

COMPUTER

高等院校计算机技术



“十二五”规划教材

Java 面向对象编程

◎ 主 编 吴育锋 周建东
副主编 罗江英 赵泽茂 卢 敏



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

Java面向对象编程

主 编 吴育锋 周建东
副主编 罗江英 赵泽茂 卢 敏



ZHEJIANG UNIVERSITY PRESS

浙江大学出版社

图书在版编目 (CIP) 数据

Java 面向对象编程 / 吴育锋等主编. — 杭州: 浙江大学出版社, 2015.6

ISBN 978-7-308-14906-8

I. ① J… II. ① 吴… III. ① JAVA 语言—程序设计
IV. ① TP312

中国版本图书馆 CIP 数据核字 (2015) 第 168412 号

内容提要

本书从初学者的角度出发, 以培养读者 Java 编程实践能力为目的来设计教材内容。按照“任务驱动”的思想设计了新颖的框架结构, 并按每个任务的需求精心组织知识点。全书共 10 章, 内容涵盖了 Java 开发环境的搭建与运行机制、基本语法、面向对象编程思想、GUI 编程、I/O、多线程技术、网络编程和 JDBC 技术等 Java 开发中重点用到的多种知识点和技术。每章都以一个任务描述作为开头, 继而进行任务分析, 技术准备 (知识点学习), 然后提出解决方案, 最终进行任务实现。在技术准备环节, 本书采用了大量典型的例子, 通过剖析案例、分析代码, 帮助读者养成良好的编程习惯。这一系列环节构成了一个整体, 展示了相关知识的综合运用, 最终加深读者对知识的理解, 达到熟练掌握相关技术要点的目的。

本书可作为高等院校本、专科计算机类、软件工程类、信息类相关专业教材, 也可以作为对 Java 语言感兴趣的读者的自学用书。

Java 面向对象编程

主编 吴育锋 周建东

责任编辑 吴昌雷

责任校对 陈慧慧 丁佳雯

封面设计 刘依群

出版发行 浙江大学出版社

(杭州市天目山路 148 号 邮政编码 310007)

(网址: <http://www.zjupress.com>)

排 版 杭州立飞图文制作有限公司

印 刷 杭州杭新印务有限公司

开 本 787mm×1092mm 1/16

印 张 20.5

字 数 504 千

版 次 2015 年 6 月第 1 版 2015 年 6 月第 1 次印刷

书 号 ISBN 978-7-308-14906-8

定 价 39.00 元

版权所有 翻印必究 印装差错 负责调换

浙江大学出版社发行部邮购电话 (0571)88925591; <http://zjdxcbbs.tmall.com>

高等院校计算机技术“十二五” 规划教材编委会

顾 问

李国杰 中国工程院院士,中国科学院计算技术研究所所长,浙江大学计算机学院院长

主 任

潘云鹤 中国工程院常务副院长,院士,计算机专家

副主任

陈 纯 浙江大学计算机学院常务副院长、软件学院院长,教授,浙江省首批特级专家

卢湘鸿 北京语言大学教授,教育部高等学校文科计算机基础教学指导委员会副主任

冯博琴 西安交通大学计算机教学实验中心主任,教授,2006—2010年教育部高等学校计算机基础课程教学指导委员会副主任委员,全国高校第一届国家级教学名师

何钦铭 浙江大学软件学院副院长,教授,2006—2010年教育部高等学校理工类计算机基础课程教学指导分委员会委员

委 员(按姓氏笔画排列)

马斌荣 首都医科大学教授,2006—2010年教育部高等学校医药类计算机基础课程教学指导分委员会副主任,北京市有突出贡献专家

石教英 浙江大学 CAD&CG 国家重点实验室学术委员会委员,浙江大学计算机学院教授,中国图像图形学会副理事长

刘甘娜 大连海事大学计算机学院教授,原教育部非计算机专业计算机课程教学指导分委员会委员

庄越挺 浙江大学计算机学院副院长,教授,2006—2010年教育部高等学校计算机科学与技术专业教学指导分委员会委员

- 许端清 浙江大学计算机学院教授
- 宋方敏 南京大学计算机系副主任,教授,2006—2010年教育部高等学校理工类计算机基础课程教学指导分委员会委员
- 张长海 吉林大学计算机学院副院长,教授,2006—2010年教育部高等学校理工类计算机基础课程教学指导分委员会委员
- 张 森 浙江大学教授,教育部高等学校文科计算机基础教学指导委员会副主任,全国高等院校计算机基础教育研究会副理事长
- 邹逢兴 国防科技大学教授,全国高校第一届国家级教学名师
- 陈志刚 中南大学信息学院副院长,教授,2006—2010年教育部高等学校计算机科学与技术专业教学指导分委员会委员
- 陈根才 浙江大学计算机学院副院长,教授,2006—2010年教育部高等学校农林类计算机基础课程教学指导分委员会委员
- 陈 越 浙江大学软件学院副院长,教授,2006—2010年教育部高等学校计算机科学与技术教学指导委员会软件工程专业教学指导分委员会委员
- 岳丽华 中国科学技术大学教授,中国计算机学会数据库专委会委员,2006—2010年教育部高等学校计算机科学与技术专业教学指导分委员会委员
- 耿卫东 浙江大学计算机学院教授,CAD&CG国家重点实验室副主任
- 鲁东明 浙江大学计算机学院教授,浙江大学网络与信息中心主任

序 言

在人类进入信息社会的 21 世纪,信息作为重要的开发性资源,与材料、能源共同构成了社会物质生活的三大资源。信息产业的发展水平已成为衡量一个国家现代化水平与综合国力的重要标志。随着各行各业信息化进程的不断加速,计算机应用技术作为信息产业基石的地位和作用得到普遍重视。一方面,高等教育中,以计算机技术为核心的信息技术已成为很多专业课教学内容的有机组成部分,计算机应用能力成为衡量大学生业务素质与能力的标志之一;另一方面,初等教育中信息技术课程的普及,使高校新生的计算机基本知识起点有所提高。因此,高校中的计算机基础教学课程如何有别于计算机专业课程,体现分层、分类的特点,突出不同专业对计算机应用需求的多样性,已成为高校计算机基础教学改革的重要内容。

浙江大学出版社及时把握时机,根据 2005 年教育部“非计算机专业计算机基础课程指导分委员会”发布的“关于进一步加强高等学校计算机基础教学的几点意见”以及“高等学校非计算机专业计算机基础课程教学基本要求”,针对“大学计算机基础”、“计算机程序设计基础”、“计算机硬件技术基础”、“数据库技术及应用”、“多媒体技术及应用”、“网络技术与应用”六门核心课程,组织编写了大学计算机基础教学的系列教材。

该系列教材编委会由国内计算机领域的院士与知名专家、教授组成,并且邀请了部分全国知名的计算机教育领域专家担任主审。浙江大学计算机学院各专业课程负责人、知名教授与博导牵头,组织有丰富教学经验和教材编写经验的教师参与了对教材大纲以及教材的编写工作。

该系列教材注重基本概念的介绍,在教材的整体框架设计上强调针对不同专业群体,体现不同专业类别的需求,突出计算机基础教学的应用性。同时,充分考虑了不同层次学校在人才培养目标上的差异,针对各门课程设计了面向不同对象的教材。除主教材外,还配有必要的配套实验教材、问题解答。教材内容丰富,体例新颖,通俗易懂,反映了作者们对大学计算机基础教学的最新探索与研究成果。

希望该系列教材的出版能有力地推动高校计算机基础教学课程内容的改革与发展,推动大学计算机基础教学的探索和创新,为计算机基础教学带来新的活力。

中国工程院院士
中国科学院计算技术研究所所长
浙江大学计算机学院院长



前 言

面向对象技术被认为是程序设计方法学的革命性突破,其思想是模拟现实世界的思维方式,使数据与操作相统一,使开发软件的方法与过程尽可能接近人类认识世界、解决问题的方法与过程。而 Java 语言正是面向对象技术成功应用在程序设计中的典范。

Java 程序设计语言具有面向对象、平台无关性、可移植性、健壮性和安全性等优良特性,同时 Java 技术对各个层次的分布应用(包括嵌入式应用、桌面应用与企业应用)都提供了全面系统的方法与技术。因此,目前 Java 已不仅仅是一种单纯的程序设计语言,而变成了一种支撑 Internet 应用的庞大技术体系,是一种开发软件的标准、规范与架构的统称。

在编写风格上,本书不追求对 Java 技术的全面介绍,而是根据项目需要,通过任务描述、任务分析、技术准备(知识点)、基本语法、解决方案、任务实现的递进方式适当讲解,并通过项目拓展的方式启发读者对相关知识的学习,符合人们对事物的认知规律。

在内容编排上,本书始终通过经典的实际项目来讲述面向对象和 Java 编程的基本思想、方法和技术,使读者通过项目的开发,巩固相关的内容,并掌握实际编程技能。教材内容以最新 Java 技术版本为背景,紧跟 Java 技术发展潮流。本书层次分明,对基本知识、技术论述详尽,又注重深度与广度的拓展,教学时可按需进行取舍,适应个性化教育的需要。

全书共分 10 章。第 1 章介绍 Java 语言的历史和特点,Java 程序运行环境的选择和搭建、程序的开发步骤等内容。第 2 章介绍 Java 语言的关键字、数据类型、运算符、表达式、选择结构、循环结构、数组等。第 3 章和第 4 章介绍面向对象技术中的对象、类、封装、继承、多态、接口及包等重要内容。第 5 章介绍 Java 语言中一些常用的类库,如 System 类、字符串处理类、Math 类、基本数据类型封装类、日期处理类和集合类等。第 6 章介绍图形用户界面的设计技术,包括图形界面开发基础、最新的 Swing 包、界面布局管理器、事件处理、菜单及对话框设计等。第 7 章介绍 Java 语言的输入、输出及文件操作功能的实现。第 8 章介绍 Java 特有的多线程技术,包括多线程的基本机制、线程的实现方法、等待和同步调度技术等。第 9 章主要介绍 Java 中强大的网络编程技术。第 10 章介绍数据库编程技术,包括数据库的基本概念、Java 数据库连接技术 JDBC 和数据库增删改查的编程实现。

本教材既可以作为计算机专业和非计算机专业的基础教材,也可以作为需要使用计算机的工程人员和科技工作者的自学参考书。本教材在编写与出版的过程中得到了许多朋友的帮助,这里一并表示诚挚的谢意。同时真诚地希望广大读者,特别是讲授此课程的老师对本教材提示意见和指正,以便再版时改进和完善本教材。

本书由吴育锋、周建东主编,罗江英等参加编写。因编写时间紧张,加之水平有限,书中疏漏之处,敬请广大读者批评指正。

目 录

第 1 章 Java 系统概述	1	【强化习题】	61
1.1 任务描述	1	第 3 章 面向对象基础	62
1.2 任务分析	1	3.1 任务描述	62
1.3 技术准备(知识点)	1	3.2 任务分析	62
1.3.1 Java 的起源和发展	1	3.3 技术准备(知识点)	63
1.3.2 Java 的特点	2	3.3.1 类与对象	63
1.3.3 Java 的运行原理	4	3.3.2 类的属性	67
1.3.4 Java 的开发工具——JDK	5	3.3.3 类的方法	69
1.3.5 第一个 Java 程序	7	3.3.4 构造方法	71
1.3.6 Eclipse 工具	9	3.3.5 封装性	73
1.4 解决方案	12	3.4 解决方案	76
1.5 任务实现	12	3.5 任务实现	77
【强化习题】	13	【强化习题】	79
第 2 章 Java 编程基础	14	第 4 章 继承与多态	80
2.1 任务描述	14	4.1 任务描述	80
2.2 任务分析	14	4.2 任务分析	80
2.3 技术准备(知识点)	15	4.3 技术准备(知识点)	85
2.3.1 Java 标识符	15	4.3.1 继承机制	85
2.3.2 常量和变量	16	4.3.2 抽象类	92
2.3.3 标准输入和标准输出	25	4.3.3 接口	95
2.3.4 运算符和表达式	25	4.3.4 多态性	98
2.3.5 选择结构	35	4.4 解决方案	101
2.3.6 循环结构	41	4.5 任务实现	101
2.3.7 break 和 continue	46	【强化习题】	104
2.3.8 数组	51	第 5 章 常用 Java API	105
2.4 解决方案	57	5.1 任务描述	105
2.5 任务实现	58		

5.2 任务分析	105	7.4 拓展知识	246
5.3 技术准备(知识点)	106	7.4.1 异常的概念	246
5.3.1 Object 类	106	7.4.2 Java 异常处理机制	246
5.3.2 System 类	107	7.4.3 throw 语句	250
5.3.3 String 类	109	7.4.4 自定义异常	251
5.3.4 StringBuffer 类	113	7.5 解决方案	252
5.3.5 Math 类	115	7.6 任务实现	253
5.3.6 基本数据类型封装类	116	【强化习题】	255
5.3.7 日期处理类	117	第 8 章 多线程技术	257
5.3.8 集合类	120	8.1 任务描述	257
5.4 解决方案	124	8.2 任务分析	257
5.5 任务实现	124	8.3 技术准备(知识点)	257
【强化习题】	126	8.3.1 什么是多线程	257
第 6 章 GUI 编程基础	127	8.3.2 创建线程	258
6.1 任务描述	127	8.3.3 线程的状态	260
6.2 任务分析	127	8.3.4 线程的优先级	260
6.3 技术准备(知识点)	128	8.3.5 线程的调度	261
6.3.1 GUI 概述	128	8.3.6 线程的同步控制	263
6.3.2 Swing 基本组件	129	8.3.7 死锁	267
6.3.3 菜单	162	8.4 解决方案	268
6.3.4 对话框	180	8.5 任务实现	269
6.3.5 布局管理器	197	【强化习题】	273
6.3.6 常用事件	208	第 9 章 网络编程基础	274
6.4 解决方案	216	9.1 任务描述	274
6.5 任务实现	217	9.2 任务分析	274
【强化习题】	222	9.3 技术准备(知识点)	275
第 7 章 输入输出	223	9.3.1 网络的概念	275
7.1 任务描述	223	9.3.2 网络通信模型	276
7.2 任务分析	223	9.3.3 网络通信方式	277
7.3 技术准备(知识点)	224	9.3.4 网络编程的步骤	278
7.3.1 标准输入/输出	224	9.3.5 Java 编程网络常用类	279
7.3.2 文件	225	9.3.6 基于 TCP 协议的网络编程	282
7.3.3 流的概述	228	9.3.7 UDP 协议的网络编程	284
7.3.4 节点流	229	9.4 解决方案	289
7.3.5 处理流	237	9.5 任务实现	289

【强化习题】	300	10.3.2 JDBC 常用类和接口	302
第 10 章 JDBC 技术简介	301	10.3.3 JDBC 应用程序的基本步骤 ...	306
10.1 任务描述	301	10.3.4 SQL 简介	308
10.2 任务分析	301	10.4 解决方案	309
10.3 技术准备(知识点)	301	10.5 任务实现	309
10.3.1 JDBC 简介	301	【强化习题】	314
		参考文献	315

第 1 章 Java 系统概述

1.1 任务描述

朋友小王从网络中下载了一个小游戏的源代码(FiveChessGame.java),只知道这个小游戏是用一种叫 Java 的程序语言开发的,但不知道如何才能运行起来,于是向你求助,请你帮助他将这个小游戏运行起来。

1.2 任务分析

要完成以上任务,必须解决以下问题:

1. Java 是一种什么样的程序设计语言,它有什么特点?
2. Java 这种程序设计语言和以前学过的 C 语言是不是一样,经过编译,连接后就能直接运行?
3. 如果 Java 的源代码需要经过编译才能运行,那么编译工具是什么,到哪里能找到 Java 源代码的编译工具?
4. 以前学习 C 语言,可以在 Visual C++ 6.0 和 Turbo C 这两种开发环境中编译和运行,Java 有没有类似的环境?如果有,Java 的编译和运行环境如何搭建?

1.3 技术准备(知识点)

1.3.1 Java 的起源和发展

Java 来自于 Sun 公司的一个叫 Green 的项目,其原先的目的是为家用电子消费产品开发一个分布式代码系统,该系统可以把 e-mail 发给电冰箱、电视机等家用电器,对它们进行控制,和它们进行信息交流。开始他们准备采用 C++实现这个系统,但 C++太复杂,安全性差。最后他们基于 C++开发了一种新语言 Oak(Java 的前身)。Oak 是一种用于网络的精巧而安全的语言,Sun 公司曾以此投标一个交互式电视项目,结果被 SGI 打败,于是 Oak 就被束之高阁了。1993 年 7 月,伊

利诺斯大学的 NCSA 推出了一个在 Internet 上广为流行的 WWW 浏览器 Mosaic 1.0 版。为了增强 WWW 的交互(B/S),还需要通过一种机制来使它具有动态性。受到 Mosaic 和 Netscape 的启发, Oak 项目组成员用 Java 编制了 HotJava 浏览器,得到了 Sun 公司首席执行官 Scott McNealy 的支持,揭开了 Java 进军 Internet 的序幕。Java 技术的发展历程简述如下:

1990 年, Sun 公司 James Gosling 领导的小组设计了一种平台独立的语言 Oak,主要用于为各种家用电器编写程序。

1994 年,由于 Sun 公司的创始人之一 Bill Joy 的介入, Oak 被改名为 Java。

1995 年 5 月 23 日, Sun 公司在 SunWorld'95 会议上正式发布 Java 和 HotJava 浏览器。

1996 年 1 月, Sun 公司宣布成立新的业务部门——JavaSoft 部,用以开发、销售并支持基于 Java 技术的产品,由 Alan Baratz 任总裁。公司同时推出 Java 开发工具包 JDK(Java Development Kit) 1.0,为开发人员提供用来编制 Java 应用软件所需的工具。

1996 年 2 月, Sun 公司发布了 Java 芯片系列,包括 PicoJava, MicroJava 和 UltraJava,并推出 Java 数据库连接——JDBC(Java Database Connectivity)。

1996 年 3 月, Sun 公司推出 Java WorkShop。

2010 年, Sun 公司被 Oracle 公司并购。

工业界一致认为:“Java 的出现是(20 世纪)80 年代以来计算机界的一件大事。”微软总裁 Bill Gates 也认为:“Java 是长期以来最卓越的程序设计语言。”

1.3.2 Java 的特点

Java 白皮书对 Java 语言的定义如下:

“Java: A simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multi-threaded, and dynamic language.”

Java 的特点可以归结为:

1. 简单(simple)

Java 的设计目的是让专业程序员觉得既易学又好用。假设用户有编程经历,将不觉得 Java 难掌握;如果已经理解面向对象编程的基本概念,学习 Java 将更容易;如果是一个经验丰富的 C++ 程序员,那就更好了,学习 Java 简直不费吹灰之力。因为 Java 继承了 C/C++ 的语法和许多 C++ 面向对象的特性,大多数程序员在学习 Java 时都不会觉得太难。另外, C 语言中许多容易混淆的概念(比如指针),要么被 Java 弃之不用了,要么以一种更清楚、更易理解的方式被实现。

2. 面向对象(object-oriented)

尽管受到其前辈的影响, Java 并没被设计成兼容其他语言源代码的程序语言。Java 语言可以更直接、更易用、更实际地接近对象。通过对近几十年来出现的面向对象软件的优点的借鉴, Java 设法在纯进化论者的“任何事物都是一个对象”和实用主义者的“不讨论对象不对象”的论点之间找到平衡。Java 的对象模型既简单又容易扩展。

3. 分布式(distributed)

Java 是为 Internet 的分布式环境而设计的,因此它可以处理 TCP/IP 协议。事实上,通过 URL 地址存取多个资源与直接存取一个文件的差别是不太大的。Java 原来的版本(Oak)包括了内置

的地址空格消息传递(intra-address-space)特性。这允许位于两台不同的计算机上的对象可以远程执行过程。Java 还发布了叫作远程方法调用(Remote Method Invocation,RMI)的软件包。这个特性使客户机/服务器编程达到了无与伦比的抽象级。

4. 解释型(interpreted)

Java 是一种解释型语言,通过把程序编译为 Java 字节码这样一个中间过程,产生跨平台运行的程序。字节码可以在提供 Java 虚拟机(JVM)的任何一种系统中被解释执行。

5. 健壮(robust)

使程序运行失败的两个主要原因是:内存管理错误和误操作引起的异常情况(也就是运行时错误)。在传统的编程环境下,内存管理是一项困难、乏味的任务。例如,在 C/C++中,程序员必须手工地分配并且释放所有的动态内存。这有时会导致问题,因为程序员可能忘记释放原来分配的内存,或者释放了其他部分程序正在使用的内存。Java 通过替用户管理内存分配和释放,可以从根本上消除这些问题(事实上,释放内存是完全自动的,因为 Java 为闲置的对象提供内存垃圾自动收集功能)。在传统的环境下,异常情况可能经常由“被零除”或“文件未找到”这样的情况引起,而我们又必须用既繁杂又难以理解的一大堆指令来对它们进行管理。Java 通过提供面向对象的异常处理机制来解决这个问题。对于一个写得很好的 Java 程序,所有的运行时的错误都可以并且应该由该程序自身进行管理。

6. 安全(secure)

Java 在网络应用程序为用户的计算机提供了一道防火墙(firewall),消除了用户的安全顾虑。当使用一个兼容 Java 的 Web 浏览器时,用户可以安全地下载 Java 小应用程序,而不必担心病毒的感染或恶意的攻击。Java 实现这种保护功能的方式是:将 Java 程序限制在 Java 运行环境中,不允许它访问计算机的其他部分。能下载小应用程序并能确保它对客户机不会造成危害是 Java 的安全性一个最重要的方面。

7. 结构中立(architecture-neutral)和可移植性(portable)

Java 设计者考虑的主要问题是程序代码的持久性和可移植性。程序员面临的一个主要问题是,不能保证今天编写的程序明天仍然能在同一台机器上顺利运行。操作系统升级、处理器升级以及核心系统资源的变化,都可能导致程序无法继续运行。Java 设计者对解决这个问题做过多种尝试。Java 虚拟机(JVM)的应用就是试图解决这个问题的。他们的目标是“只要写一次程序,在任何地方、任何时间该程序都能运行”。在一定程度上,Java 实现了这个目标。

8. 高性能(high-performance)

先前的许多尝试解决跨平台运行的方案对性能要求都很高。其他解释执行的语言系统,如 BASIC, Tcl, PERL 都有无法克服的性能缺陷。然而,Java 却可以在性能较低的 CPU 上顺利运行。Java 的字节码经过仔细设计,因而使用 JIT(Just In Time)编译技术能很容易地将字节码直接转换成高性能的本机代码,因而“高效且跨平台”对 Java 来说不再是矛盾。

9. 多线程(multi-threaded)

设计 Java 的目标之一是满足人们创建交互式网上程序的需要。为此,Java 支持多线程编程,用 Java 编写的应用程序可以同时执行多个任务。Java 运行时,系统在多线程同步方面具有成熟的解决方案,这使用户能够创建出运行平稳的交互式系统。Java 的多线程机制非常好用,因此用

户只需关注程序细节的实现,而不用担心后台的多任务系统。

10. 动态(dynamic)

Java 程序带有多种的运行时的类型信息,用于在运行时校验和解决对象访问问题。这使得在一种安全、有效的方式下动态地连接代码成为可能,在程序运行时,系统中字节码内的小段程序可以动态地被更新。

1.3.3 Java 的运行原理

1. Java 虚拟机(Java Virtual Machine)

Java 虚拟机(JVM)是可运行 Java 代码的假想计算机。只要根据 JVM 规范描述将解释器移植到特定的计算机上,就能保证经过编译的任何 Java 代码能够在该系统上运行。如图 1-1 所示,Java 源程序经过编译器(javac.exe)编译后,会生成字节码文件(扩展名为.class),然后通过解释器(java.exe)让字节码文件在 JVM 上解释运行。可见,Java 程序是由 JVM 负责解释运行的,而非操作系统。这样做的好处是可以实现跨平台运行程序,也就是说针对不同的操作系统,可以编写相同的程序,只需要安装不同版本的 JVM 即可。这种方式使得 Java 语言具有“一次编写,到处运行(write once, run anywhere)”的特性。

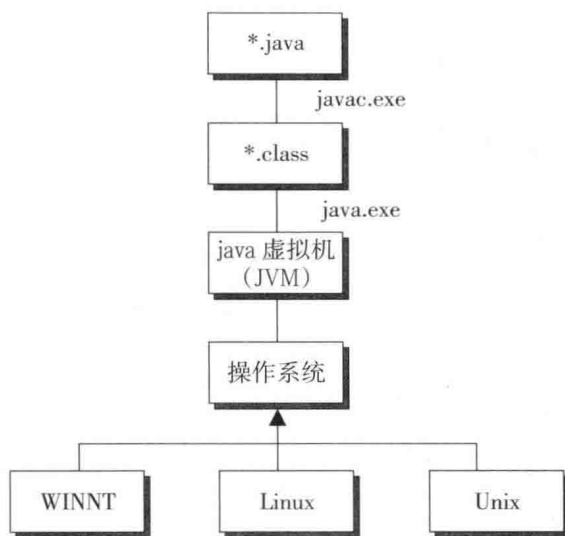


图 1-1 Java 虚拟机

2. Java 字节码

如图 1-1 所示,Java 源代码的扩展名是.java,经过 Java 编译器(javac.exe)编译后,输出的并不是可执行的代码,而是字节码(bytecode)文件,扩展名是.class,然后经过 java 解释器(java.exe),让字节码在 JVM 上被解释运行。字节码是一套被设计用来在 JVM 上执行的高度优化的指令集,这可能有点让人吃惊,因为像 C++等语言的编译结果是可执行的代码。事实上,出于对性能考虑,许多现代语言都被设计为编译型,而不是解释型。然而,正是由于可以通过 JVM 运行 Java 字

节码,才有助于 Java 更容易地在 Internet 这个大范围的环境下运行程序。原因非常简单:只要在各种平台上都安装相应版本的 Java 虚拟机就可以了。

通常,被解释的程序的运行速度会比相同内容的被编译为可执行代码的程序运行速度慢一些。但是对 Java 来说,这两者之间的差别不太大。使用字节码能够使 Java 运行时系统的程序执行速度比用户想象的快得多。Sun 公司在 Java 2 发行版中提供了一个字节码编译器——JIT (Just In Time, 即时)。虽然 Java 被设计为解释执行的程序,但 JIT 即时编译执行的方法能使性能得到较大提高。

1.3.4 Java 的开发工具——JDK

1. JDK(Java Development Kit)

即 Java 开发工具包。我们可以从甲骨文公司的官方网站 (<http://www.oracle.com>) 上下载到它的最新版本。如图 1-2 所示,官网上下下载的 JDK 是一个简单的命令行工具集,包括编译 Java 源代码的编译器、执行 Java 字节码的解释器、测试 Java Applet 的浏览器,以及其他一些实用工具,还包括了 Java 运行时环境(Java Runtime Environment, JRE)。JRE 由 JVM 和 Java 运行时类库(Java Runtime Classes)构成。

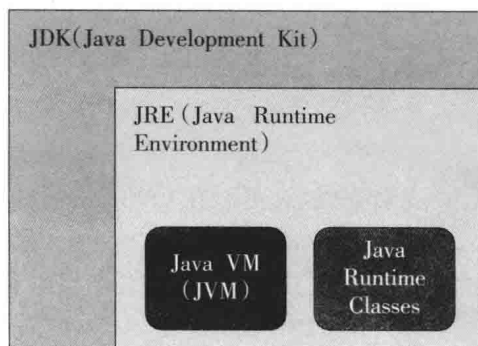


图 1-2 JDK

2. JDK 的安装和配置

要开发 Java 程序首先必须要配置好环境变量,而 Java 的运行环境的配置比较麻烦,相信有些读者也会有这种体会。下面来看一下 JDK 的安装过程。在这里 JDK 选用的是 JDK-7u60 版本(安装文件 jdk-7u60-windows-i586.exe)。

如图 1-3、图 1-4 所示, JDK 的安装采用默认的安装设置即可。

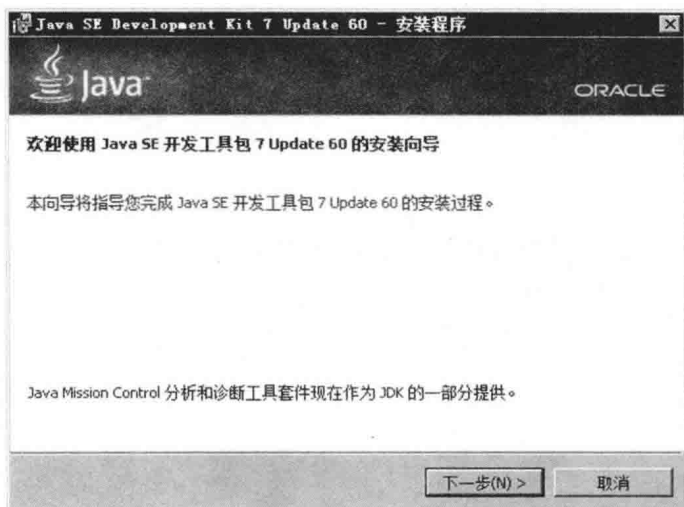


图 1-3 启动 JDK 安装程序

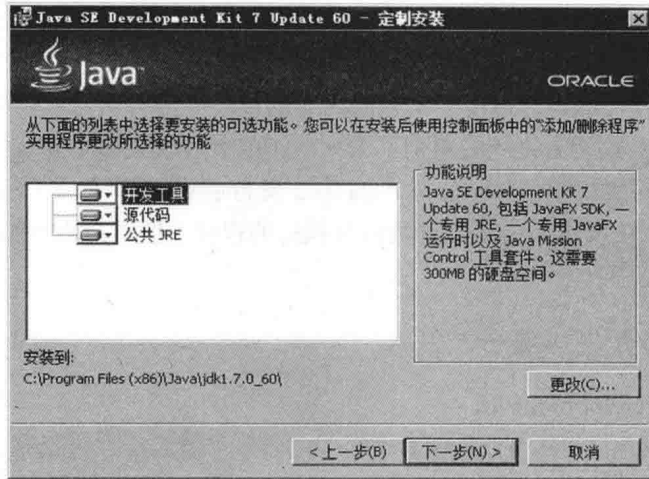


图 1-4 选择 JDK 安装路径

安装完成后,可以在 C:\Program Files(x86)\Java\jdk1.7.0_60\bin 目录下,找到编译器(Javac.exe)和解释器(java.exe),而这两个命令并不是 Windows 自带的命令,所以使用它们的时候需要配置好环境变量,这样就可以在任何目录下使用这两个命令了。配置环境变量的过程如下:

(1)右键点击“我的电脑”→选择“属性”→选择“高级”→“环境变量”→“Path”。出现如图 1-5 所示的环境变量设置界面,在系统变量 Path 后面追加 C:\Program Files(x86)\Java\jdk1.7.0_60\bin,注意:追加的路径和原有路径用西文分号分隔。



图 1-5 环境变量设置界面

(2)在环境变量设置界面中新建一个系统变量 classpath,如图 1-6 所示,设置 classpath 变量的值为:

C:\Program Files(x86)\Java\jdk1.7.0_60\lib;C:\Program Files(x86)\Java\jdk1.7.0_60\lib\dt.jar;

C:\Program Files(x86)\Java\jdk1.7.0_60\lib\tools.jar;

注意:上述路径的标点符号都是西文半角形式。



图 1-6 添加 classpath 系统变量

1.3.5 第一个 Java 程序

Java 程序分为两种形式:一种是网页上使用的 Applet 程序(Java 小程序),另一种是 Application 程序(Java 应用程序),下面以 Java Application 程序为例,给读者演示 Java 程序的编写、编译和运行过程。

【例 1-1】 HelloWorld.java

```
public class HelloWorld{
    // 是程序的起点,所有程序由此开始运行
    public static void main(String args[]){
        // 此语句表示向屏幕上打印 "Hello World !" 字符串
        System.out.println("Hello World !");
    }
}
```

将上面的程序保存为 HelloWorld.java 文件,并在命令行窗口中输入指令:

```
javac helloworld.java
```

编译没有提示错误后输入指令:

```
java HelloWorld
```