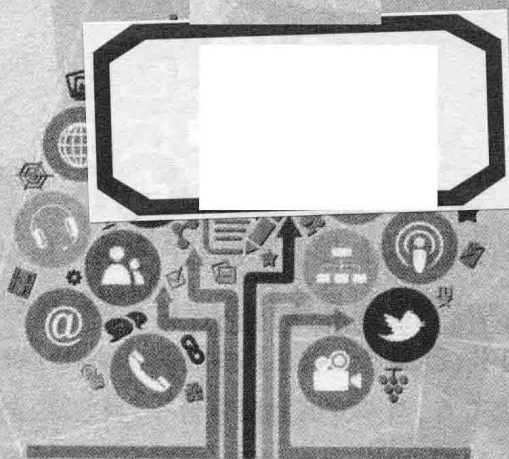




Java 程序设计 实验教程

李 梓 主编
李 娟 于海涛 李宏博 副主编





Java 程序设计 实验教程

李 梓 主编
李 娟 于海涛 李宏博 副主编



黑龙江大学出版社
HEILONGJIANG UNIVERSITY PRESS

图书在版编目 (CIP) 数据

Java 程序设计实验教程 / 李梓主编. — 哈尔滨 :
黑龙江大学出版社, 2016. 11
ISBN 978-7-5686-0063-7

I. ① J… II. ① 李… III. ① JAVA 语言—程序设计—
教材 IV. ① TP312.8

中国版本图书馆 CIP 数据核字 (2016) 第 263743 号

Java 程序设计实验教程

Java CHENGXU SHEJI SHIYAN JIAOCHENG

李 梓 主 编 李 娟 于海涛 李宏博 副主编

责任编辑 李 丽 李 卉
出版发行 黑龙江大学出版社
地 址 哈尔滨市南岗区学府三道街 36 号
印 刷 哈尔滨市石桥印务有限公司
开 本 720×1000 1/16
印 张 20
字 数 317 千
版 次 2016 年 11 月第 1 版
印 次 2016 年 11 月第 1 次印刷
书 号 ISBN 978-7-5686-0063-7
定 价 35.80 元

本书如有印装错误请与本社联系更换。

版权所有 侵权必究

前 言

Java 是一种可以编写跨平台应用程序的面向对象的程序设计语言,是由 Sun 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台(即 JavaEE、JavaME、JavaSE)的总称。Java 自面世后就变得非常流行,并且发展迅速。Java 技术具有卓越的通用性、高效性、平台移植性和安全性,真正做到了“一次编写,处处运行”,其广泛应用于个人计算机、数据中心、游戏控制台、科学超级计算机、移动电话和互联网,同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下,Java 更具备了显著优势和广阔前景。

国外 80% 以上的大学计算机教学中使用 Java 作为第一门程序设计语言,而我国大量高校依旧在使用 C 语言。Java 是面向对象概念的简单且完美的体现,因而非常适合作为入门语言;而先学习 C 语言会导致学生在学习面向对象编程时遇到“范式迁移”的困扰。

鉴于上述困扰,编者所在高校近年来逐渐用 Java 取代了原来的 C 语言作为第一门程序设计语言课程,同时在向应用技术型院校转型的大潮中,逐年加大了程序设计类课程在总体教学体系中的权重,特别是实验的环节比重,将 Java 体系融入各学期。目前 Java 系列课程包括 Java 程序设计基础、Java 程序设计高级、Java Web 程序设计、Java 框架技术、Android 程序设计基础、Android 程序设计高级等。同时还开设 Web 前端技术、数据库技术、Linux 技术相关课程进行辅助和补充,进而形成了完整的 Java 课程体系。

本书定位于“Java 程序设计基础”的教学教材和实验教材,因而对于 Java 集合框架体系、Java 泛型、JavaGUI 编程、JDBC 数据库编程、Java 注释、Java 网络通信编程和 Java 反射机制等 Java 高级特性在本书范围内并未涉及,而是将它们留在后继 Java 课程中进行讲述,随后的数据结构、数据库原理、计算机网络等课程的开设,更加有助于加快学生对这些知识点的消化和理解。

本书来源于教学实践,由多名从事 Java 语言教学的教师结合多年教学经验编纂而成。采用“基础优先,讲练结合”的教学方式,循序渐进地介绍了 Java 程序设计的基础。本书对每个知识点都进行了深入的分析,并针对每个知识点精

心设计了相关案例,帮助读者理解和掌握 Java 的编程基础,提高读者的实践操作能力。本书可作为计算机类专业的本科或专科教材或实验教材,也可作为信息类相关专业的选修教材。

本书由李梓老师制定大纲并统稿。本书共六章,第一章和第五章由李宏博老师编写,第二章由于海涛老师编写,第三章由李梓老师编写,第四章、第六章和附录由李娟老师编写。全体编写组成员齐心协力、尽心尽力地开展工作,力争把教材编写得更好,但由于水平有限,不当和错误之处在所难免,敬请同行专家和广大读者斧正。

编 者
2016 年 10 月

目 录

第一章 Java 语言概述	1
1.1 Java 语言的发展	2
1.2 Java 语言的特点	3
1.3 Java 语言的应用	5
1.4 怎样学好 Java	6
1.5 开发环境	7
1.6 Java 程序的编写	10
1.7 Java 运行机制	11
1.8 Java 学习与帮助资源的使用	12
1.9 项目设计与完成	13
1.10 练习题	13
第二章 Java 语言基础	16
2.1 知识准备	17
2.1.1 Java 源文件结构	17
2.1.2 语法规则	19
2.1.3 数据类型	21
2.1.4 常量和变量	26
2.1.5 运算符和表达式	29
2.1.6 语句	39
2.1.7 数组	51
2.2 基础实验	57
2.2.1 实验目的	57
2.2.2 实验内容	57
2.3 知识拓展	63
2.3.1 二维数组的声明	63
2.3.2 使用二维数组	65

2.4 综合实验	68
2.5 总结与思考	70
2.5.1 总结	70
2.5.2 练习题	71
第三章 面向对象程序设计	79
3.1 知识准备	80
3.1.1 面向对象程序设计思想	80
3.1.2 类和对象	82
3.1.3 构造方法	87
3.1.4 类的访问与封装	89
3.1.5 继承与覆盖	101
3.1.6 接口与多态	112
3.2 基础实验	162
3.2.1 实验目的	162
3.2.2 实验内容	162
3.3 知识拓展	172
3.3.1 对象的转型	172
3.3.2 内部类	176
3.4 综合实验	179
3.5 总结与思考	188
3.5.1 总结	188
3.5.2 思考	189
3.5.3 练习题	190
第四章 异常处理	200
4.1 知识准备	201
4.1.1 异常和异常类	201
4.1.2 异常处理	204
4.2 基础实验	210
4.2.1 实验目的	210
4.2.2 实验内容	210

4.3 知识拓展	213
4.4 综合实验	215
4.5 总结与思考	217
4.5.1 总结	217
4.5.2 思考	218
4.5.3 练习题	219
第五章 流和文件	222
5.1 知识准备	223
5.1.1 流的概念	223
5.1.2 文件类及其应用	224
5.1.3 字节流	227
5.1.4 字符流	232
5.1.5 标准输入输出	236
5.2 基础实验	239
5.2.1 实验目的	239
5.2.2 实验内容	239
5.3 知识拓展	247
5.4 综合实验	248
5.5 总结与思考	258
5.5.1 总结	258
5.5.2 思考	258
5.5.3 练习题	259
第六章 多线程	262
6.1 知识准备	263
6.1.1 线程	263
6.1.2 线程的创建	263
6.1.3 线程的状态与生命周期	269
6.1.4 线程同步	270
6.2 基础实验	274
6.2.1 实验目的	274

6.2.2 实验内容	274
6.3 知识拓展	280
6.3.1 线程的调度	280
6.3.2 死锁	285
6.4 综合实验	287
6.5 总结与思考	291
6.5.1 总结	291
6.5.2 思考	292
6.5.3 练习题	292
附 录	296
一、系统概述	296
二、系统的主要内容和基本设计要求	296
三、系统设计与实现	298

第一章 Java 语言概述

学习目标

- (1) 了解 Java 的发展和基本原理
- (2) 理解 Java 对 Internet 的贡献
- (3) 理解字节码的重要性
- (4) 掌握 Java 语言的特点
- (5) 熟悉 Java 开发环境
- (6) 了解 Java 的应用

学习内容

- (1) Java 语言的发展
- (2) Java 语言的特点
- (3) Java 语言的应用
- (4) Java 的运行机制
- (5) 开发环境的搭建及使用
- (6) Java 学习与帮助资源的使用

教学重点和难点

重点:

- (1) Java 语言的特点及应用
- (2) Java 的运行机制
- (3) 开发环境的搭建及使用

难点:

- (1) Java 的运行机制

(2)Java 学习与帮助资源的使用

1.1 Java 语言的发展

Java 是印度尼西亚爪哇岛的英文名称,因盛产咖啡而闻名。Java 语言中的许多库类名称多与咖啡有关,如 JavaBeans(咖啡豆)、NetBeans(网络豆)以及 ObjectBeans(对象豆)等等。Java 的标识也正是一杯冒着热气的咖啡。

Java 平台和语言最开始只是 Sun 公司在 1990 年 12 月开始研究的一个内部项目。Sun 公司的一名叫作帕特里克·诺顿的工程师被自己开发的 C 和 C++ 语言编译器搞得焦头烂额,因为其中的 API 极其难用。帕特里克决定改用一种新程序设计语言,同时他也获得了研究公司的一个叫作“Stealth 计划”的项目的机会。

1994 年 10 月,HotJava 和 Java 平台为公司高层进行演示。1994 年,Java 1.0 a 版本已经可以提供下载,但是 Java 和 HotJava 浏览器的第一次公开发布却是在 1995 年 5 月 23 日 Sun World 大会上进行的。1996 年 1 月,Sun 公司成立了 Java 业务集团,专门开发 Java 技术。

Java 的发展历史:

- (1)1995 年 5 月 23 日,Java 语言诞生。
- (2)1996 年 1 月,第一个 JDK - JDK 1.0 诞生。
- (3)1997 年 2 月 18 日,JDK 1.1 发布。
- (4)1998 年 12 月 8 日,JAVA 2 企业平台 J2EE 发布。
- (5)1999 年 6 月,Sun 公司发布 Java 的三个版本:标准版(J2SE)、企业版(J2EE)和微型版(J2ME)。三种技术的关系如图 1.1 所示。
- (6)2000 年 5 月 8 日,JDK 1.3 发布。
- (7)2000 年 5 月 29 日,JDK 1.4 发布。
- (8)2001 年 9 月 24 日,J2EE 1.3 发布。
- (9)2002 年 2 月 26 日,J2SE 1.4 发布,自此 Java 的计算能力有了大幅提升。
- (10)2004 年 9 月 30 日,J2SE 1.5 发布,成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性,J2SE 1.5 更名为 Java SE 5.0。

(11)2005 年 6 月,J2EE 更名为 Java EE,J2SE 更名为 Java SE,J2ME 更名为 Java ME。

(12)2006 年 12 月,Sun 公司发布 JDK 6.0。

(13)2011 年 7 月 28 日,甲骨文公司在收购 Sun 公司的发布 Java 7.0 的正式版。

(14)2014 年 3 月 19 日,甲骨文公司发布 Java 8.0 的正式版。

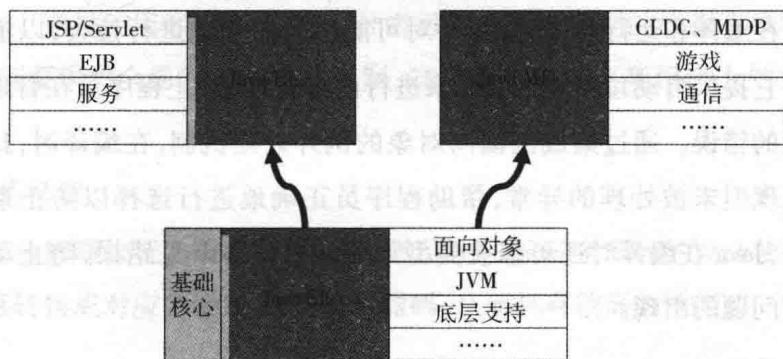


图 1.1 Java 标准版、企业版和微型版三种技术的关系

1.2 Java 语言的特点

Java 语言有以下特点:面向对象、分布式、简单、鲁棒、可移植、体系结构中立、安全、解释执行、动态性、多线程以及高性能。

1. 面向对象

Java 语言的设计集中于对象及其接口,它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法,实现了模块化和信息隐藏;而类则提供了一类对象的原型,并且通过继承机制,子类可以使用父类所提供的方法,实现了代码的复用。

2. 分布式

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议,用户可以通过 URL 地址在网络上很方便地访问其他对象。

3. 简单性

Java 语言是一种面向对象的语言,它通过提供最基本的方法来完成指定的



任务,只需理解一些基本的概念,就可以用它编写出适合于各种情况的应用程序。Java 略去了运算符重载、多重继承等模糊的概念,并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。另外,Java 也适合于在小型机上运行,它的基本解释器及类的支持只有 40 KB 左右,加上标准类库和线程的支持也只有 215 KB 左右。

4. 鲁棒性

Java 在编译和运行程序时,都要对可能出现的问题进行检查,以消除错误的产生。它提供自动垃圾收集功能来进行内存管理,防止程序员在管理内存时容易产生的错误。通过集成的面向对象的例外处理机制,在编译时,Java 提示出可能出现但未被处理的异常,帮助程序员正确地进行选择以防止系统的崩溃。另外,Java 在编译时还可捕获类型声明中的许多常见错误,防止动态运行时不匹配问题的出现。

5. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络中的不同机器上。同时,Java 的类库中也实现了与不同平台的接口,使这些类库可以移植。

6. 体系结构中立

Java 解释器生成与体系结构无关的字节码指令,只要安装了 Java 运行时系统,Java 程序就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示,Java 解释器得到字节码后,对它进行转换,使之能够在不同的平台运行。

7. 安全性

用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针,一切对内存的访问都必须通过对象的实例变量来实现,这样就防止了程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员,同时也避免了指针操作中容易产生的错误。

8. 解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译信息,使得连接过程更加简单。



9. 动态性

Java 的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行,并且 Java 通过接口来支持多重继承,因此比严格的类继承具有更灵活的方式和扩展性。

10. 多线程

多线程机制使应用程序能够并行执行,而且同步机制保证了对共享数据的正确操作。通过使用多线程,程序设计者可以分别用不同的线程完成特定的行为,而不需要采用全局的事件循环机制,这样就很容易实现网络上的实时交互行为。

11. 高性能

与其他解释执行的语言如 BASIC、TCL 不同,Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码,从而具有较高的性能。

1.3 Java 语言的应用

Java 技术自 1995 年问世以来,在我国的应用和开发也得到了迅速普及,总体来看,这些应用主要集中于企业应用开发。有关单位调查显示,从开发领域的分布情况上看,Web 开发占了一半以上,为 57.9%;Java ME 移动或嵌入式应用占 15%;C/S 应用占 11.7%;系统编程占 15.4%。近 30% 的开发者用 Java 从事 C/S 应用或系统级应用的开发。具体应用在如下几个领域:

1. 行业和企业信息化

Sun、IBM、Oracle、BEA 等国际厂商相继推出各种基于 Java 技术的应用服务器以及各种应用软件,带动了 Java 在金融、电信、制造等领域的应用。如清华大学计算机系利用 Java、XML 和 Web 技术研制开发了多个软件平台。

2. 电子政务及办公自动化

东方科技、金蝶、中创等公司开发的 J2EE 应用服务器在电子政务及办公自动化中也得到应用。

3. 嵌入式设备及消费类电子产品

无线手持设备、通信终端、医疗设备、信息家电(如数字电视、电冰箱)、汽车



电子设备等是当前比较热门的 Java 应用领域。

4. 辅助教学

在辅助教学方面,东南大学与中兴通讯股份有限公司开发了远程教学系统,用于本地网上教学、课后学习和异地远程教育;清华大学用 Java 进行了“计算机软件基础课”教学改革,分析研究 Java 教学软件 BlueJ 的汉化方案。

1.4 怎样学好 Java

1. 重视接口

随着软件工程理论不断发展,人们开始意识到面向对象技术中继承的众多缺点,开始努力用聚合代替继承。软件工程解决扩展性的重要原则就是抽象描述,直接使用的工具就是接口。接口近年来逐渐成为 Java 编程方法的核心。大部分开发是建立在规范基础之上的,不需要自己建立复杂的继承关系和庞大的类。因此读懂规范和用好规范已经成为应用程序开发人员的首要任务,Java 各项规范的主要描述手段就是接口。

2. 学好集合框架

Java 描述复杂数据结构的主要方式是集合框架。Java 没有指针,而是通过强大的集合框架描述数组、对象数组等复杂的数据结构。学好这些数据结构的描述方法对于应用程序编写,特别是涉及服务器、三层结构编程至关重要。由于很多语言没有这么强大的集合框架体系,很多初学者不知所措,更不知道拿来做什么用,因此应该对集合框架给予足够的重视。

3. 掌握静态方法和属性

静态方法和属性用于描述某一类对象群体的特征,而不是单个对象的特征。理解静态方法和属性对于理解类与对象的关系是十分有帮助的。在大量的 Java 规范中,静态方法和属性被频繁使用。因此学习者应该理解静态方法和属性。Java 在方法和属性的调用上是一致的,区别只表现在声明的时候,这和 C++ 是不同的。

4. 多线程

Java 程序员热衷于多线程程序编写,并认为这是对逻辑能力的挑战。其实



在大量应用中根本就不需要编写多线程程序,或者说大多数编写应用程序的程序员不会去编写多线程程序。这是因为多线程机制都内置到基础平台当中了。程序员应该了解的是多线程原理和多线程安全,这对于今后准确地把握程序是至关重要的。

5. 了解网络编程

Java 号称是最强大的网络编程语言,但是大多数应用程序开发人员是从来不会自己开发底层网络程序的,因此了解原理就足够了。网络机制是依靠平台实现的,除非自己开发平台,否则是不需要知道 socket(套接字,是两台机器之间的通信端点)怎么实现,如何监听访问。因此在这方面花太多的时间就偏离了“将来的应用开发是在成熟的平台上展开,而不是自己从底层开发平台”这一假设。

6. Java 的异常捕获

异常捕获强迫程序员用显著的与逻辑方法完全不同的方式描述异常,对于程序描述的完整性和严谨性有很大的意义。Java 的初学者应该学习好这种异常捕获机制,养成良好的编程习惯。

1.5 开发环境

Java 的开发环境主要使用 JDK(Java Development Kit),它是整个 Java 的核心,其中包括 Java 编译器、Java 运行工具、Java 文档生成工具、Java 打包工具等。为了获得更大的可移植性,并且由于不是所有潜在用户都安装了最新的 JVM,所以在安装 Java 开发环境时,不建议选择最新的版本。2009 年 Sun 公司被甲骨文公司收购,所以目前读者可以从甲骨文公司的官网下载 Java 各版本的 JDK,网址为 <https://www.oracle.com/java/index.html>。

安装 JDK 分为以下两个步骤:

- (1)准备好 JDK 安装文件。
- (2)配置环境变量 path。

安装的 JDK 步骤如图 1.2、图 1.3 和图 1.4 所示,我们以 JDK8 为例。

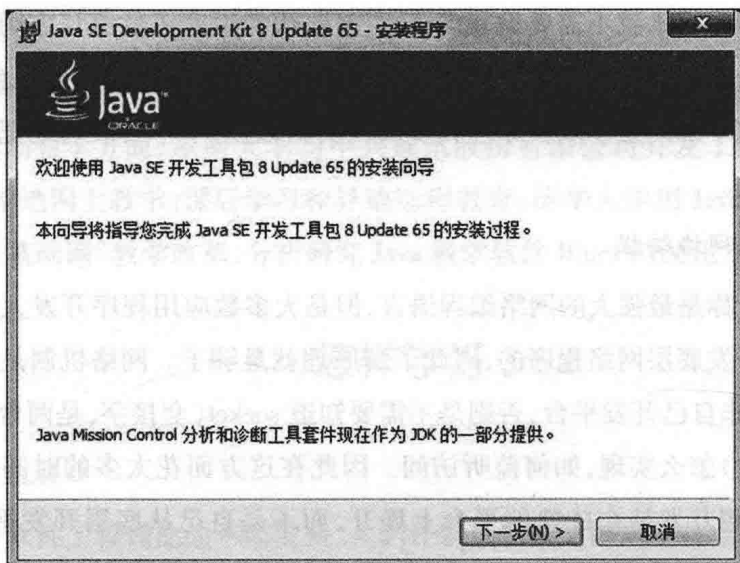


图 1.2 运行 Java 安装程序

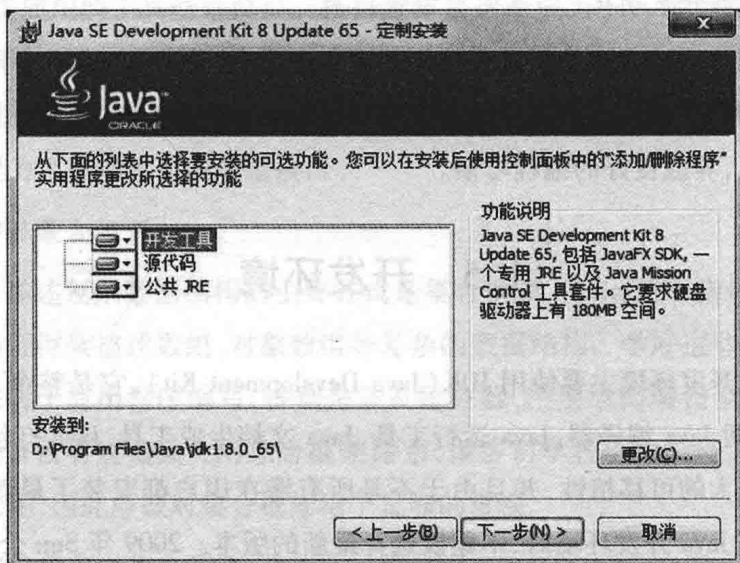


图 1.3 更改安装目录