

新世纪

计算机基础教育丛书

丛书主编

谭浩强

Java程序设计

题解与上机指导

辛运伟 饶一梅 秦晓东 编著



清华大学出版社

新世纪
计算机基础教育丛书

丛书主编
谭 浩 强

Java程序设计 题解与上机指导

编 著
辛运伟 饶一梅 秦晓东

清华大学出版社
北 京

内 容 简 介

本书是和《Java 程序设计》(辛运韩 饶一梅 张钧 编著,清华大学出版社)一书配套使用的参考书。本书对《Java 程序设计》一书中各章的全部习题进行解答,对所有的编程题目都给出了相应的程序代码,并配有程序执行结果。另外,本书增加了第 15 章,其中的题目都是比较实用的,目的是帮助读者进一步熟悉 Java 语言。

通过学习《Java 程序设计》,并配合使用本书,能使读者更深入地了解 Java 语言,同时也能使用该语言编程,以完成特定任务。

本书概念清晰、实用性强,可供学习 Java 语言的读者参考使用。

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。

图书在版编目(CIP)数据

Java 程序设计题解与上机指导/辛运韩,饶一梅,秦晓东编著. —北京:清华大学出版社,2003

(新世纪计算机基础教育丛书/谭浩强主编)

ISBN 7-302-06719-8

I. J… II. ①辛… ②饶… ③秦… III. Java 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 045587 号

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.com.cn>

责任编辑: 焦虹 付宇光

印 刷 者: 北京四季青印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 10 字数: 229 千字

版 次: 2003 年 7 月第 1 版 2003 年 7 月第 1 次印刷

书 号: ISBN 7-302-06719-8/TP·5011

印 数: 1~5000

定 价: 15.00 元

21世纪终于来临了,在新的世纪,人们自然对未来有许多美好的愿望和设想。现代科学技术的飞速发展,改变了世界,也改变了人类的生活。作为新世纪的大学生,应当站在时代发展的前列,掌握现代科学技术知识,调整自己的知识结构和能力结构,以适应社会发展的要求。新世纪需要具有丰富的现代科学知识、能够独立解决面临的任務、充满活力、有创新意识新型人才。

掌握计算机知识和应用无疑是培养新型人才的一个重要环节。计算机既是现代科学技术的结晶,又是大众化的工具。学习计算机知识不仅是为了掌握一种技能,更重要的是:它能启发人们对先进科技的向往,激发创新意识,推动对新知识的学习,培养自学能力,锻炼动手实践的本领。因而它是高等学校全面素质教育中极为重要的一部分。

自20世纪80年代初以来,高等学校中计算机教育(尤其是非计算机专业中的计算机教育)发展迅速,计算机教育的内容不断扩展,程度不断提高,它所起的作用也愈来愈显著。

在实践中,大家已认识到,计算机应用人才队伍是由两部分人组成的:一部分是计算机专业出身的计算机专业人才,他们是计算机应用人才队伍中的骨干力量;另一部分是各行各业中应用计算机的人员。这后一部分人一般并非从计算机专业毕业,他们人数众多,既熟悉自己所从事的专业,又掌握计算机的应用知识,善于用计算机作为工具去解决本领域中的任务。他们是计算机应用人才队伍中的基本力量。事实上,大部分应用软件都是由非计算机专业出身的计算机应用人员研制的。他们具有的这个优势是其他人难以代替的。从这个事实可以看到在非计算机专业中深入进行计算机教育的必要性。

非计算机专业中的计算机教育,无论目的、内容、教学体系、教材、教学方法等各方面都与计算机专业有很大的不同,决不应该照搬计算机专业的模式和做法。全国高等院校计算机基础教育研究会自1984年成立以来,始终不渝地探索高校计算机基础教育的特点和规律,在20世纪80年代中期,最早提出了按层次进行教育的方案。计算机应用是分层次的,不同的人在不同的层次上使用着计算机;同样,计算机教育也是分层次的,以适应不同应用层次的要求。全国有一千多所高等学校,好几百个专

业,学校的类型、条件和基础差别很大,不可能按同一模式、同一要求、同一内容进行教学。按层次组织教学,可以使不同专业、不同学校根据自己的情况选择教学内容,做到“各取所需”。

经过十多年的实践,几经调整,许多高校形成了按以下三个层次组织教学的方案:第一层次为计算机公共基础,学习计算机基本知识和基本操作;第二层次为计算机技术基础,内容包括程序设计、数据库、网络和多媒体技术等;第三层次为计算机应用基础,结合专业应用的需要学习有关计算机应用课程。每一层次中设立若干门课程,包括必修课和选修课。

1988年起,我们根据层次教学的方案,组织编写了“计算机基础教育丛书”,邀请有丰富教学经验的专家学者先后编写了20多种教材,由清华大学出版社出版。丛书出版后,迅速受到广大高校师生的欢迎,对高等学校的计算机基础教育起了积极的推动作用。广大读者反映这套教材定位准确、内容丰富、通俗易懂,符合广大非计算机专业学生的特点。许多高校都采用了我们编写的教材。丛书总发行量达到700多万册,这在全国是罕见的。

在新世纪来临之际,我们在该丛书成功的基础上组织编写了这套“新世纪计算机基础教育丛书”,以适应新形势的要求。本丛书有以下特点:

(1) 内容新颖。根据新世纪的需要,重新确定丛书的内容,以符合计算机科学技术的发展和教学改革的要求。本丛书除保留了原丛书中经过实践考验,且深受群众欢迎的优秀教材外,还编写了许多新的教材,在这些教材中反映了近年来迅速得到推广应用的一些计算机新技术,以后还将根据发展不断补充新的内容。

(2) 适合按层次组织教学的需要。在新世纪大多数学校是采用层次教学模式的,但不同的学校和专业所达到的层次不同。本丛书采用模块形式,提供了各种课程的教材,内容覆盖高校计算机基础教育的三个层次。丛书中既有供理工类专业用的教材,也有供文科和经济类专业用的教材;既有必修课的教材,也包括一些选修课的教材。各类学校都可以从中选择到合适的教材。

(3) 符合大学非计算机专业学生的特点。本丛书针对非计算机专业学生的特点,以应用为目的,以应用为出发点,强调实用性。本丛书的作者都是长期在一线从事高校计算机基础教育的教授和副教授,对学生的基础、特点和认识规律有深入的研究,在教学实践中积累了丰富的经验,可以说,每一本教材都是他们长期教学经验的总结。在教材的写法上,既注意概念的严谨和清晰,又特别注意采用读者容易理解的方法阐明看似深奥难懂的问题,做到例题丰富、通俗易懂、便于自学。这一点是本丛书一个十分重要的特点。书是写给读者看的,读者如果看不懂,只能算

写作的失败。

(4) 采用多样化的形式。除了文字教材这一基本形式外,有些教材还配有习题解答和上机指导。我们还准备采用现代教学方式,陆续制作电子出版物,以利于学生自学。

总之,本丛书的指导思想是:内容新颖、概念清晰、实用性强、通俗易懂、层次配套。简单概括为:“新颖、清晰、实用、通俗、配套”。我们经过多年实践形成的这一套行之有效的创作风格相信会受到广大读者的欢迎。判别一本书的优劣,读者最有发言权。

本丛书多年来得到了各方面人士的指导、支持和帮助,尤其是得到了全国高等院校计算机基础教育研究会的各位专家和各高校老师们的支持和帮助,我们在此表示由衷的感谢。

本丛书肯定会有不足之处,竭诚希望得到广大读者的批评指正。

丛书主编

全国高等院校计算机基础教育研究会理事长

谭浩强

2000年1月1日

前言

Forced Forced Forced Forced

现 在 Java 语言越来越受到人们的重视,学习和使用 Java 语言的人愈来愈多。为了使广大读者,特别是 Java 语言的自学者更快、更准确地理解、学习 Java,我们编写了这本《Java 程序设计题解与上机指导》,专门配合《Java 程序设计》(辛运韩 饶一梅 张钧 编著,清华大学出版社)一书的使用。

本书对《Java 程序设计》一书中各章的全部习题作了完整解答。这些解答只是参考,希望能起到抛砖引玉的作用。对所有的编程题目都给出了相应的程序代码,并在 Windows 平台的 Java 1.4.1 环境下调试通过,在习题中也给出了程序的执行结果。请读者注意,所有的答案和代码都不是惟一的,读者可以参考本书中的内容和其他参考书中的内容得出更全面的答案。至于程序代码,其实现的方式就更加多种多样。相信读者能在本书的代码基础之上,能够编写出功能更全面、效率更高的程序。

在《Java 程序设计》中的习题的基础上,本书增加了第 15 章,其中的题目涉及了比较实用的内容,目的是帮助读者进一步熟悉 Java 语言。

计算机技术是不断发展、不断完善的技术,Java 语言也是如此。从诞生之日起,它的版本就一直在更新。在《Java 程序设计》一书出版之后,Java 又有新版本面世,在本书编写时的最新版本是 Java 1.4.1。新版本对原版本中的某些概念和实现方式都有所改进。例如 AWT 事件模型由层次事件模型转变为委托事件模型,新增添了 javax.swing 组件库,在很多类中的方法都有所改变等。为了保持本书与《Java 程序设计》一书的一致性,对原书中的题目,哪怕其中涉及的概念已不再使用,本书仍作出全部解答。与此同时,本书还详尽介绍了新版本中对应的新概念,以帮助读者了解它们的延续性。此外,本书中的程序代码都是在最新的 Java 1.4.1 环境下实现的,并将该版本不再推荐使用的旧方法全部改为推荐使用的新方法。

本书是一本教学参考书。希望读者在使用、调试本书中的代码的同时,既能加深对 Java 语言的理解,又能提高程序设计的能力,并在此过程中不断发现问题、思考问题、解决问题,把本书作为掌握知识的一个工具和桥梁。

本书由辛运伟编写,饶一梅、秦晓东实现了部分程序代码。由于作者水平有限,对 Java 语言的掌握不够全面,书中难免有错误和不妥之处,恳请广大读者特别是同行专家批评指正。

编 者

2003 年 6 月

目 录

Catalog Catalog Catalog Catalog

第 1 章	概述	1
第 2 章	标识符、关键字和数据类型	6
第 3 章	表达式和流控制	14
第 4 章	数组和字符串	21
第 5 章	对象和类的进一步介绍	28
第 6 章	Java 语言中的异常	35
第 7 章	Java 的图形用户界面设计	40
第 8 章	AWT 事件模型	54
第 9 章	AWT 组件库	63
第 10 章	Java Applet	85
第 11 章	Java 语言的深入理解	92
第 12 章	Java 数据流	98
第 13 章	线程	111
第 14 章	Java 的网络功能	116
第 15 章	综合练习	123

第1章 概 述

1.1 Java 语言是由哪个公司推出的? 简述 Java 语言的特点。

解: Java 语言是美国 Sun Microsystem 公司推出的新一代面向对象程序设计语言, 特别适合于 Internet 应用程序开发, 自诞生之日起, 就受到了全世界的关注。

它有如下显著的特点:

(1) 语法简单, 功能强大

Java 语言的语法非常像 C++, 但它删去了 C++ 中不常用且容易出错的地方。如, Java 语言中没有指针、结构等概念, 没有预处理器, 程序员不必自己释放占用的内存空间, 因此在一定程度上减少了因内存混乱而导致的系统崩溃。另外, Java 语言强调其面向对象的特性, 用它可以编制出非常复杂的系统。虽然 Java 语言具备强大的功能, 但其解释器只占用了很少的内存, 因此适合在各种类型的机器上运行。

(2) 分布式与安全性

Java 语言强调网络特性, 内置了 TCP/IP、HTTP、FTP 协议类库, 具备强大且易于使用的联网能力, 便于开发网上应用系统。

Java 程序的三级代码安全检查机制可以有效地防止非法代码的侵入, 阻止对内存的越权访问, 避免病毒的侵害, 因此它成为了 Internet 上最安全的技术之一。

(3) 与平台无关

提到 Java, 有一句著名的口号: 一次编写, 到处运行。Java 语言规定了统一的数据类型。Java 编译器将 Java 程序编译成二进制代码, 即字节码(bytecode)。运行时环境针对不同的处理器指令系统, 把字节码转换为不同的具体指令, 因此 Java 可以跨平台使用, 特别适合于网络应用, 同时也为 Java 程序跨平台的无缝移植提供了很大便利。

(4) 解释运行, 高效率

Java 程序经编译器编译, 生成的字节码经过了精心设计, 并进行了优化, 因此运行速度较快, 克服了以往解释性语言运行效率低的缺点。在最新的 Java 版本中又加入了编译功能, 使得执行效率大幅度提高, 基本达到了编译语言的水平。

(5) 多线程

Java 内置了语言级多线程功能, 可使用户程序并行执行。Java 提供的同步机制可保证各线程对共享数据进行正确操作, 完成各自的特定任务。在硬件条件允许的情况下, 这些线程可以直接对应到各个 CPU 上, 充分发挥硬件性能, 以减少用户等待时间。

(6) 动态执行

Java 执行代码是在运行时动态载入的, 程序可以自动进行版本升级。在网络环境下, 可用于瘦客户机架构, 以减少维护工作。另外, 类库中增加的新方法和其他实例不会影响原有程序的执行。

(7) 丰富的 API 文档和类库

Java 为用户提供了详尽的 API 文档说明。Java 开发工具包中的类库包罗万象、应有尽有,程序员的开发工作可以在较高层次上展开。这也正是 Java 语言受欢迎的重要原因之一。

1.2 什么是 Java 虚拟机? 它包括哪几部分?

解: Java 虚拟机(Java virtual machine, JVM)规范中给出了它的定义: JVM 是在一台真正的机器上用软件方式实现的一台假想机。Java 虚拟机是运行 Java 程序必不可少的机制,它是编译后的 Java 程序和硬件系统之间的接口,程序员可以把 JVM 看作一个虚拟的处理器。编译后的 Java 程序指令并不直接在硬件系统的 CPU 上执行,而是由 JVM 执行。它不仅解释执行编译后的 Java 指令,而且还进行安全检查。它是 Java 程序能在多平台间进行无缝移植的可靠保证,同时也是 Java 程序的安全检验引擎。

JVM 的具体实现包括: 指令集(等价于 CPU 的指令集)、寄存器组、类文件格式、栈、垃圾收集堆、内存区。

1.3 简述 JVM 的工作机制。

解: JVM(Java 虚拟机)是运行 Java 程序必不可少的机制。编译后的 Java 程序指令并不直接在硬件系统的 CPU 上执行,而是由 JVM 执行。JVM 是编译后的 Java 程序和硬件系统之间的接口,程序员可以把 JVM 看作一个虚拟的处理器。它不仅解释执行编译后的 Java 指令,而且还进行安全检查。它是 Java 程序能在多平台间进行无缝移植的可靠保证,也是 Java 程序的安全检验引擎。

JVM 是在一台真正的机器上用软件方式实现的一台假想机。JVM 使用的代码存储在 .class 文件中。JVM 的部分指令很像真正的 CPU 指令,包括算术运算、流控制和数组元素访问等。

Java 虚拟机规范提供了编译所有 Java 代码的硬件平台。因为编译是针对假想机的,所以该规范能让 Java 程序独立于平台。它适用于每个具体的硬件平台,以保证为 JVM 编译的代码的运行。JVM 不但可以用软件实现,而且可以用硬件实现。

JVM 的代码格式为压缩的字节码,效率较高。由 JVM 字节码表示的程序必须保持原来的类型规定。Java 主要的类型检查是在编译时由字节码校验器完成。Java 的任何解释器必须能执行符合 JVM 定义的任何类格式的任何类文件。

Java 虚拟机规范对运行时数据区域的划分及字节码的优化并不作严格的限制,它们的实现依平台的不同而有所不同。

1.4 Java 的垃圾收集机制与其他语言相比有什么特点?

解: 许多程序设计语言允许在程序运行时动态分配内存,并将所分配的内存块的开始地址以指针的形式返回给程序,供程序员使用。一旦不再需要所分配的内存、程序或运行时环境最好将内存释放,避免内存越界时得到意外结果。

在 C 和 C++ (及其他许多语言)中,由程序开发人员负责内存的释放,但程序开发人员并不总是清楚内存应该在何时释放。而如果不及时释放不再需要的内存,或释放了仍在使用的内存,就可能导致程序崩溃或系统混乱。这些不能正确使用内存的程序被称为有“内存漏洞”。

在 Java 中,程序员不必亲自释放内存,它提供了后台系统级线程,记录每次内存分配

的情况,并统计每个内存指针的引用次数,引用次数为 0 则表示不再使用。在 Java 虚拟机运行时环境闲置时,垃圾收集线程将检查是否存在引用次数为 0 的内存指针。如果有,则垃圾收集线程将把该内存“标记”为“清除”(释放)。

在 Java 程序生存期内,垃圾收集将自动进行,无需用户释放内存,从而消除了内存漏洞。

1.5 上机调试《Java 程序设计》1.2 节中的程序 1-1,直到获得正确结果。

解:程序中,第 6 行代码将在屏幕上显示一条信息:

Hello World!

编译并执行程序,执行结果如图 1-1 所示。

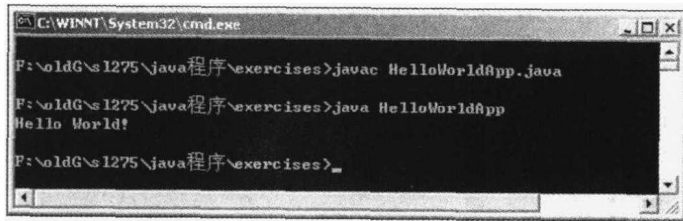


图 1-1 程序 1-1 的执行结果

1.6 练习使用浏览器查看 Java API 文档。

解:目前大多数读者使用的个人工作环境可能是 Windows 系列操作系统,包括 Windows 98、Windows NT、Windows 2000、Windows Me、Windows XP 等,所以本书中的解答以 Windows 环境为平台,并使用最新的 JDK Document 1.4 版本。

假设读者目前的系统中已安装有 JDK 1.4 文档,双击并启动它。此时出现如图 1-2 所示窗口。

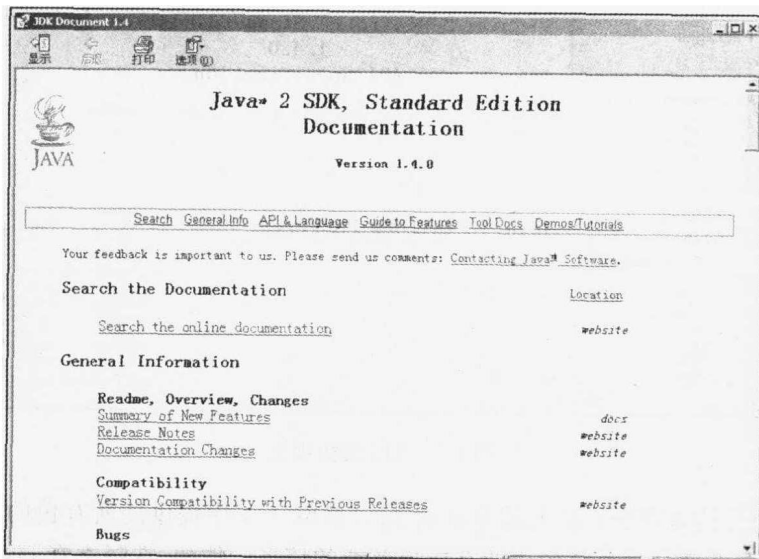


图 1-2 JDK 的初始界面

单击 API & Language, 进入 API & Language Documentation 窗口, 如图 1-3 所示。

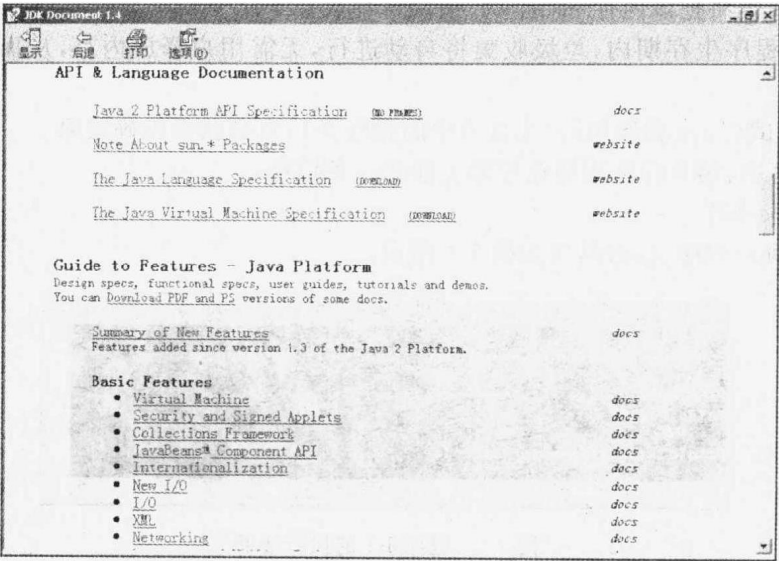


图 1-3 API 文档

单击 Java 2 Platform API Specification, 进入如图 1-4 所示的窗口。

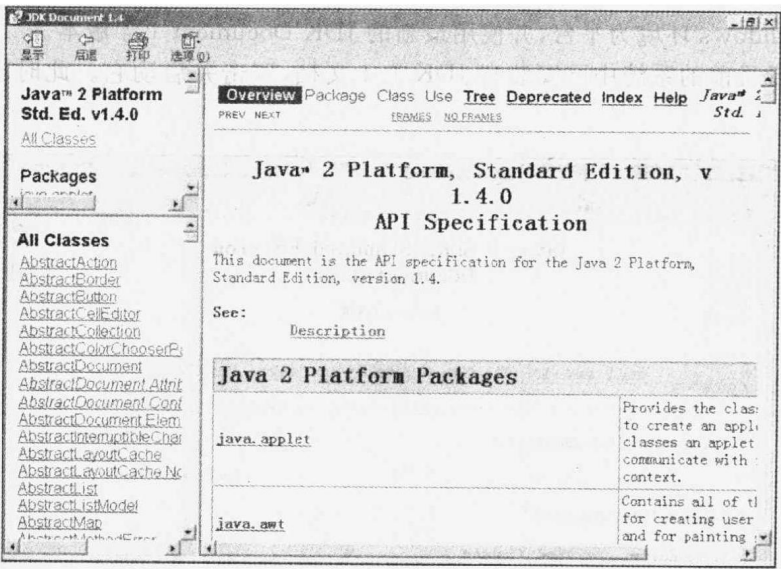


图 1-4 API 文档规范

这个窗口分为 3 部分：左上部分显示 Java JDK 1.4 中提供的所有包的信息, 选中某个包后, 将在左下部分显示该包中所有接口及类的信息。例如, 选择查看 java.lang 包, 窗口显示的内容如图 1-5 所示。

如果想进一步查看包中 Integer 类的信息, 应选中 Integer, 右侧窗口部分将显示

java.lang 中 Integer 类的所有接口及类的内容, 向下拉动滚卷条, 定位到所需内容的位置就可以了。

一般地, 一个类中的信息包括以下几部分:

- Field Summary
- Constructor Summary
- Method Summary
- Field Detail
- Constructor Detail
- Method Detail

Field Summary 中列出了类中成员变量的信息, 包括名称、类型及含义。Field Detail 中将详细介绍这些成员变量。

Constructor Summary 中列出类构造方法的信息 (包括参数列表), 并解释所创建的实例。构造方法的详细信息显示在 Constructor Detail 部分中。

在 Method Summary 中可以查找到要使用的方法名, 在 Method Detail 中将详细介绍该方法的使用, 包括调用参数表及返回值情况。例如, 图 1-6 所示的窗口中显示的是 toString 方法的详细情况, 该方法将返回整数所对应的字符串。

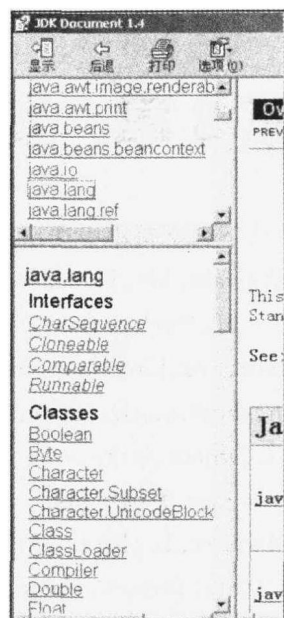


图 1-5 java.lang 包内信息列表

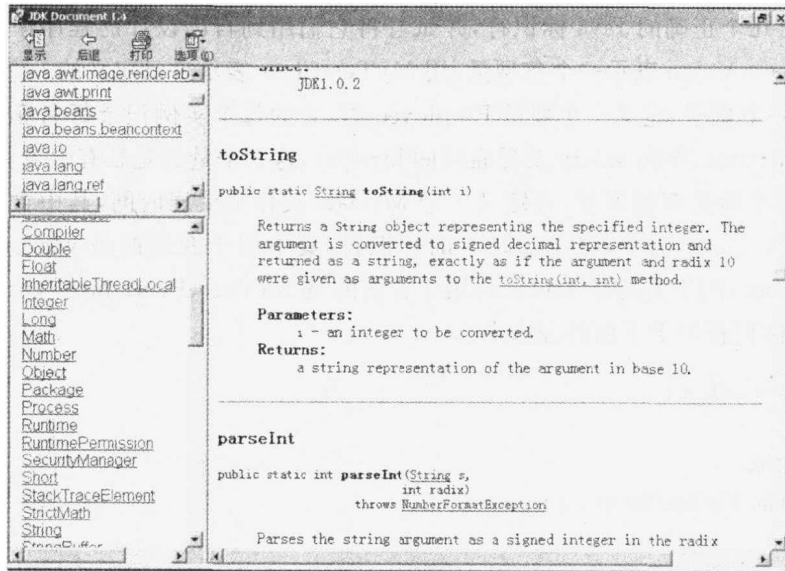


图 1-6 toString 方法的介绍

第2章 标识符、关键字和数据类型

2.1 从下列字符串中选出正确的 Java 关键字：

abstract、bit、boolean、case、character、comment、double、else、end、endif、extend、false、final、finally、float、for、generic、goto、if、implements、import、inner、instanceof、interface、line、long、loop、native、new、null、old、oper、outer、package、print、private、rest、return、short、static、super、switch、synchronized、this、throw、throws、transient、var、void、volatile、where、write。

解：Java 关键字包括：

abstract、boolean、case、double、else、false、final、finally、float、for、generic、goto、if、implements、import、inner、instanceof、interface、line、long、native、new、null、outer、package、private、rest、return、short、static、super、switch、synchronized、this、throw、throws、transient、var、void、volatile。

不是 Java 关键字的有：

bit、character、comment、end、endif、extend、line、loop、old、oper、print、where、write。

2.2 请写出几个正确的 Java 标识符，并试着将它们用到自己设计的程序中。例如用变量 IntegerValue 表示一个整型量，用 MyTestClass 表示自己定义的一个类等。

解：举一个例子，定义一个雇员 Employee 类，它的每个实例记录每名雇员的基本情况，包括姓名 name、年薪 salary 及受雇时间 hireDay，这 3 个量都是私有变量。其中，姓名是字符串型，年薪是双精度型，并定义一个 MyDate 类作为受雇时间，其中含有雇员参加工作的年、月、日。本例中，定义了以下的标识符，其中用于变量的是 year、month、day、staff、byPercent；用于类的是 MyDate；用于方法的是 SetYear、Print、hireYear。

将这些标识符用于下面的程序中。

```
import java.util.*;

class MyDate
{
    public MyDate(int y, int m, int d)
    {
        year = y;
        month = m;
        day = d;
    }
    public int getYear()
    {
        return year;
    }
}
```

```

    }
    private int year;
    private int month;
    private int day;
}

class Employee
{
    public Employee(String n, double s, MyDate d)
    {
        name = n;
        salary = s;
        hireDay = d;
    }
    public void print()
    {
        System.out.println(name + " " + salary + " " + hireYear());
    }
    public void raiseSalary(double byPercent)
    {
        salary *= 1 + byPercent / 100;
    }
    public int hireYear()
    {
        return hireDay.getYear();
    }
    private String name;
    private double salary;
    private MyDate hireDay;
}

public class EmployeeTest
{
    public static void main(String[] args)
    {
        Employee[] staff = new Employee[3];

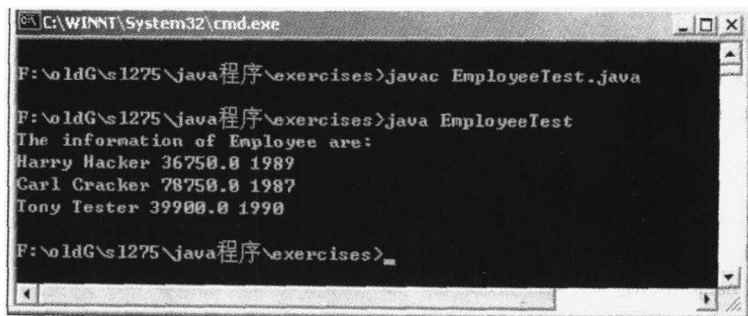
        staff[0] = new Employee("Harry Hacker", 35000, new MyDate(1989,10,1));
        staff[1] = new Employee("Carl Cracker", 75000, new MyDate(1987,12,15));
        staff[2] = new Employee("Tony Tester", 38000, new MyDate(1990,3,15));

        int i;
        System.out.println("The information of Employee are:");
        for (i = 0; i < 3; i++) staff[i].raiseSalary(5);
        for (i = 0; i < 3; i++) staff[i].print();
    }
}

```

使用命令行编译并运行该程序。程序中已经输入了 3 名雇员的基本信息,并按 5% 的比例涨年薪,则程序的输出如图 2-1 所示。

2.3 请叙述标识符的定义规则。指出在下面的标识符中,哪些是不正确的,并说明原因:



```
C:\WINNT\System32\cmd.exe
F:\oldG\s1275\java程序\exercises>javac EmployeeTest.java
F:\oldG\s1275\java程序\exercises>java EmployeeTest
The information of Employee are:
Harry Hacker 36750.0 1989
Carl Cracker 78750.0 1987
Tony Tester 39900.0 1990
F:\oldG\s1275\java程序\exercises>
```

图 2-1 EmployeeTest 的执行结果

here, _there, this, that, it, 2tol, _it。

解：变量、常量、方法、对象和类等需要使用一个名称来表示，这个名称就是标识符。就是说，标识符在程序中可用作变量名、方法名、接口名、类名等。一般而言，标识符由编程者自行定义，但要遵循相应的语法规则。在 Java 中，对于标识符的规则主要有：

- (1) 在 Java 语言中，标识符是以字母、下划线(_)或美元符号(\$)开头，由字母、数字、下划线(_)或美元符号(\$)组成的字符串；
- (2) 标识符区分大小写；
- (3) 标识符的长度没有限制；
- (4) 注释不能插在一个标识符或关键字中；
- (5) Java 有许多关键字，它们都有各自的特殊意义和用法，不得用它们作为标识符。标识符内可以包含关键字，但不能与关键字完全一样。如“thisOne”是一个合法的标识符，但“this”是关键字，不能当作标识符。

除以上所列几项之外，标识符中不能含有其他符号，例如+、=、*及%等，当然也不允许插入空格。实际上，Java 源代码使用的是 Unicode 码，而不是 ASCII 码。Unicode 码用 16 位表示一个字符，其字符集中的字符数可达 65 535 个，比通常使用的 ASCII 码字符集(通常只有 255 个)大得多。Unicode 兼容了许多不同的字母表，包括常见语种的字母。英文字母、数字和标点符号在 Unicode 和 ASCII 字符集中有相同的值。

在所给的标识符中，不正确的有：this、2tol。其中 this 是 Java 中的关键字，不能用作标识符；2tol 是以数字开头的字符串，也不符合 Java 对标识符的规定。

2.4 Java 中共有哪些基本数据类型，它们分别用什么符号来表示，各自的取值范围是多大？试着对每种数据类型定义一个变量，并给它赋一个值。

解：Java 中定义的各种数据类型占用固定的内存长度，与当前系统的软硬件环境无关。例如，int 型肯定是个 32 位的整数，而不管是在 UNIX、Windows 还是 Macintosh 环境下。这个特点体现了 Java 的平台无关性。另外 Java 还为每种数据类型都预定义了一个默认值，以保证在任何情况下对变量的取值都是正确的。这个特点体现了 Java 的安全性。

Java 的数据类型共分为两大类，一类是基本类型，一类是复合数据类型。基本类型