

21世纪高等学校规划教材 | 计算机科学与技术



Java程序设计 教程与项目实训

温秀梅 司亚超 主编



清华大学出版社

21世纪高等学校规划教材 | 计算机科学与技术

Java程序设计 教程与项目实训

温秀梅 司亚超 主编

清华大学出版社
北京

内 容 简 介

本书通过具有代表性的例子、详尽的讲解和丰富的练习,全面介绍了Java 2的相关知识。本书在体系结构的编排上由浅入深,重点难点突出,对于典型例题进行了分析解释,在附录中整合了习题、实验、课程设计、模拟题等,使全书结构严谨、通俗易懂,兼有普及与提高的双重功能。

本书由四部分组成:第一部分(第1~6章、第8章)对Java基本特性、面向对象程序设计知识、语法进行了较为系统的介绍;第二部分(第7章、第9~11章)是关于Java图形用户界面、网络编程及数据库技术等内容的介绍;第三部分(第12章)是项目实训;第四部分是附录,包括习题、实验、课程设计及模拟题等。

本着少而精的原则,全书层次清晰、结构紧凑,知识信息含量高,特别适合作为计算机专业、非专业的本科生教学或计算机应用培训班的教材,还可以作为自学或函授学习的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 程序设计教程与项目实训/温秀梅,司亚超主编. —北京:清华大学出版社,2017

(21世纪高等学校规划教材·计算机科学与技术)

ISBN 978-7-302-47370-1

I. ①J… II. ①温… ②司… III. ①JAVA语言—程序设计—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第124173号

责任编辑:黄芝李晔

封面设计:傅瑞学

责任校对:徐俊伟

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:24.25 字 数:606千字

版 次:2017年8月第1版 印 次:2017年8月第1次印刷

印 数:1~2000

定 价:49.50元

产品编号:075543-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn

前言

Java 是一种纯面向对象程序设计语言。Java 具有跨平台使用、安全性好、支持多线程和分布式计算等特点,广泛应用于基于 Internet 的网络应用开发。

编写本书之前,作者已从事了多年的“Java 语言程序设计”“C/C++ 程序设计”等课程的教学及科研工作,对于 Java 语言的基本概念、功能及应用有着较深入的理解和丰富的实践经验。本书在内容安排、教学深度、习题、实验及课程设计等方面做了细致的考量,更加贴近“Java 程序设计”课程的教学需求。

本书以现代教育理念为指导,在讲授方式上注意结合应用开发实例,注重培养学生理解面向对象程序设计思想,以提高分析问题和解决实际问题的能力;采用由浅入深、理论与实践相结合的教学思路,通过大量的实例阐述 Java 程序设计的基本理念及编程技巧。

本书由四部分组成:第一部分(第 1~6 章、第 8 章)对 Java 基本特性、面向对象程序设计知识、语法进行了较为系统的介绍;第二部分(第 7 章、第 9~11 章)是关于 Java 图形用户界面、网络编程及数据库技术等内容的介绍;第三部分(第 12 章)是项目实训;第四部分是附录,包括实验、课程设计及模拟题等。

如若教学课时不足,第 9 章到第 12 章内容可以有选择地讲解,为了提高教学质量,建议增加更多的课外学时。此外,读者在进行编程实践时应当养成查阅“帮助文档”的习惯。

本着少而精的原则,全书层次清晰、结构紧凑,知识信息含量高,特别适合作为计算机专业、非专业的本科生教学或计算机应用培训班的教材,还可以作为自学或函授学习的参考书。

本书由河北建筑工程学院温秀梅、司亚超任主编,孟凡兴、孙皓月任副主编,参加编写工作的还有祁爱华、付江龙、赵明瞻、范晶晶、吕国、冯英伟、王庆林、陈素军,全书由温秀梅教授进行审校并统稿。

在本书的编写过程中,参考了大量相关文献,在此对文献的作者表示深深的谢意。

本书在编写和出版过程中得到了清华大学出版社相关人员的大力支持和帮助,在此表示诚挚的感谢。

感谢您选择本书。由于作者水平有限,书中难免有疏漏、错误和不妥之处,恳请广大专家和读者批评指正,我们将不胜感激。

编 者

2017 年 3 月

目 录

第 1 章 Java 概述	1
1.1 Java 的诞生与发展	1
1.2 Java 的特点	2
1.3 Java 开发运行环境的安装与配置	4
1.3.1 安装 JDK	5
1.3.2 安装 Java 帮助文档	6
1.3.3 配置环境变量	7
1.4 Java 的工作原理与运行系统	10
1.4.1 Java 的工作原理	10
1.4.2 Java 的运行系统	11
1.5 简单的 Java 程序	12
1.5.1 Java 应用程序(Application)	12
1.5.2 Java 小应用程序(Applet)	18
习题	21
第 2 章 Java 语言基础	22
2.1 标识符和关键字	22
2.1.1 标识符	22
2.1.2 关键字	23
2.2 基本数据类型	23
2.3 常量、变量	25
2.3.1 常量	25
2.3.2 变量	28
2.4 运算符与表达式	31
2.4.1 赋值运算符和赋值表达式	32
2.4.2 算术运算符和算术表达式	35
2.4.3 关系运算符和关系表达式	36
2.4.4 逻辑运算符和逻辑表达式	37
2.4.5 条件运算符和条件表达式	37
2.4.6 位运算符和位表达式	37
2.5 流程控制语句	38
2.5.1 顺序结构	39

2.5.2	选择结构	40
2.5.3	循环结构	49
2.6	Java 开发类库组成	63
习题		65
第3章	类和对象	66
3.1	概述	66
3.1.1	对象	66
3.1.2	类	67
3.1.3	消息	67
3.1.4	面向对象系统的特性	68
3.2	类	69
3.2.1	类的定义	69
3.2.2	方法重载(method overloading)	73
3.2.3	构造方法	73
3.3	对象的定义和使用	74
3.3.1	创建对象	74
3.3.2	对象的使用	75
3.3.3	对象的清除	76
3.3.4	方法的参数传递	76
3.4	实例变量、实例方法和类变量、类方法	78
3.4.1	变量与方法	78
3.4.2	变量的作用域	80
3.4.3	变量的初始值	82
3.5	this 关键字	82
3.6	包	83
3.6.1	包的定义	83
3.6.2	设置类路径	85
3.6.3	包的使用	85
3.7	内部类和匿名类	87
3.7.1	内部类	87
3.7.2	匿名类	90
3.8	访问控制和类的封装性	91
3.8.1	访问控制	91
3.8.2	类的封装性	92
习题		92
第4章	数组、字符串和向量	93
4.1	数组	93

4.1.1	一维数组	93
4.1.2	二维数组	105
4.2	字符串	110
4.2.1	String 类	111
4.2.2	StringBuffer 类	118
4.2.3	StringTokenizer 类	121
4.3	向量	123
	习题	127
第 5 章	继承性和多态性	128
5.1	继承性	128
5.1.1	派生类的定义	128
5.1.2	super 关键字	130
5.1.3	protected 和 final 关键字	134
5.2	多态性	137
5.3	抽象类和接口	140
5.3.1	抽象类	140
5.3.2	接口	143
5.4	初始化块	152
	习题	155
第 6 章	异常处理	156
6.1	异常处理概述	156
6.1.1	什么是 Java 异常	156
6.1.2	异常处理的目的	156
6.2	异常类的分类与层次	158
6.2.1	异常类的层次结构	158
6.2.2	Exception 类	159
6.2.3	自定义异常类	160
6.3	Java 的异常处理机制	160
6.3.1	捕获异常	161
6.3.2	声明抛出异常	163
	习题	166
第 7 章	图形用户界面	168
7.1	Java GUI 概述	168
7.1.1	AWT 与 Swing	168
7.1.2	Swing 组件	170
7.1.3	Swing 容器	171

7.1.4	基于 Swing 的 Java GUI 设计思路	177
7.2	图形与绘图	177
7.2.1	绘图表面	177
7.2.2	图形环境和图形对象	178
7.2.3	颜色	180
7.2.4	字体	181
7.2.5	在面板上绘图	182
7.3	事件处理与 Swing 常用组件	185
7.3.1	Java 事件处理机制	186
7.3.2	精简事件处理程序的技巧	188
7.3.3	JLabel 组件与鼠标事件处理	191
7.3.4	按钮、文本字段与动作事件	195
7.3.5	单选按钮、复选框、组合框与项目事件	200
7.3.6	列表与 ListSelectionEvent 事件	206
7.3.7	进度条与时间组件	208
7.3.8	菜单与工具栏	209
7.3.9	表格组件与树	214
7.3.10	桌面面板与内部窗口	217
7.4	布局管理器	220
7.4.1	FlowLayout 流布局管理器	221
7.4.2	BorderLayout 边界布局管理器	223
7.4.3	GridLayout 网格布局管理器	225
7.4.4	CardLayout 卡片布局管理器	227
7.4.5	BoxLayout 盒式布局管理器与 Box 类	229
	习题	232
第 8 章	文件、流和输入输出技术	233
8.1	I/O 流概述	233
8.2	I/O 流类	234
8.2.1	字符流	235
8.2.2	字节流	236
8.2.3	标准输入输出	237
8.3	过滤流与流的串接	239
8.4	文件流类与文件操作	241
8.4.1	File 类	241
8.4.2	FileInputStream、FileOutputStream 类与读写二进制文件	244
8.4.3	RandomAccessFile 类	249
8.4.4	FileReader、FileWriter 类与读写文本文件	250
8.5	对象序列化	253

8.5.1 对象流.....	253
8.5.2 序列化举例.....	254
习题.....	256
第9章 多媒体与多线程.....	257
9.1 使用 Graphics2D 类绘图.....	257
9.1.1 Java 2D 图形	257
9.1.2 绘制 Java 2D 图形	258
9.2 图像操作	259
9.2.1 图像创建、加载和显示	260
9.2.2 图像映射.....	262
9.3 加载和播放音频剪辑	264
9.4 多线程	266
9.4.1 多线程概述.....	266
9.4.2 创建线程的方式.....	266
9.4.3 线程的生命周期及控制.....	270
9.4.4 多线程的互斥与同步.....	273
9.4.5 守护线程.....	281
9.5 动画	281
9.5.1 使用 Timer 类显示动画	281
9.5.2 使一系列图像成为动画.....	285
习题.....	289
第10章 利用 JDBC 访问数据库	290
10.1 数据库与 JDBC 概述	290
10.1.1 数据库概述	290
10.1.2 SQL	291
10.1.3 什么是 JDBC	293
10.1.4 JDBC 与 ODBC	294
10.1.5 ODBC 数据源设置	294
10.2 通过 JDBC 访问数据库	296
10.2.1 JDBC 数据库驱动程序	296
10.2.2 JDBC API	298
10.2.3 编写 JDBC 程序的一般步骤	302
10.3 ATM 应用程序	303
习题	309
第11章 Java 网络程序设计	310
11.1 Java 与网络通信	310

11.2	URL 通信	312
11.2.1	InetAddress 类	313
11.2.2	URL 类	314
11.2.3	URLConnection 类	316
11.3	TCP Socket 通信	317
11.4	Datagram 通信	324
	习题	329
第 12 章	Java 项目实训	330
12.1	设计要求	330
12.2	功能模块设计	330
12.2.1	功能模块划分	330
12.2.2	概要设计	331
12.3	详细设计与实现	332
12.3.1	系统登录	332
12.3.2	主窗体的设计与实现	335
12.3.3	内部窗体的设计与实现	336
12.3.4	“增加、删除、修改”功能的设计与实现	342
12.4	项目测试与总结	348
12.4.1	项目测试	348
12.4.2	项目总结	349
附录 A	Java 支持的编码集	351
附录 B	应用程序的打包与发布	352
附录 C	实验	355
附录 D	Java 课程设计	361
附录 E	模拟考试题	365
附录 F	模拟考试题参考答案	369
附录 G	参考课时安排	372
	参考文献	373

第1章

Java概述

Java 是 Sun 公司(2009 年 4 月 20 日美国 Oracle 公司收购了 Sun 公司)推出的新一代面向对象程序设计语言,具有安全、体系结构中立、面向对象、简单、稳定、解释执行、多线程等显著特点,是目前使用最为广泛的网络编程语言之一。

1.1 Java 的诞生与发展

Java 源于 Sun 公司的一个名为 Green 的项目,该项目的早期成员是 Patrick Naughton、James Gosling 和 Mike Sheridan,现在大家较为熟悉的成员是 James Gosling,其最初的开发目的是为家用消费电子产品开发一个分布式代码系统,通过把 E-mail 发给电冰箱、电视机等家用电器来实现对它们的控制。在开始的时候准备采用 C++,但 C++ 太复杂,且安全性差,最后基于 C++ 开发了一种新的语言 Oak(Java 的前身)。Oak 是一种用于网络的精巧、安全的语言,Sun 公司曾以该语言投标一个交互式电视项目,但结果被 SGI 打败,恰巧这时 Mark Andreessen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员,他们用 Java 编制了 HotJava 浏览器,得到了 Sun 公司首席执行官 Scott McNealy 的支持,触发了 Java“进军”Internet。Java 的名称来源于印度尼西亚的一个岛屿——“爪哇”,此岛盛产咖啡。Java 应用程序标题栏默认图标是一杯热气腾腾的咖啡,这就是 Java 程序的标志,其寓意是为世人端上一杯热气腾腾的咖啡。

Java 的快速发展得益于 Internet 和 Web 的出现,Internet 上有各种不同的计算机,它们可能使用完全不同的操作系统和 CPU 芯片,但仍希望运行相同的程序,Java 的出现标志着真正的分布式系统的出现。

自从 1995 年 Sun 公司正式发布 Java 1.0 版以来,在全球范围内引发了经久不衰的 Java 热潮,Java 的版本依次更新为 v1.1、v1.2、v1.3、v1.4、v1.5(5.0 版本)、v1.6(6.0 版本)、v1.7(7.0 版本)、v1.8(8.0 版本),其内容也有了巨大的改进和扩充。其中,Java SE 8 版本新增了 Lambda 表达式和 Nashorn 引擎。新增 Lambda 表达式使 Java 更易于编写多核处理器代码,新增 Nashorn 使 Java 程序可以和 JavaScript 代码互操作。

1999 年,Sun 公司重新组织 Java 平台的集成方法,加强了 Java 企业级应用平台的功能。目前,Java 程序支持智能消费型电子产品的开发、各种应用程序的开发,尤其是网络程序的开发,Java 语言拥有“互联网上的世界语”的美称。

Java 不仅是一种编程语言,还是一个开发运行平台。目前,Java 平台划分为 Java SE、

Java EE 和 Java ME 共 3 个平台,针对不同的市场目标和设备进行定位。

(1) Java SE(Java Platform, Standard Edition): Java 标准版。利用该平台可以开发 Java 桌面应用程序和低端的服务器应用程序,也可以开发 Java Applet 小应用程序。用户在学习 Java 的过程中,主要用 J2SE 进行程序的开发。

(2) Java EE(Java Platform, Enterprise Edition): Java 企业版。Java EE 定义了基于组件的多层企业级应用的开发标准,面向企业级和高端服务器的 Internet 应用开发。Java EE 是基于 Java SE 的,主要包括 Enterprise JavaBeans(EJB)、Java Servlets API 以及 Java Server Pages(JSP)等技术,并为企业级应用的开发提供了各种服务和工具。

(3) Java ME(Java Platform, Micro Edition): Java 微型版。该平台是用于嵌入式应用的 Java 平台,它是为无线电子市场设计的,被应用在各种各样的电子消费产品上,例如智能卡、手机、PDA 和电视机顶盒等产品。

1.2 Java 的特点

Java 的特点主要表现在简单、面向对象、垃圾自动回收机制、分布式计算、稳定、安全、解释执行、体系结构中立、可移植性、多线程以及异常处理等方面。

1. 简单性

Java 的语法规则和 C/C++ 类似,所以熟悉 C/C++ 的编程人员较容易掌握 Java 的编程技术。Java 摒弃了 C++ 中容易引发程序错误的概念,例如指针、运算符重载、多重继承等。通过 Java 提供的丰富类库大大简化了 Java 程序的开发。Java 对环境要求非常低,它的基本解释器及类的支持只有几万字节,加上标准类库和线程的支持也只有二十多万字节。

2. 面向对象

Java 是一种纯面向对象的程序设计语言。面向对象的程序设计思想是对面向过程程序设计思想的一次革命,是一种以数据(对象)及其接口为中心的程序设计技术,面向对象程序设计可以说是定义程序模块如何“即插即用”的机制。与以前的结构化程序设计相比,它能够更好地模拟现实世界,有利于开发人员对应用程序的构思。Java 的面向对象机制实际上可以看作是 C++ 面向对象机制的延伸。Java 提供了简单的类机制和动态的构架模型,对象中封装了它的数据成员(状态变量)和成员方法(函数、过程),实现了模块化和信息隐藏;而类是创建一组相似对象的一种模板,通过继承、重载和覆盖机制,子类可以使用或者重新定义父类(也称为超类)所提供的成员,从而实现代码的复用。

3. 垃圾自动回收机制

在程序的执行过程中,部分内存使用后处于闲置状态,如果不及时回收这些垃圾内存就会造成内存泄露,甚至导致系统崩溃。在 C++ 中,由程序员负责内存的回收,内存管理增加了程序员的工作量。而在 Java 运行环境中,始终存在着一个系统级的线程,专门跟踪内存的使用情况,定期检测不再使用的内存并进行自动回收。Java 的垃圾自动回收机制在简化程序开发的同时提高了程序的稳定性和可靠性。

4. 分布式计算

Java 具有支持分布式计算的特征。在分布式计算中,“分布”具有两层含义:一是数据分布,即应用系统所操作的数据可以分散地存储在不同的网络节点上;二是操作分布,即应用系统的计算可由不同的网络节点完成。相应地,Java 实现两种层次上的分布。

(1) 数据分布支持:通过 Java 的 URL 类,Java 程序可以访问网络上的各类信息资源,其访问方式完全类似于本地文件系统。

(2) 操作分布支持:Java 通过嵌入到 WWW 页面中的小应用程序(Applet)将计算从服务器分布至客户机。Applet 由 WWW 浏览器在客户端执行,从而避免了网络拥挤,提高了系统效率。

5. 健壮性

Java 非常重视及早地检查错误,Java 编译器可以查出许多其他语言在运行时才能发现的错误。Java 丢弃了其他语言中容易引起错误的某些数据类型,例如它不支持指针,从而避免了因重写内存而造成数据破坏的可能性。Java 的垃圾自动回收机制、异常处理机制和简单性等特点提高了程序的健壮性(又称鲁棒性)。

6. 安全性

Java 的设计目的是提供一个用于网络/分布式的计算环境,因此 Java 强调安全性,例如确保无病毒、小应用程序运行的安全控制等。Java 的验证技术是以公钥(public key)加密算法为基础,而且从环境变量、类加载器、文件系统、网络资源和名字空间等方面实施安全策略。Java 程序本身不操纵内存,这减少了内存出错的可能性,而让操作系统管理内存,这有效地避免了恶意代码的破坏和病毒的入侵,极大地提高了在网络环境下运行应用程序的安全性,同时也大大减轻了程序员的负担。

7. 解释执行

Java 解释器(interpreter)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时的信息,使得执行过程更加简单。

8. 体系结构中立(平台无关性)

网络上充满了各种不同类型的计算机和操作系统,为了使 Java 程序能够在网络的任何地方运行,Java 编译器生成了与体系结构无关(即体系结构中立(architecture neutral))的字节码文件(又称类文件),可以在所有提供 Java 虚拟机(Java Virtual Machine,JVM)的多种不同主机、不同处理器上运行。

9. 可移植性

Java 解释器生成与体系结构无关的字节码文件(*.class),只要是安装了 Java 虚拟机的计算机都可以执行这种字节码文件,Java 虚拟机把这种字节码文件翻译成所在系统的机器代码并执行。因此,Java 具有高度的可移植性,实现了其他程序设计语言所不具备的“一

次编写,到处运行”(write once,run everywhere)的平台无关性。使用 Java 开发的系统的移植工作几乎为零,一般情况下只需要对配置文件、批处理文件进行相应修改即可实现平滑移植。

10. 多线程

线程是比进程更小的可并发执行的单位。在 C++ 语言中没有内置的多线程 (multithread) 机制,因此必须调用操作系统的多线程功能来进行多线程程序设计。而 Java 语言却提供了多线程支持。Java 语言在两个方面支持多线程:一方面是 Java 环境本身就是多线程的,若干个运行的系统线程负责必要的无用单元回收、系统维护等系统级操作;另一方面是 Java 语言内置的多线程机制可以大大简化多线程应用程序的开发。同时,Java 语言的线程还包括一组同步原语,这些原语负责对线程进行并发控制。利用 Java 语言的多线程 API,开发人员可以方便地写出支持多线程的应用程序,从而提高程序的执行效率。

11. 高性能

虽然 Java 是解释执行的,但是它仍然具有非常高的性能,在一些特定的 CPU 上,Java 字节码可以快速地转换成机器码来执行。而且 Java 字节码的格式设计就是针对机器码的转换,在实际转换时相当简便,自动的寄存器分配与编译器对字节码的一些优化可使之生成高质量的代码。随着 Java 虚拟机的改进和即时编译技术的出现使得 Java 的执行速度有了更大的提高。Java 解释器在一秒钟内可调用 30 万个过程,编译目标代码的速度与 C/C++ 的性能没什么区别。

12. 动态性

Java 的设计使得它适用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且不会影响用户程序的执行,并且 Java 通过接口支持多重继承,使之比严格的类继承具有更灵活的方式和可扩展性。另外,类又是在运行时动态装载的,这就使得 Java 可以在分布环境中动态地维护程序及类库。

1.3 Java 开发运行环境的安装与配置

要实现“一次编写,到处运行”的目标,就必须提供相应的 Java 平台。学习 Java 必须从 Java SE 开始,因此,本书基于 Java SE 来学习 Java 程序设计。

Java SE 包括两部分:Java 开发工具集(Java SE Development Kit,JDK)和 Java 运行时环境(Java SE Runtime Environment,JRE)。JDK 是整个 Java 的核心,包括 Java 运行时环境、Java 开发工具及其 API、Java 基础类库。登录到 Sun 公司的网站 <http://java.sun.com> (自动跳转到 <http://www.oracle.com/technetwork/java/index.html>) 免费下载 Java SE Development Kit(jdk-8u121-windows-x64.exe),本书使用 jdk1.8.0_121 版本,大家可根据自己所用机器的操作系统免费下载不同的 JDK。

1.3.1 安装 JDK

以 Windows 平台的 JDK 安装为例,下载的安装程序是一个自解压的 jdk-8u121-windows-x64.exe 文件,执行该程序就可在 Windows 上开始安装。安装过程比较简单,当出现如图 1.1 所示的对话框时,单击“更改”按钮可以更改安装目录。还可根据实际需要选择 JDK 的安装部件,在默认情况下,可在“向导”指引下安装所有的 JDK 组件。

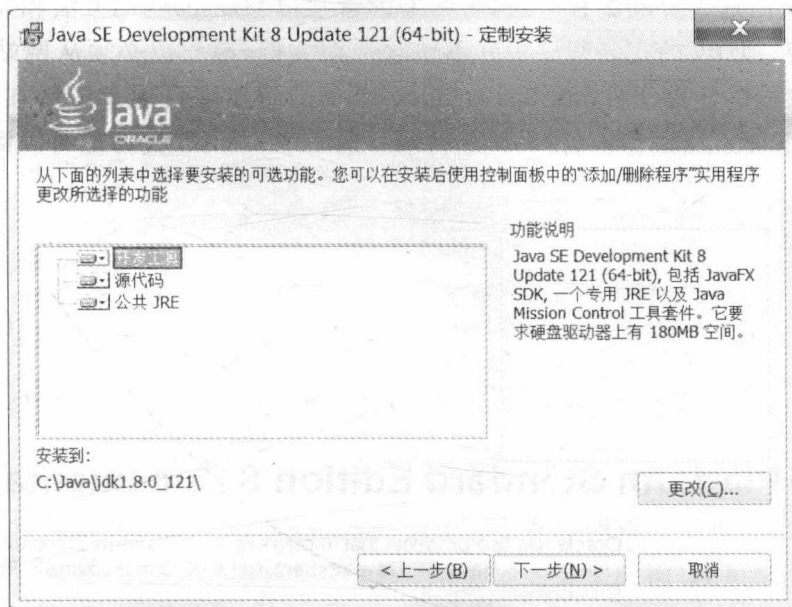


图 1.1 JDK 安装组件和目录选择对话框

下面是对安装组件的说明。

(1) 开发工具: Java 的开发工具包,用来编译、运行和打包 Java 程序。

(2) 源代码: 此组件提供 Java 类库的源码,展示了 Java 的一些核心技术。

(3) 公共 JRE: 公共的 Java 运行时环境组件,选择此组件,将在浏览器中安装最新的 Java 插件,使浏览器支持最新的 Java 类库。

安装成功后,在 JDK 文件夹下会出现 bin、db、include、jre、lib 等子文件夹(如图 1.2 所示),下面是对这些子文件夹的说明。

bin——包含编译器 javac、解释器 java、小应用程序浏览器 appletviewer 等可执行文件。

db——包含开源的 Apache Derby Java 数据库相关文件。如果没有选择安装 Java DB,则该文件夹为空。

include——存放本地文件(在 Windows 系统下是一些 C 语言头文件),支持 Java 本地接口和 Java 虚拟机调试程序接口的本地代码编程。

jre——存放 Java 运行时环境(JRE)相关的文件。

lib——存放 Java 开发工具所需的类库及其他资源文件,包括 tools.jar、dt.jar 等。

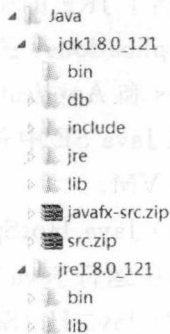


图 1.2 JDK 目录结构