



教育部实用型信息技术人才培养系列教材

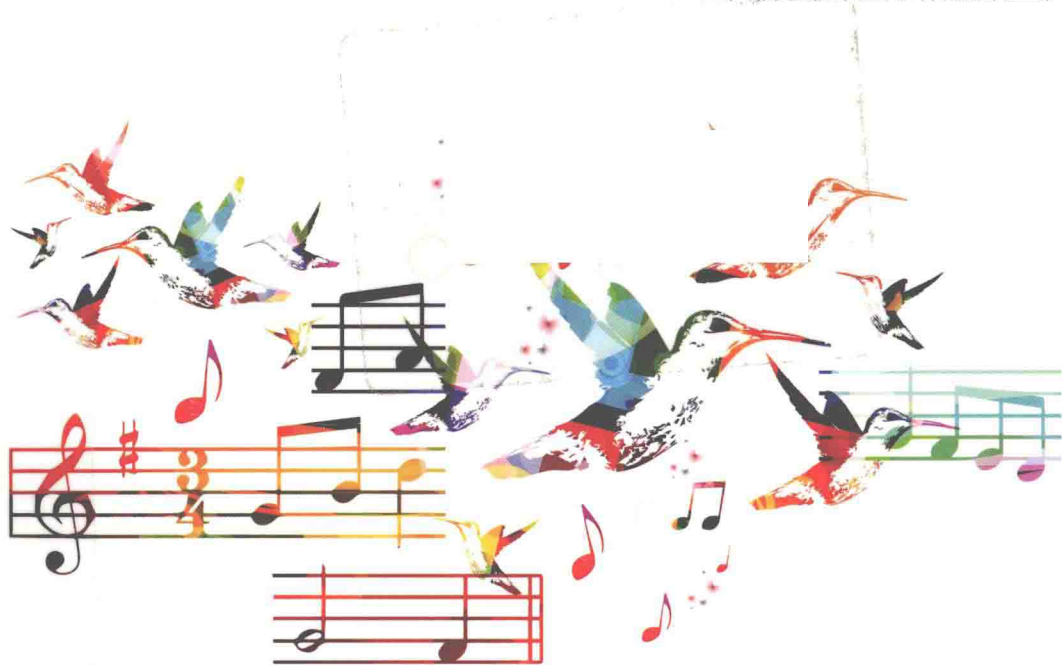
Java

程序设计

教程与上机实验

边金良 孙红云 | 编著

全国信息技术应用培训教育工程工作组 | 审定



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



教育部实用型信息技术人才培养系列教材

图灵社区会员专享图书

图灵社区会员专享图书

图灵社区会员专享图书

图灵社区会员专享图书

图灵社区会员专享图书

图灵社区会员专享图书

图灵社区会员专享图书

图灵社区会员专享图书

Java

程序设计

教程与上机实验

边金良 孙红云 | 编著

全国信息技术应用培训教育工程工作组 | 审定



人民邮电出版社

北京

图书在版编目 (CIP) 数据

Java程序设计教程与上机实验 / 边金良, 孙红云编
著. — 北京: 人民邮电出版社, 2015. 5
教育部实用型信息技术人才培养系列教材
ISBN 978-7-115-38222-1

I. ①J… II. ①边… ②孙… III. ①JAVA语言—程序设计—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第040089号

内 容 提 要

本书是在编者多年从事 Java 教学和项目开发的基础上编写而成的, 以面向对象的编程思想为指导, 通过大量的项目案例, 详细讲述了 Java 程序设计的基本方法和技巧。

全书共 11 章, 第 1 章主要讲解 Java 的基础知识; 第 2 章主要介绍 Java 语言的程序基础, 包括 Java 中的常量和变量、运算符、语句、数组和方法。第 3~4 章主要介绍面向对象的程序设计, 包括继承、封装和多态。第 5~11 章主要介绍 Java 语言中的一些高级应用, 包括异常处理机制、线程、常用 API、IO 输入输出、图形用户界面、网络编程、Java 数据库操作等。

本书知识点讲解精细, 编程实例切合工作实际, 具有很强的操作性和代表性。通过本书的学习, 读者能在快速掌握 Java 语言核心内容的基础上, 学会使用 Java 语言进行程序开发的流程、方法和技巧。

本书不仅可以作为高等学校、高职高专院校的教材, 也可以作为相关培训机构的培训教材。同时, 对于正在使用 Java 作为编程语言的程序员也有一定的参考价值。

◆ 编 著 边金良 孙红云

责任编辑 李 莎

责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鑫正大印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 18.5

字数: 491 千字

印数: 1—2 500 册

2015 年 5 月第 1 版

2015 年 5 月北京第 1 次印刷

定价: 39.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

教育部实用型信息技术人才培养系列教材编辑委员会

(暨全国信息技术应用培训教育工程专家组)

主任委员 侯炳辉 (清华大学 教授)

委 员 (以姓氏笔划为序)

方美琪 (中国人民大学 教授)

甘仞初 (北京理工大学 教授)

孙立军 (北京电影学院动画学院 院长)

刘 灵 (中国传媒大学广告学院 副院长)

许 平 (中央美术学院设计学院 副院长)

张 骏 (中国传媒大学动画学院 副院长)

陈 明 (中国石油大学 教授)

陈 禹 (中国人民大学 教授)

杨永川 (中国人民公安大学 教授)

彭 澎 (云南财经大学现代艺术设计学院 教授)

蒋宗礼 (北京工业大学 教授)

赖茂生 (北京大学 教授)

执行主编 薛玉梅 (全国信息技术应用培训教育工程负责人
教育部教育管理信息中心开发处处长 高级工程师)

执行副主编

于 泓 (教育部教育管理信息中心)

王彦峰 (教育部教育管理信息中心)

薛 佳 (教育部教育管理信息中心)

出版说明

信息化是当今世界经济和社会发展的趋势，也是我国产业优化升级和实现工业化、现代化的关键环节。信息产业作为一个新兴的高科技产业，需要大量高素质复合型技术人才。目前，我国信息技术人才的数量和质量远远不能满足经济建设和信息产业发展的需要，人才的缺乏已经成为制约我国信息产业发展和国民经济建设的重要瓶颈。信息技术培训是解决这一问题的有效途径，如何利用现代化教育手段让更多的人接受到信息技术培训是摆在我们面前的一项重大课题。

教育部非常重视我国信息技术人才的培养工作，通过对现有教育体制和课程进行信息化改造、支持高校创办示范性软件学院、推广信息技术培训和认证考试等方式，促进信息技术人才的培养工作。经过多年的努力，培养了一批又一批合格的实用型信息技术人才。

全国信息技术应用培训教育工程（又称 ITAT 教育工程）是教育部于 2000 年 5 月启动的一项面向全社会进行实用型信息技术人才培养的教育工程。ITAT 教育工程得到了教育部有关领导的肯定，也得到了社会各界人士的关心和支持。通过遍布全国各地的培训基地，ITAT 教育工程建立了覆盖全国的教育培训网络，对我国的信息技术人才培养事业起到了极大的推动作用。

ITAT 教育工程被专家誉为“有教无类”的平民学校，以就业为导向，以大、中专院校学生为主要培训目标，也可以满足职业培训、社区教育的需要。培训课程能够满足广大公众对信息技术应用技能的需求，对普及信息技术应用起到了积极的作用。据不完全统计，在过去 8 年中共有 150 余万人次参加了 ITAT 教育工程提供的各类信息技术培训，其中有近 60 万人次获得了教育部教育管理信息中心颁发的认证证书。ITAT 教育工程为普及信息技术、缓解信息化建设中面临的人才短缺问题做出了一定的贡献。

ITAT 教育工程聘请来自清华大学、北京大学、人民大学、中央美术学院、北京电影学院、中国传媒大学等单位的信息技术领域的专家组成专家组，规划教学大纲，制订实施方案，指导工程健康、快速地发展。ITAT 教育工程以实用型信息技术培训为主要内容，课程实用性强，覆盖面广，更新速度快。目前工程已开设培训课程 20 余类，共计 50 余门，并将根据信息技术的发展，继续开设新的课程。

本套教材由清华大学出版社、人民邮电出版社、机械工业出版社、北京希望电子出版社等出版发行。根据教材出版计划，全套教材共计 60 余种，内容将汇集信息技术应用各方面的知识。今后将根据信息技术的发展不断修改、完善、扩充，始终保持追踪信息技术发展的前沿。

ITAT 教育工程的宗旨是：树立民族 IT 培训品牌，努力使之成为全国规模最大、系统性最强、质量最好，而且最经济实用的国家级信息技术培训工程，培养出千千万万个实用型信息技术人才，为实现我国信息产业的跨越式发展做出贡献。

全国信息技术应用培训教育工程负责人 薛玉梅
系列教材执行主编

编 者 的 话

Java 是目前最为流行的网络开发语言之一，目前已经在桌面级、企业级以及移动通信方面得到了广泛的应用。本书以 Java 2 为基础，以 Eclipse 作为开发工具，全面介绍了 Java 程序设计的知识。

本书内容及特点

本书重点讲解了 Java 程序设计的基础知识，自始至终贯彻“边用边学，实例导学”的思想模式，在内容的安排上，充分考虑到初学者的接受能力和实际需要，首先将相关知识融合到短小精悍的实际案例中进行详细讲解，最后通过对实际项目案例的操作，对相关知识进行综合巩固，教会读者运用 Java 进行程序设计的方法。另外，为了使读者做到手脑结合、理论与实践相结合，真正掌握并巩固所学知识，在每章最后都安排了习题与操作题。

本书共 11 章，具体内容如下。

第 1 章：Java 基础概述。本章主要对 Java 语言做了一个简单的介绍，具体包括 Java 的发展简史及特点，Java 开发环境的搭建，JDK 的下载与安装，编写、编译和运行 Java 应用程序，Eclipse 的下载及使用等。

第 2 章：Java 程序基础。本章主要讲解了 Java 的编程基础，具体包括 Java 中的常量和变量，Java 中的运算符，Java 中的各种语句，数组的各种语言、数组的使用方法等。

第 3 章：面向对象程序设计。本章主要讲解了 Java 面向对象程序设计的基础，具体包括面向对象和面向过程的区别、类和对象、对象在内存中的存储、类的封装、类的构造方法、this 关键字的用法、static 关键字的作用等。

第 4 章：面向对象的高级属性。本章主要讲解了 Java 面向对象程序设计的高级属性，具体包括继承、final 关键字、多态、抽象类和接口、包等。

第 5 章：Java 的异常处理机制。本章主要讲解了 Java 中对异常的处理，具体包括异常和异常处理、Java 中异常的体系结构、异常捕获和异常处理、throws 关键字、自定义异常等。

第 6 章：线程。本章主要讲解了 Java 中线程的基本知识，具体包括进程和线程、单线程和多线程、多线程的创建、后台线程和前台线程、联合线程、线程的安全问题、同步代码块、同步函数、线程的锁、线程的各个状态等。

第 7 章：Java 常用 API。本章主要讲解了 Java API 的知识，具体包括 API 的作用、String 类和 StringBuffer 类、基本数据类型的包装类、Collection 接口和 Map 接口、System 类和 Runtime 类等。

第 8 章：IO 输入输出。本章主要讲解了 Java IO 输入流和输出流的知识，具体包括 File 类、RandomAccessFile 操作文件、流的概念和分类、字符流和字节流的转换、IO 包中类层次的关系等。

第 9 章：图形用户界面。本章主要讲解了 Java 图形用户界面的知识，具体包括图形用户界面所用到的包和常用术语、常用的容器类、常用组件、AWT 事件处理的机制、布局管理等。

第 10 章：网络编程。本章主要讲解了 Java 网络通信的知识，具体包括网络编程的基础知识和基础

概念、TCP 和 UDP、Socket、URL 等。

第 11 章: Java 数据库操作。本章主要讲解了 Java 操作数据库的知识, 具体包括 JDBC 的用途、JDBC 的体系结构、JDBC 访问数据库的步骤等。

随书教学资料包

为了使读者能更好地学习、使用本书, 本书提供以下教学资料包, 该资料包可在人民邮电出版社教育服务与资源网 (<http://www.ptpedu.com.cn>) 上下载。

Java 源程序文件: 本书所有案例的源代码文件。

视频文件: 本书所有案例的视频文件。

读者对象

本书主要面向初级用户, 尤其适合立志从事 Java 程序开发的人员或相关专业的学生。

本书由边金良、孙红云执笔。此外, 参加本书编写的还有史宇宏、张传记、白春英、陈玉蓉、林永、刘海芹、卢春洁、秦真亮、史小虎、孙爱芳、唐美灵、王莹、张伟、徐丽、张伟、赵明富、朱仁成、王海宾、樊明、张洪东、罗云凤、郑成栋、安述照等。在此感谢所有关心和支持我们的同行们。由于编者水平有限, 书中难免有不妥之处, 恳请广大读者批评指正。

我们的联系信箱是 lisha@ptpress.com.cn, 欢迎读者来信交流。

编 者

目 录

第 1 章 Java 基础概述	1	2.1.1 常量	20
■ 1.1 ■ Java 语言简介	2	2.1.2 变量	20
1.1.1 了解 Java 语言平台及其应用编程接口	2	2.1.3 数据类型的转换	22
1.1.2 Java 语言的特点	2	2.1.4 变量的作用域	24
■ 1.2 ■ Java 开发环境的搭建	3	■ 2.2 ■ Java 中的运算符	25
1.2.1 JDK 的安装	3	2.2.1 赋值运算符	25
1.2.2 环境变量的设定	4	2.2.2 算术运算符	25
■ 1.3 ■ Java 中的常用术语	5	2.2.3 关系运算符	26
■ 1.4 ■ 了解 Java 程序的开发方式	6	2.2.4 逻辑运算符	27
1.4.1 使用记事本开发第一个 Java 程序	6	2.2.5 复合赋值运算符	28
1.4.2 使用集成开发环境开发 Java 程序	8	2.2.6 运算符的优先级	28
■ 1.5 ■ Java 中的标识符和关键字	11	■ 2.3 ■ Java 的结构化程序设计	29
1.5.1 标识符	11	2.3.1 顺序结构语句	29
1.5.2 关键字	11	2.3.2 选择结构语句	29
■ 1.6 ■ 了解 Java 的组成结构	12	2.3.3 循环结构语句	33
1.6.1 Java 程序的一般结构	12	■ 2.4 ■ 数组	39
1.6.2 Java 中的注释	13	2.4.1 数组的基本概念	39
■ 1.7 ■ Java 源程序的结构与分析	13	2.4.2 二维数组	41
1.7.1 功能需求与分析	13	2.4.3 与数组有关的操作	42
1.7.2 程序结构分析	14	■ 2.5 ■ Java 中的方法	43
■ 1.8 ■ 上机实训	15	2.5.1 方法的概念和作用	43
1.8.1 【实训一】使用记事本开发 Java 程序	15	2.5.2 详解方法的定义	45
1.8.2 【实训二】使用 Eclipse 开发程序	15	2.5.3 方法的调用	46
1.8.3 实训中的技术要点	16	2.5.4 方法参数的传递过程	46
■ 1.9 ■ 编写程序过程中常见的问题	16	2.5.5 方法的重载	47
■ 1.10 ■ 练习与上机	17	■ 2.6 ■ 上机实训	48
第 2 章 Java 程序基础	19	■ 2.7 ■ 练习与上机	50
■ 2.1 ■ Java 中的常量和变量	20	第 3 章 面向对象程序设计	53
		■ 3.1 ■ 面向对象的概念	54
		■ 3.2 ■ 类和对象	54
		3.2.1 对象	54

3.2.2 类	55	4.4.3 在方法中定义内部类	114
3.2.3 创建和使用对象	57	4.4.4 匿名内部类	114
3.2.4 类的封装	66	■ 4.5 ■ 包	115
3.2.5 访问控制	70	4.5.1 package 语句及应用	115
■ 3.3 ■ 构造方法	71	4.5.2 import 语句及应用	116
3.3.1 构造方法的用途	71	4.5.3 常用的 JDK 包	117
3.3.2 构造方法的注意事项	75	4.5.4 Java 的命名规则	117
■ 3.4 ■ this 引用句柄	77	■ 4.6 ■ jar 文件	117
■ 3.5 ■ 方法的参数传递	78	4.6.1 jar 文件包	117
3.5.1 基本数据类型的参数传递	78	4.6.2 打包 jar 文件	118
3.5.2 引用数据类型的参数传递	78	4.6.3 使用 jar 文件	119
■ 3.6 ■ static 关键字	81	■ 4.7 ■ 上机实训	119
3.6.1 静态变量	81	4.7.1 实训一——面向对象的多态	119
3.6.2 静态方法	82	4.7.2 实训二——面向接口的程序设计	121
3.6.3 静态代码块	83	■ 4.8 ■ 练习与上机	122
3.6.4 main 方法详解	84	第 5 章 Java 的异常处理机制	125
■ 3.7 ■ 上机实训	84	■ 5.1 ■ 了解异常	126
■ 3.8 ■ 练习与上机	86	5.1.1 生活中的异常	126
第 4 章 面向对象的高级属性	89	5.1.2 程序中的异常	126
■ 4.1 ■ 类的继承	90	5.1.3 异常和异常处理	126
4.1.1 继承的概念	90	5.1.4 异常体系结构	127
4.1.2 Java 中的继承	90	■ 5.2 ■ 异常的捕获和处理	128
4.1.3 使用继承注意的问题	99	5.2.1 异常处理的基本语法	128
■ 4.2 ■ 对象的多态性	99	5.2.2 使用 try/catch 语句块捕获异常	128
4.2.1 对象的类型转换	99	5.2.3 使用 try/catch/finally 语句块捕获异常	133
4.2.2 instanceof 操作符	101	5.2.4 使用 try/catch/finally 语句块应注意的问题	133
4.2.3 Object 类	101	■ 5.3 ■ throws 关键字、自定义异常和 throw 关键字	134
4.2.4 面向对象的多态编程	101	5.3.1 自定义异常	134
4.2.5 面向对象的思想总结	105	5.3.2 自定义异常	135
■ 4.3 ■ 抽象类和接口	105	5.3.3 throw 关键字	135
4.3.1 抽象类	105	■ 5.4 ■ 上机实训	136
4.3.2 接口	106	5.4.1 实训一——处理程序可能出现的异常	136
4.3.3 面向接口编程	109		
■ 4.4 ■ 内部类	112		
4.4.1 类中定义内部类	112		
4.4.2 内部类被外部引用	114		

5.4.2 实训二——自定义异常	138
■ 5.5 ■ 练习与上机	140
第 6 章 线程	143
■ 6.1 ■ 线程简介	144
6.1.1 线程概述	144
6.1.2 创建多线程	145
■ 6.2 ■ 与线程有关的操作	149
6.2.1 后台线程	149
6.2.2 联合线程	149
6.2.3 向线程传递数据	151
6.2.4 从线程返回数据	152
■ 6.3 ■ 线程同步	153
6.3.1 线程的不安全问题	153
6.3.2 实现线程安全性	154
6.3.3 线程的死锁	157
■ 6.4 ■ 线程间的通信	158
6.4.1 问题的引出	158
6.4.2 解决思路	158
6.4.3 线程通信——生产者消费者问题	159
■ 6.5 ■ 线程的生命周期和状态控制	160
6.5.1 线程的生命周期	160
6.5.2 线程的状态控制	161
■ 6.6 ■ 上机实训	162
■ 6.7 ■ 练习与上机	164
第 7 章 Java 常用 API	165
■ 7.1 ■ 理解 API	166
7.1.1 Java API 简介	166
7.1.2 使用 Java API 文档	166
■ 7.2 ■ 字符串操作类	167
7.2.1 String 类	167
7.2.2 StringBuffer 类	170
■ 7.3 ■ 基本数据类型的包装类	170
■ 7.4 ■ 常用的集合类	172
7.4.1 集合的分类	172
7.4.2 Collection 接口	173
7.4.3 List 接口	175
7.4.4 Map 接口	178
■ 7.5 ■ Hashtable 类和 Properties 类	179
7.5.1 Hashtable 类	179
7.5.2 Properties 类	180
■ 7.6 ■ System 类和 Runtime 类	182
7.6.1 System 类	182
7.6.2 Runtime 类	183
■ 7.7 ■ 上机实训	183
■ 7.8 ■ 练习与上机	185
第 8 章 IO 输入输出	187
■ 8.1 ■ File 类	188
■ 8.2 ■ RandomAccess File 类	189
■ 8.3 ■ Java IO 流	191
8.3.1 流的概念	191
8.3.2 流的分类	192
8.3.3 字节流	193
8.3.4 字符流	195
8.3.5 过滤流与包装类	196
8.3.6 字节流与字符流的转换	202
8.3.7 IO 包中的类层次关系图	203
■ 8.4 ■ 上机实训	204
8.4.1 实训一——使用 Random AccessFile 类读取文件的内容	204
8.4.2 实训二——使用 FileWriter 将 内容追加到文件末尾	205
■ 8.5 ■ 练习与上机	205
第 9 章 图形用户界面	207
■ 9.1 ■ Swing 概述	208
9.1.1 AWT 包和 Swing 包	208
9.1.2 一个 GUI 实例	208
9.1.3 与 GUI 有关的术语	209
■ 9.2 ■ 常用组件	209
9.2.1 容器组件	209
9.2.2 标签	211

9.2.3 文本输入类组件	213	10.2.2 基于 TCP 的网络编程	246
9.2.4 选择类组件	217	10.2.3 基于 UDP 的网络编程	252
9.2.5 菜单	220	■ 10.3 ■ 基于 URL 的网络编程	255
9.2.6 对话框	222	10.3.1 统一资源定位器	255
9.2.7 表格	224	10.3.2 从 URL 读取 WWW 网络资源	257
9.2.8 组件表面重绘	228	10.3.3 通过 URLConnetction 连接 WWW	257
9.2.9 改变应用程序的观感	229	■ 10.4 ■ 上机实训	259
■ 9.3 ■ AWT 事件	229	■ 10.5 ■ 练习与上机	261
9.3.1 事件处理机制	229	第 11 章 Java 数据库操作	263
9.3.2 用事件监听器处理事件	231	■ 11.1 ■ JDBC 概述	264
9.3.3 事件	234	11.1.1 什么是 JDBC	264
9.3.4 使用匿名内置类实现事件监听	235	11.1.2 JDBC 的用途	264
■ 9.4 ■ 布局管理器	235	11.1.3 JDBC 与 ODBC 和其他 API 的 比较	264
9.4.1 FlowLayout	236	11.1.4 JDBC 的体系结构	265
9.4.2 BorderLayout	236	■ 11.2 ■ 使用 JDBC 操作数据库	266
9.4.3 GridLayout	237	11.2.1 java.sql 包	266
9.4.4 自定义布局管理器	237	11.2.2 JDBC 访问数据库的步骤	266
■ 9.5 ■ 可视化界面设计	237	11.2.3 PreperedStatement 接口	271
■ 9.6 ■ 上机实训	240	11.2.4 ResultSetMetaData 接口	273
■ 9.7 ■ 练习与上机	241	11.2.5 CallableStatement 接口	274
第 10 章 简单的网络编程	243	11.2.6 JDBC 事务	276
■ 10.1 ■ 网络编程的基本概念	244	■ 11.3 ■ 上机实训	278
10.1.1 网络编程的基础知识	244	■ 11.4 ■ 练习与上机	282
10.1.2 网络编程的基础概念	244	附录 练习题参考答案	283
10.1.3 TCP 和 UDP	245		
■ 10.2 ■ 基于 Socket 的网络编程	246		
10.2.1 Socket	246		

第1章

Java 基础概述

📖 学习目标

学习有关 Java 的基本知识,并能用 Java 开发一个简单的程序,掌握 Java 程序的结构。主要包括 Java 语言平台、Java 程序接口、Java 运行环境的搭建、Java 常用术语介绍、Java 程序的开发方式、Java 程序的组成结构以及 Java 中的标识符和关键字等。同时,通过完成本章上机实训,更好地掌握 Java 的基本框架。

📖 学习重点

掌握 Java 开发环境的搭建,主要是 JDK 的下载与安装;能够编写、编译和运行 Java 应用程序;掌握 Eclipse 的下载及使用。

📖 主要内容

- ◆ Java 语言简介
- ◆ Java 开发环境的搭建
- ◆ Java 中的常用术语
- ◆ Java 程序的开发方式
- ◆ Java 中的标识符和关键字
- ◆ Java 程序的组成结构
- ◆ Java 源程序的结构与分析
- ◆ 上机实训
- ◆ 编写程序过程中常见的问题

1.1 Java 语言简介

Java 是由 Sun Microsystems 公司（简称 Sun 公司）于 1995 年 5 月推出的一种适合于各类计算环境、纯面向对象的高级计算机编程语言。随着其功能的不断完善，Java 已逐步从一种单纯的计算机高级编程语言发展成为一种重要的 Internet 平台，被广泛应用于企业系统开发、桌面系统开发，以及消费电子产品软件开发等方面。这一节首先了解 Java 语言的平台及特点。

1.1.1 了解 Java 语言平台及其应用编程接口

Java 平台是由 Java 虚拟机和 Java 应用编程接口构成的。Java 引进了虚拟机原理，并运行于虚拟机中。同时，Java 应用编程接口为独立于操作系统的标准接口，这使得用户在个人计算机硬件或操作系统平台上安装一个 Java 平台之后，就可以顺利运行 Java 应用程序。现在，Java 平台又嵌入了几乎所有的操作系统，这样，用户只要编译一次 Java 程序，就可以在各种系统中顺利运行。

1.1.2 Java 语言的特点

Java 作为一款高级计算机编程语言，其具有以下特点。

◆ 跨平台与数据共享的特点。

由于 Java 引进了虚拟机原理，并运行于虚拟机中，Java 虚拟机（Java Virtual Machine）又是建立在硬件和操作系统之上的，它实现了 Java 二进制代码的解释执行功能，提供了不同平台的接口，因此，使用 Java 编写的程序能运行于各种操作系统，并能在世界范围内进行数据共享。

提示：Java 语言的跨平台，可以理解任何写好的 Java 程序既可以运行在 Windows 平台上也可以运行在 Linux 平台上，目前流行的 Android 智能机的应用程序就是 Linux 平台上运行的 Java 应用程序。

◆ 易掌握与安全可靠的特点。

Java 的编程类似于 C++ 的编程，对于有 C++ 语言基础的用户来说，掌握 Java 的精髓是轻而易举的事，即使是没有 C++ 语言基础的用户，只要用心学习，也能轻松掌握 Java 的编程技能。另外，Java 舍弃了 C++ 的指针对存储器地址的直接操作，程序运行时，内存由操作系统来分配，这样就避免了病毒通过指针侵入系统，非法访问 Java 语言，因此，Java 的安全可靠是其他计算机语言无法比拟的。

◆ 程序的简洁性和易于维护的特点。

由于 Java 吸取了 C++ 面向对象的概念，将数据封装于类中，用户只须把主要精力用在类和接口的设计和应用上，只须编写一次程序代码，就可以利用类的封装性、继承性等优点，对程序进行反复利用，这就使得 Java 程序更简洁和便于维护。

◆ 子类单一继承父类的特点。

Java 提供了众多的一般对象的类，类的继承关系是单一的，而非多重的。一个子类只有一个父类，子类的父类又有一个父类，其相互关系成树状，树的根部就相当于父类，为 Object 类。Object 类功能强大，它所派生的子类就相当于树的枝杈。在 Java 程序开发中，经常会使用 Object 类来派生出其他子类，子类通过继承可以使用父类。这种子类单一继承父类的特点，即使得 Java 用户在编写程序时省却了许多重复性的工作，同时也使程序便于维护。

◆ 方便的网络文件的使用特点。

Java 建立在扩展 TCP/IP 网络平台上，库函数提供了用 HTTP 和 FTP 协议传送和接收信息的方法，这使得用户使用网络上的文件时非常方便，如同使用本机文件一样。

◆ 强大的数据检测的特点。

计算机程序在编译和运行的过程中出错是难免的，一旦程序出错，要想快速、准确地找到出错的原因似乎不太容易。Java 自己操纵内存，同时实现了真数组；另外，Java 还提供了 Null 指针检测、数组边界检测、异常出口、Byte Code 校验等功能，因此减少了内存出错和覆盖数据的可能性；其类型检查还能够帮助用户检查出开发早期

出现的许多错误。Java 的这些功能特征,大大缩短了开发 Java 应用程序的周期。

1.2 Java 开发环境的搭建

Java 程序一个显著的特点就是跨平台操作,也就是我们常说的“一次编译,到处运行”,其根本原因是 Java 应用程序需要一套自己的运行环境。通常把这个运行环境称为 JDK,这一节就来学习 JDK 的安装以及系统环境变量的设置的知识。

1.2.1 JDK 的安装

Sun Microsystems 公司免费提供了 Java 开发工具包,用户可以登录 Sun 公司的网站(<http://java.sun.com/javase/downloads/index.jsp>)获取免费的 Java 开发包安装程序。本书中所给出的例子程序均在版本为 1.6.0_13 的 JDK 下运行通过,目前最新版本为 1.7.0_09。下面下载并安装 Java 程序。

【任务 1】在 Windows 操作系统下搭建 JDK 环境。

Step 1 登录 Sun 公司的网站,找到 JDK 的开发包安装程序并进行下载。

Step 2 下载完 Java 开发包后需要进行安装。双击 Java 开发包安装程序,出现的安装启动界面如图 1-1 所示。

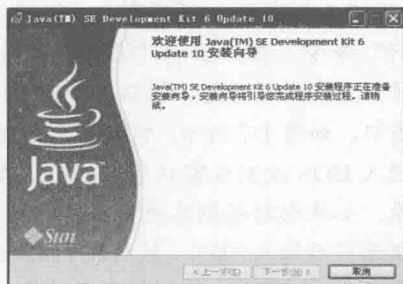


图 1-1 安装启动界面

Step 3 单击“下一步”按钮,弹出“许可证协议”界面,如图 1-2 所示,阅读相应条款后,

单击“接受”按钮继续安装。

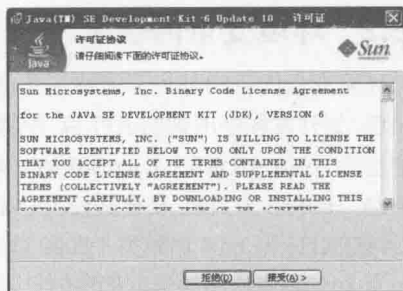


图 1-2 “许可证协议”界面

Step 4 用户可以选择安装开发包的部分或全部内容,如图 1-3 所示。

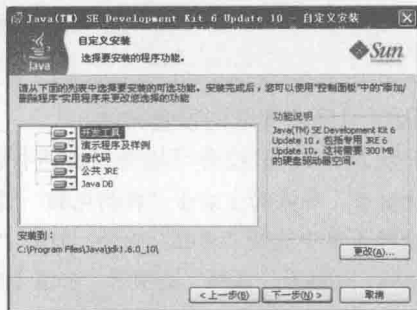


图 1-3 安装内容选择

Step 5 如果用户想要更改开发包的安装路径,则单击“更改”按钮,弹出如图 1-4 所示的对话框。在该对话框中更改了安装路径,单击“确定”按钮关闭该对话框。

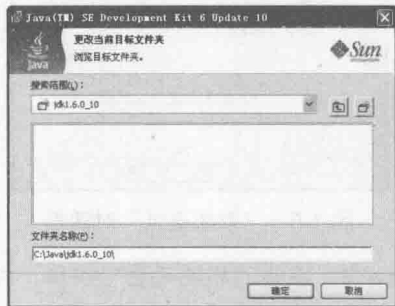


图 1-4 安装路径选择

Step 6 在选定了安装内容和安装路径后,单击“下一步”按钮进入正式安装环节。其他版本的开发包安装过程中除了显示界面略有不同,

操作方法类似。

1.2.2 环境变量的设定

设定环境变量的目的是为了能够正常使用所安装的开发包,如开发程序用到的 API 包。配置的 Java 环境变量包括 JAVA_HOME、CLASSPATH 和 Path,其中, JAVA_HOME 为 JDK 安装路径 (Java 安装目录), CLASSPATH 为 Java 加载类 (Class 或 Lib) 路径, Path 为 Java 编译程序和运行程序的目录。

提示: 初次接触 Java 的读者可能对“类”、“对象”等相关名词有些陌生,没关系,在本章的 1.3 节将对 Java 中的相关名词做详细解释。另外,在本书的后面章节中,还会对相关名词及其作用和应用方法做更深入的讲解。

下面学习环境变量的设置方法。

【任务 2】 为本机设置环境变量并进行测试。

Step 1 在桌面上右击“我的电脑”图标,在弹出的快捷菜单中选择“属性”命令,打开“系统属性”对话框,切换至“高级”选项卡,如图 1-5 所示。

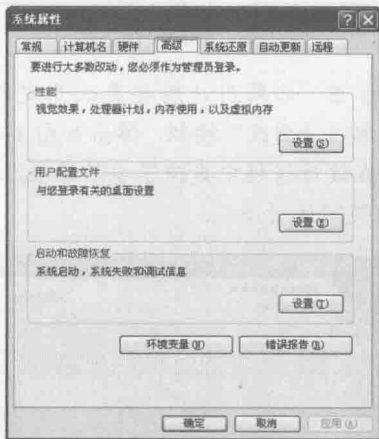


图 1-5 “系统属性”对话框

Step 2 单击“环境变量”按钮,打开“环境变量”对话框。

Step 3 新建 JAVA_HOME 环境变量。单击“新建”按钮,弹出“新建用户变量”对话框,设置变量名为“JAVA_HOME”,变量值为“C:\Program Files\Java\jdk1.6.0_13”,然后单击“确定”按钮。

注意: 上面的变量值“C:\Program Files\Java\jdk1.6.0_13”是 JDK 的安装目录,读者可根据自己安装的实际情况进行设定,如安装在 D 盘就写成“D:\”。

Step 4 新建 CLASSPATH 变量。单击“新建”按钮,弹出“新建用户变量”对话框,设置变量名为“CLASSPATH”,变量值为“;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar”,然后单击“确定”按钮。

Step 5 编辑 Path 变量。在“系统变量”列表中双击变量名为“Path”的变量,弹出“编辑系统变量”对话框,在“变量值”文本框中把“%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin”添加在原变量值的开头处,如图 1-6 所示。



图 1-6 编辑环境变量

Step 6 测试 Java 环境变量。选择“开始”→“运行”命令,在弹出的“运行”对话框中输入“cmd”,单击“确定”按钮,即可进入 DOS 控制台窗口,如图 1-7 所示。需要注意的是,用这种方式进入 DOS 控制台窗口后,当前目录为系统默认目录。如果想转换到其他盘符,例如 D 盘,可以在提示符后输入“D:”,然后按 Enter 键即可。

输入“java -version”,然后按 Enter 键,控制台会输出当前所使用的 JDK 的版本信息,说明我们已经正确地构建 Java 的开发环境,如图 1-8 所示。

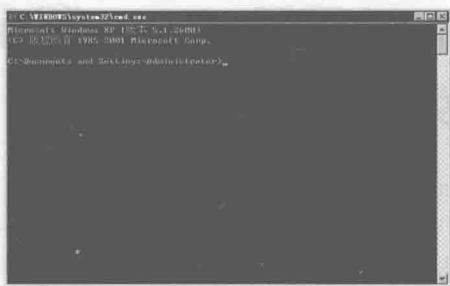


图 1-7 DOS 控制台窗口



图 1-8 正确构建 Java 的开发环境

1.3 Java 中的常用术语

在正式学习 Java 之前,首先了解 Java 中的一些常用术语,这对于初次接触 Java 的读者非常重要。

1. 关键字

关键字是指在 Java 中具有专门的意义和用途的一些字符,也被称为保留字(Reserved Word)。关键字不能当作一般的标识符使用。例如:

```
int age;
```

其中, int 是关键字,整条语句表示 age 是一个 int 类型的数据。



提示: Java 中的关键字有很多,有关关键字及其含义的内容将在本章 1.5 节进行详细讲解。

2. 类

在面向对象的程序设计语言中,类是对一类

“事物”的属性与行为的抽象。例如, Person (人) 是对地球上包括你、我、他、张三、李四等的所有具有特殊智能的生物的抽象。“你”、“我”、“他”、“张三”、“李四”等都属于“Person (人)”这一类所包含的个体。现实世界中可以抽象出许许多多的类,如汽车类、书籍类、水果类等。

3. 对象

简单地说,对象是类的具体的个体。比如,张三是 Person 类的一个对象。Person 可能存在无数个对象(就好像地球上存在数十亿人一样)。

4. 属性和方法

属性就是对象具有的各种特征,这些特征使得任何一个对象都能区别于其他的对象;方法就是对对象执行的操作。例如, Person 类具有姓名、身份证号、出生日期等属性,具有吃饭、睡觉、学习、跑、跳等方法。

5. 类和对象的关系

(1) 类是一个抽象的概念,它不存在于现实中的时间和空间里。类仅仅是为所有的对象定义了抽象的属性与行为,就像“Person (人)”这个类,它虽然可以包含很多个体,但它本身不存在于现实世界中。

(2) 对象是类的一个具体实例,它是一个实实在在存在的东西。

(3) 类是一个静态的概念。类本身不携带任何数据,当没有为类创建任何对象时,类本身不存在于内存空间中。

(4) 对象是一个动态的概念。每一个对象都存在着有别于其他对象的属于自己的独特的属性和行为。对象的属性可以随着它自己的行为而发生改变。

6. 类和对象在 Java 中的表示

在 Java 中用关键字 class 表示一个类,用大括号“{”和“}”定义属性和方法,如下。

```
class person {
    //定义属性部分
    string name;
    string personId;
```

```
//定义方法部分
public void eat(){
    ...
}
public void sleep(){
    ...
}
}
```

上述语句表示一个类，类的名字为 `person`，该类有两个属性（分别是 `name` 和 `personId`）和两种方法（分别是 `eat` 和 `sleep`）。

 提示：有关类和对象，将在本书第 3 章进行详细讲解。

7. 访问控制修饰符

访问控制修饰符说明类或类的成员的可访问范围，即可以被哪些类和对象访问。修饰符一共有 4 个：`public`、`protected`、`default` 和 `private`。

访问修饰符的作用如表 1-1 所示。

表 1-1 常用的访问修饰符

修饰符	同一类中	同一包中	不同包中的子类	不同包中的非子类
public	Yes	Yes	Yes	Yes
protected	Yes	Yes	Yes	No
default	Yes	Yes	No	No
private	Yes	No	No	No

 提示：有关访问控制修饰符，将在本书第 3 章 3.2.5 节进行详细讲解。

8. 数组

所谓数组，就是在程序设计中，为了处理方便，把具有相同类型的若干变量有序地组织起来的一种形式。这些按序排列的同类数据元素的集合称为数组。数组用“数组名+[]”表示，例如，`String[] args` 表示一个 `args` 的字符串数组类型。

 提示：有关数组，将在本书 2.4 节进行详细讲解。

1.4 了解 Java 程序的开发方式

Java 是一个跨平台的高级计算机语言，其程序的开发主要有两种方式，分别是使用任何一种文本编辑器（如记事本）开发程序和使用集成开发环境开发程序，不管使用哪种方式开发 Java 应用程序，一般都要遵循以下三步。

第一步：编写源程序，可用任意的文本编辑程序保存为扩展名为 `.java` 的文件。

第二步：编译源程序，将 `.java` 文件编译为 `.class` 文件。编译程序需要使用可执行文件 `javac.exe`，该文件位于 Java 安装目录的 `bin` 子目录中，称为 Java 编译器（Java Compiler——`javac` 名称的由来），用于对指定的 Java 源代码进行编译。

格式为：`javac 文件名.java`

第三步：运行程序，在 Java 虚拟机下将 `.class` 文件解释并运行。

格式为：`java 文件名`

 注意：Java 源程序的文件扩展名一定要为 `.java`，而编译后的 `.class` 文件的文件名和 Java 源程序文件的文件名一样。

下面对这两种程序开发方式分别进行讲解。

1.4.1 使用记事本开发第一个 Java 程序

这一小节首先学习使用记事本开发一个 Java 应用程序，重点掌握 Java 应用程序的结构。

【任务 3】使用记事本开发一个 Java 应用程序，要求输出“Hello World 欢迎使用本系统”，程序效果如图 1-9 所示。

Step 1 编写源程序。使用记事本输入下列源代码。

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("HelloWorld");
    }
}
```