 方法的动态调用	110
4.3.3 抽象类	113
4.4 接口与包	114
4.4.1 接口	114
4.4.2 包	116
4.5 访问控制与修饰符	117
4.5.1 类成员的访问权限	117
4.5.2 类的修饰符	121
习题	121

第 5 章 Java 资源及其利用	125
5.1 Java 的标准库结构	125
5.2 语言包	126
5.2.1 基本对象类	126
5.2.2 系统类	127
5.2.3 包装类	130
5.2.4 字符串类和缓冲串类	136
5.2.5 数学类	144
5.2.6 异常处理	146
5.3 输入与输出包	154
5.3.1 文件类	154
5.3.2 流的综述	156
5.3.3 文件的输入输出流	159
5.3.4 随机访问文件类	166
5.3.5 产生文本文件	168
5.4 其他常用资源简介	171
5.4.1 java.net 包	171
5.4.2 java.util 包	175
习题	180
第 6 章 图形用户界面 GUI	183
6.1 建立图形用户界面	183
6.1.1 图形用户界面的概念	183
6.1.2 建立和添加组件	185
6.1.3 Java 的事件处理	191
6.1.4 组件 JPanel 与布局管理器	195
6.2 几个重要的图形组件	201
6.2.1 JCheckBox	201
6.2.2 JRadioButton 和 RadioButtonGroup	205
6.2.3 JTextArea 和 JScrollPane	213
6.2.4 JTabbedPane	216
6.3 GUI 组件包与 JComponent	223
6.3.1 Swing 和 AWT 包	224
6.3.2 JComponent	225
习题	230
第 7 章 Applet	231
7.1 Applet 的设计基础	231

7.1.1	Applet 的结构	231
7.1.2	对 Applet 功能的限制	235
7.1.3	向小应用程序传递参数	235
7.2	在 Applet 中使用图形、字体和颜色	237
7.2.1	使用图形:Graphics 类	237
7.2.2	使用字体:Font 类	239
7.2.3	使用颜色:Color 类	241
7.3	在 Applet 中使用图像和声音	244
7.3.1	使用图像	244
7.3.2	使用声音	247
7.4	JAR 文件——Java 存档文件	252
	习题	253
附录 A	Java 资源一览	254
附录 B	常见的出错信息	255
附录 C	常用术语表(英文原文信息)	257

Java 入门

20 世纪 90 年代初,作为一种纯面向对象的编程语言和网络程序开发平台的 Java 诞生了。Java 的问世及其新的程序设计理念,对全世界的编程人员来说,好像在疲惫干渴之时喝进了一杯新鲜的咖啡。目前,Java 已经成为当今计算机业界不可忽视的力量和重要的发展潮流与方向。为了让读者对 Java 语言及其编程方法有一个初步了解,本章主要内容如下:

- 简要介绍 Java 的由来;
- 通过实例分别说明 Java Application(应用程序)和 Java Applet(小应用程序)的基本结构,以及它们从编辑到执行的完整过程。

通过这些内容的学习,读者将初步领略 Java 的风采,并增加对 Java 功能的认识。

1.1 Java 简介

1.1.1 Java 的崛起

1991 年 4 月 8 日,从工作站起家的 Sun 公司为了把市场扩大到消费电子产品,成立了一个专门的工作小组,着手代号为 Green 的项目,目的是开发一个分布式代码系统,让人们可以把 E-mail 发送给电冰箱、电视机等家用电器,对它们进行控制,同它们进行信息交流;这个系统还应当具有独立于软件平台的特征,并容易被程序员们接受。基于这一目标,自然应当首选当时已经广为流行的 C++ 进行开发。但是 C++ 太复杂,安全性也差,最后不得不定义一套新的语言系统,于是便有了 Oak(意为橡树,源于 Green 项目组的 James Gosling 工程师透过窗户看到的一棵树)的问世。

Oak 一方面汲取了 C++ 中的有益成分,同时融进了网络应用特性,可以称得上是一种精巧而安全的语言了。可惜,Sun 以它投标一个自认为比较合适的交互式电视项目时,却一败涂地,可怜的 Oak 几乎陷入困境。

恰巧这时,Mark Andreessen 的 Mosaic(Internet 上最早出现的 Web 浏览器)和

Netscape 的巨大成功给 Oak 项目组成员带来了新的希望,他们重整旗鼓,对 Oak 进行了一次新的整合,并给了它一个据说来自几位小组成员喝 Java(爪哇)咖啡时灵机一动想到的新名字——Java。接着,他们在 Sun 公司首席执行官 Scott McNealy 的支持下,用 Java 编制了 HotJava 浏览器,吹响了进军 Internet 的号角。

Java 由于与 Internet 的成功结合而很快风靡全球,计算机产业的许多大公司购买了 Java 的许可证,包括 IBM、Apple、DEC、Adobe、HP、Oracle、Toshiba,以及最不愿的 Microsoft。众多的软件开发商开始支持 Java 的软件产品。Java 所特有的“write once, run anywhere. (一次编写,到处运行)”的与平台无关性的概念,与 Internet 的异构网络互联一拍即合。它特别适合在网络开发中使用,这一优势直接威胁到 Wintel 联盟的垄断地位,使得它成为面向对象开发工具中潜力巨大的一员。

1.1.2 面向对象的程序设计语言

面向对象的观点认为现实世界是由各种对象组成的(准确地讲,是用对象去描述或模拟现实世界中的客观事物或概念,对象与客观事物之间是有区别的),对象通过它们之间的相互作用改变彼此的状态,推动事物的发展。Java 语言是一种面向对象的程序设计语言,它通过描述对象本身以及对象间的相互关系,并通过激发对象间的相互作用,来求解问题。用 Java 语言进行程序设计,首先要建立问题的对象模型。本节介绍用 Java 语言描述对象的方法,与对象有关的其他机制将在后面的有关章节中逐步介绍。

作为对现实世界中事物的对象,要由两个部分来描述。一是其静态属性——关于对象性质、状态的信息,用“数据”来描述;一是其能动属性——对象的行为,用“方法”来描述。学习面向对象的程序设计,首先要掌握这两个方面的内容:

- 对象的数据描述和设计;
- 对象的方法描述和设计。

现实世界中的对象无穷无尽,即使是在一个问题中也要涉及众多的对象。在面向对象的程序设计语言中,并不是去定义一个一个的对象,而是先对对象分类,定义对象的模板——类(class),再根据类创建对象。

下面举例说明如何用 Java 语言描述类。为了使初学者容易理解,例子中忽略了一些暂时不用的细节。

例 1.1 电子时钟的 Java 描述。

作为一个客观事物的电子时钟,有很多静态属性,如颜色、重量、耗电量等。在用对象去描述实际的电子时钟时,需要抽取能够反映电子时钟本质的属性信息——时间,进一步细化,用 hour(时)、minute(分)、second(秒)描述(当然,也可以只用“秒”来描述时间,“时”和“分”都可由“秒”来转换得到)。这样,hour(时)、minute(分)、second(秒)作为电子时钟类的属性,它们的值反映了电子时钟的具体状态。

那么,电子时钟的能动属性又有哪些呢?能动属性往往是对静态属性的操纵和控制的行为的描述,如显示时间,设置时间等,它们的描述如下:

- getMinute ()——显示“分”;

- getHour ()——显示“时”;
- setSecond ()——设置“秒”;
- setMinute ()——设置“分”;
- setHour ()——设置“时”。

这样,电子时钟 Java 描述的类的界面如图 1.1 所示。

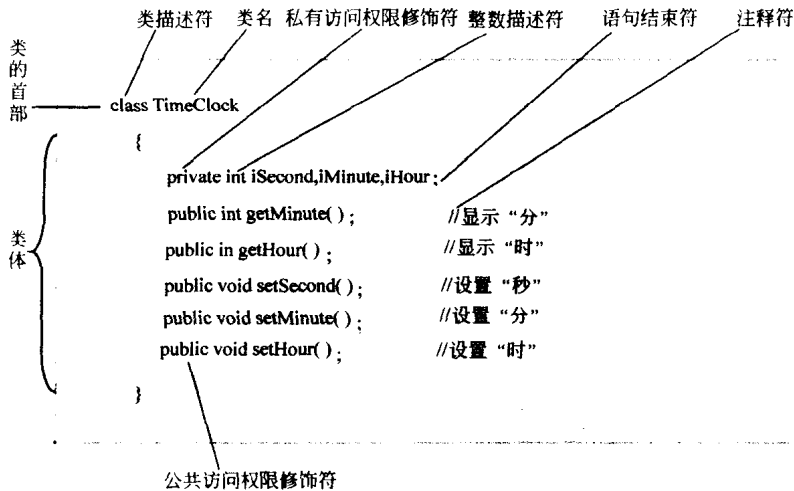


图 1.1 用 Java 描述的 TimeClock 类的界面

说明:

① 从结构上看,类由类的首部和类体构成。类的首部用关键字 class 开始,在最简单的情况下后面只是该类的名字。关键字是语言设定的固定用词,具有专门的意义和作用,程序员不能改变其拼法或将其作为类名、变量名或方法名等语法成分的名字;类名可由程序员自己按一定的格式和需要命名。本例中使用复合词 TimeClock 来作为时钟类的名字。

② 类体由紧随类的首部的一对花括号及花括号中的语句序列组成。语句是语言的具有完整陈述意义的基本单位。在 Java 语言中,每个语句用一个分号“;”结束。本例中类体中共有 6 个语句。

③ 类体中含有类的数据部分和类的方法部分,分别称为类的成员变量和成员方法,它们都是类的成员。就像一个家庭中的财产、事情和行为有隐私部分,也有公开部分一样,类的成员的可访问性由其前的访问权限修饰符描述:

- private: 此成员只有本类中的方法可以访问,说明这部分成员(信息)是类的隐私;
- public: 此成员可由任何 Java 包中的任何类访问,说明这部分成员(行为)外界可以使用。

这是两种最基本的访问权限修饰符。关于 Java 包和其他访问权限修饰符将在以后介绍。

④ 方法由方法首部和方法体组成。方法首部是方法的用户界面,含有方法名、方法

与其他方法的通信方式等信息;方法体用于描述如何实现方法所执行的操作。这里暂时不考虑这些方法的具体实现。

一个方法只有被其他方法调用时才执行其规定的行为操作。于是出现了两个方法之间的通信问题:

- 在调用时,调用者可以向被调用的方法传递一些数据——称为参数,参数放在方法名后面的圆括号中。如果没有参数传递,就使用空的圆括号,因为这里的圆括号除了放参数以后,还表示其前面的名字是一个方法名,本例中的方法都不接收调用者的参数。
- 有些方法执行后要向调用者(不是标准输出)送回一个数据,这样的数据称为方法的返回值。对这种方法应说明其返回值是什么类型(整数型、实数型等)。本例中,方法 `getMinute()` 和 `getHour()` 的返回值分别是“分”和“时”,用整数表示较为合适,所以把它们都说明为“int”类型;`setSecond()`、`setMinute()` 和 `setHour()` 只用于设置“秒”、“分”和“时”,不送回任何数据,用“void”加以说明,表示方法没有返回值。

⑤ 注释是对程序、程序的局部或一个语句的说明,让看(读)程序的人理解程序设计者的意图。注释不影响程序的逻辑功能。为了不使注释与程序的代码相混淆,需要按一定的格式用注释符将注释部分与程序的代码部分区分开。Java 允许使用以下 3 种注释格式。

- 在“//”后面写注释,只能写在一行中,并且后面不能再写任何程序代码,称单行注释。
- 在“/*”和“*/”之间写的注释,可以占用几行,并且后面可以再写程序代码。如

```
/* This is a Application.  
The name of this program  
is Simple Program. */
```

- 在“/**”和“*/”写注释是 Java 语言特有的注释方式。这种注释主要是为支持 JDK 工具 Javadoc 而设置的。如

```
/** This is a Application WelcomeApp.  
Program's name is WelcomeApp.java */
```

⑥ 这个 `TimeClock` 是一个时钟类的界面,当此界面中所有方法的方法体都定义后,就构成了一个时钟类,由这个时钟类可以创建多个时钟对象。

⑦ Java 是区分大小写字母的语言,`class`、`CLASS` 与 `Class` 在 Java 里面代表不同的含义。

⑧ 程序员可以自己定义类,也可以使用事先已经定义好的类。Java 中,系统开发者为了方便用户,已设计好了大量的类,用户可以直接使用这些类(见第 5 章,Java 资源库)。

⑨ 本例只是电子时钟类的设计的一个例子,它不是惟一的。在不同的应用环境中,可以根据需要设计出不同的电子时钟类。其次,本例中给出的也只是电子时钟类的一个

界面。类的界面是类的使用者和类的实现者之间的一个“协议”。从使用者的角度看,类的界面给出了这个类具有哪些功能的信息。从实现者的角度看,类的界面给出了必须提供哪些功能的信息。设计类往往从设计类的界面开始。

例 1.1 仅仅给出了一个类的界面设计的基本知识,它还不是一个完整的程序。一个完整的程序不仅要给出其所用到的每个类的界面定义,还要给出所有类方法的实现,并要定义一个主类来描述有关类对象的生成及其各对象间的关系。请看下例。

例 1.2 计算圆的周长和面积。

本题讨论的对象是圆。为此要设计一个类 Circle 和一个用于对类 Circle 进行测试的主类 UseCircle。

(1) 类 Circle——圆的定义

圆的定义如以下代码所示。注意它的方法中已包含了实现代码。

```

class Circle
{
    private float r;                // 半径
    private final double PI = 3.14159; // 最终变量
    public Circle(float x)
    {                                // 构造方法
        r = x;
    }

    public double circum ( )
    {                                // 求圆周长的方法
        return 2 * PI * r;
    }

    public double area( )
    {                                // 求圆面积的方法
        return PI * r * r;
    }
}
    
```

说明:

① Circle 类中含有两个数据成员,即半径 r 和一个常数 PI。注意这两个数的类型的不同:r 使用了 float 型,PI 使用了 double 型。表 1.1 所示是对 Java 中的最为常用的数据类型 int、float 和 double 的比较。

表 1.1 int、float 和 double 3 种数据类型的比较

数据类型名(关键字)	占用内存空间	数值范围	十进制数值精度
int	4 字节	$-2^{31} \sim 2^{31} - 1$	
float	4 字节	($\pm 3.4028235E+38$)~ ($\pm 1.40239846E-45$)	6~7 位
double	8 字节	($\pm 1.79769313486231570E+308$)~ ($\pm 4.94065645841246544E-324$)	15 位

选取不同的数据类型,会对计算结果的精度和程序的效率产生影响。因此,为了满足一定的需求,要根据不同类型所占用的内存大小(如果占用内存过大,除了会消耗更多的存储空间之外,往往还将影响运算速度)、取数的范围和数值的精度等方面来选取合适的数据类型。

② 众所周知,PI=3.14159 是一个不能修改的常量,因此在对 PI 的说明前面,除了数据类型说明之外,还使用了一个关键字“final”。“final”表示后面的量在程序执行过程中其值不能再变化,这类量称为最终变量。最终变量在程序中一般用大写。r 的前面无“final”修饰符,表示在程序执行过程中,r 的值是可以改变的,r 称为“变量”。变量像一个放东西的盒子,在程序执行过程中,其内容可以由程序进行更改。

③ Circle 类中定义了两个方法:circum()和 area()分别用于计算圆的周长和面积。

方法被调用执行后,往往要向调用者返回一个数据。如某个对象要用方法 circum()计算一个圆的周长时,就可以调用它。调用的目的是要通过它得到一个圆的周长。换言之,circum()被调用的结果是向调用它的对象返回一个圆的周长。在方法中,要使用语句 return 返回结果。return 语句返回的数据的类型称为方法的类型。本例中 circum()和 area()都向调用者返回 double 类型的数据。

④ 在 circum()方法中使用的符号“*”是乘法运算符。Java 基本的运算符有:

- +(加),-(减或负),*(乘),/(除);
- >(大于),>=(大于等于),==(等于),<(小于),<=(小于等于),!=(不等于);
- = 称为赋值号,意即将其右面的值送到左面的变量中,相当于往一个内存单元中存放入一个数据值。

由操作数和运算符按一定的语法形式组成的有运算意义的符号序列就叫做表达式。

⑤ 在 Circle 类中还定义了一个特殊的方法——类的构造方法。类的构造方法与类同名,描述由类生成对象的操作,在创建对象时由系统调用。下面通过主类定义介绍构造方法的作用。

(2) 主类

Java 应用程序可以由一个或多个类组成,其中有且仅有一个主类。主类就是含有主方法 main()的类。Java 规定,主类名必须与主类所在的应用程序文件名相同。程序运行后,Java 首先执行主类主方法中的第一个语句。本题中的主类用来测试 Circle 类。所谓测试 Circle 类,就是运行 Circle 类,观察分析其运行结果。下面是主类 UseCircle 的定义。

```
class UseCircle
{
    public static void main(String args[ ])
    {
        Circle aCirc = new Circle(5.3f);           // 主方法及其定义
                                                    // 生成一个圆对象

        double s = aCirc.circum();                // 计算该圆对象的周长
        double d = aCirc.area();                  // 计算该圆对象的面积
        System.out.println("半径为:" + aCirc.r + "的圆的周长为:" + s); // 输出周长
```



```
System.out.println("半径为:" + aCirc.r + "的圆的面积为:" + d); // 输出面积
}
}
```

说明:

① UseCircle 类中,主方法首先用 new 命令来调用 Circle 类的构造方法 Circlec,以创建一个 Circle 类的对象 aCirc,同时将其成员 r 初始化为 5.3。结合构造方法 Circle() 的定义可以看出,主方法调用 Circle() 时,向其传送了一个数据 5.3 给参数 x,而 Circle() 在执行时,用赋值语句将 x 的值赋给成员变量 r。构造方法前面不使用返回类型。

② 主方法中用语法形式

对象名.成员名;

调用 Circle 类中的成员,计算该圆对象的周长和面积。圆点·称成员运算符,用于表明所使用的成员是哪个对象的成员。因为主方法还可以用另一个参数(如 1.23f)生成另一个圆对象(如 bCirc)。显然 bCirc.r 与 aCirc.r 所指的 r 是不一样的。

③ 调用系统方法 System.out.println() 输出结果,在 System.out.println() 方法进行输出时,对于用双引号括起的参数部分按照其字符串原样输出,对于未用双引号括起的参数部分(变量或对象的数据成员)先将其值转换成相应的字符串,然后再输出之,相邻的不同参数部分之间用“+”连接。

④ main() 前面的 static 表明它是一个静态方法,可以用类名直接调用。关于 String args[] 将在 1.2.1 节介绍。

通过以上两个例子,已经对面向对象的方法和 Java 程序有了一个感性认识,随着学习的深入,将会逐步加深对面向对象的方法和程序设计的理解。

1.1.3 Java 平台和虚拟机

平台是支持程序运行的软硬件环境。Java 平台是在 Windows, Linux 等系统平台上支持的一种 Java 程序开发平台,主要由 Java 虚拟机(JavaVM)和 Java 应用程序接口(JavaAPI)两部分组成。Java 虚拟机易于被移植到基于不同硬件的平台上,是 Java 平台的基础; Java 应用程序接口由大量已做好的 Java 组件(组件是一种类)构成,这个接口提供了丰富的 Java 资源。

一种计算机语言就是一种程序设计工具。而程序是计算机工作的依据,不同的程序描述了计算机不同的工作方式和过程。尽管高级语言已经不像低级语言那样完全依赖于计算机的硬件系统,但是由于用高级语言编写的程序毕竟是适合人阅读的而不是机器可以直接识别和理解的语言,要让机器理解并执行,还必须将高级语言写的程序(一般叫做“源程序”)翻译成用机器语言表示的程序。由于不同的机器有不同的机器语言,并且由于操作系统环境的局限性,在一种平台上开发的程序运行到另一个平台上时,多少还要进行一些修改,这样就比较麻烦。

Java 破解各机器使用不同的机器语言的策略是定义出自己的一套虚拟机器——Java 虚拟机(Java VM, Java Virtual Machine)。为了说明 Java 虚拟机的工作原理,先来看图