



高等学校计算机教材建设立项项目

Java

程序设计 基础与实践

赵凤芝 邢 煜 段鸿轩 王 荣 编著

包 锋 主审

清华大学出版社





高等学校计算机教材建设

Java

程序设计 基础与实践

赵凤芝 邢 煜 段鸿轩 王 茱 编著

清华大学出版社
北 京

内 容 简 介

本书从初学者的角度,详细介绍了 Java 程序设计的重要技术。通过大量的实例和真实项目的讲解,能使读者快速掌握利用 Java 进行面向对象程序设计的方法和技术。本书基于 CDIO 的理念,应用“案例+项目驱动”的学习模式,力争做到通俗易懂、学以致用。全书共 13 章,分别为进入 Java 世界、Java 程序设计基础、Java 中数组的应用、面向对象程序设计基础——类与对象、面向对象程序设计高级特性、Java 实用类与接口、Java 异常处理、Java GUI(图形用户界面)设计、Java IO(输入输出)流、多线程编程、Java 网络编程、数据库程序设计和项目开发实战。

本书集作者多年的教学和科研经验编写而成,突出应用能力的培养,注重理论与实践相结合,由浅入深,讲解详尽,实例丰富,最后给出了一个真实项目案例的开发框架及实现方法,是一本实用性突出的教材。本书适合作为高等学校程序设计语言课程的教材,也可作为从事软件开发及相关领域的工程技术人员的自学参考书。

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 程序设计基础与实践/赵凤芝等编著. —北京:清华大学出版社,2017
ISBN 978-7-302-46926-1

I. ①J… II. ①赵… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 074346 号

责任编辑:张瑞庆

封面设计:何凤霞

责任校对:李建庄

责任印制:李红英

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 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:26.25

字 数:639 千字

版 次:2017 年 9 月第 1 版

印 次:2017 年 9 月第 1 次印刷

印 数:1~2000

定 价:59.00 元

产品编号:069478-01

前 言

随着经济的全球化、产业结构的调整以及“互联网+”大潮的兴起,信息产业的发展得到了前所未有的重视。在软件开发中,面向对象程序设计方法是目前的主流方法,而 Java 作为一种完全的面向对象的语言,它吸取了其他语言的优点,设计简洁而优美,使用方便而高效,具有简单、面向对象、与平台无关、解释型、多线程、动态、安全性强等特点,成为目前最为流行的程序开发语言之一,得到了普及和应用。Java 的风格接近 C++ 与 C#,特别是它的跨平台性,受到越来越多的程序设计人员的喜爱,在计算机的各种平台、操作系统,以及手机、移动设备、智能卡、家用电器均得到了广泛的应用。Java 可运行于多个平台,如 Windows、Mac OS 及其他多种 UNIX 版本的系统。Java 程序设计也是高等学校计算机及相关学科的核心专业课程,是培养学生软件设计能力的重要课程,在计算机学科的教学起着非常重要的作用。

Java 诞生于 1991 年,Sun Microsystems 公司(以下简称 Sun 公司)1995 年 6 月把 Java 这门革命性的语言引入世界。之所以称 Java 为革命性的编程语言,是因为用以前语言编写的传统的软件系统与具体的开发环境有关,一旦开发环境有所变化就需要对软件系统进行一番改动,耗时费力,而利用 Java 语言开发的软件系统能在所有装有 Java 解释器的计算机上运行。Java 出现的初衷是出于对独立于平台的需要,世人希望有门编程语言能编写出嵌入到各种家用电器等设备的芯片上且易于维护的程序。但是,人们发现当时的编程语言都针对 CPU 芯片进行编译,如 C、C++ 等,这样,一旦电器设备更换了 CPU 芯片就不能保证程序正常的运行,可能需要修改程序重新进行编译。Sun 公司经过调查发现当时编程语言的这个致命缺点后,于 1990 年成立了由 James Gosling 领导的软件开发小组,开始致力于开发一种与平台无关的编程语言。他们的刻苦钻研与努力造就了 Java 语言的诞生。从此,Java 被广泛接受并推动了 Web 的迅速发展。Java 由 Sun 公司注册,是 Sun 公司最著名的商标,也是 IT 行业最著名的商标之一。

由于计算机语言更新快,所以本书添加了目前关于 Java 5 之后引入的一些最新的知识,基于 CDIO 的工程教育理念,应用“案例+项目驱动”的学习模式,力争做到通俗易懂,学以致用。通过引导学生完成项目任务,达到使学生掌握必备知识和拓展知识的目的。

如何尽快掌握 Java 技术呢?

学习任何技术都要从基础开始,编写本书的目的是让初学者能够通过最简单的描述和说明来跨越学习 Java 的第一道门槛,提供一本零起点的面向对象程序设计的初级教程。读

者如果简单调研一下就不难发现,目前市面上最多的就是这类 Java 基础入门的教材。作者希望采用一种最有效的学习与培训的捷径和方法,这就是项目驱动训练法(Project-driven training),也就是用项目实践来带动理论的学习。基于此,本书采用“案例+项目”驱动的模式,深入浅出地全面介绍了 Java 语言程序设计的知识。通过项目实践,可以对技术应用有明确的目的性(为什么学),对技术原理更好地融会贯通(学什么),也可以更好地检验学习效果(学得怎样)。通过本书的学习,读者不仅学会了一种语言,而且能够在一定程度上掌握面向对象的思维方式,有能力编写真正有实际意义的应用程序,进行实际项目开发。

Java 语言基础是一个实践性很强的课程,我们的指导思想是在有限的时间内精讲多练,培养读者的实际动手能力、自学能力、开拓创新能力和综合解决问题能力,将专业能力的培养贯穿始终。

全书共分为 13 章:第 1 章 进入 Java 世界;第 2 章 Java 程序设计基础;第 3 章 Java 中数组的应用;第 4 章 面向对象程序设计基础——类与对象;第 5 章 面向对象程序设计高级特性;第 6 章 Java 实用类与接口;第 7 章 Java 异常处理;第 8 章 Java GUI(图形用户界面)设计;第 9 章 Java IO(输入输出)流;第 10 章 多线程编程;第 11 章 Java 网络编程;第 12 章 数据库设计;第 13 章 项目开发实战。在本书附录中,还介绍了 Java 编程规范。本书每章都有很多程序实例,每章后面配备了开发案例。

本书通过浅显易懂的实例引导初学者循序渐进地学习 Java 程序设计语言。本书通俗易懂,内容丰富,结构合理,注重理论与实践相结合,针对性强,突出应用能力的培养。本书结构设计独特,每章均配有典型案例和习题,与每章知识点相辅相成,简单易学,使初学者更容易掌握,是一本实用性很强的教材。本书所有例题都在 Java SE8 环境下编译通过并成功运行。本教材配套的教学资源,读者可以从清华大学出版社网站 www.tup.com.cn 免费下载。

本书由赵凤芝、邢煜、段鸿轩、王莱编著,包锋主审。其中,第 1、2、3、4 章由赵凤芝编写,第 5、6、7 章由王莱编写,第 8、9、10 章由邢煜编写,第 11、12、13 章及附录由段鸿轩编写,全书由赵凤芝负责组织编写、统稿和定稿,包锋负责审校。本书的编写得到了梁立新及相关人员的大力支持和帮助,在此一并表示感谢!鉴于作者的水平有限,书中难免有不足之处,敬请广大读者批评指正。

作 者

2017 年 5 月

目 录

第 1 章 进入 Java 世界	1
1.1 初识 Java	1
1.1.1 Java 语言的诞生与发展	1
1.1.2 Java 语言的特点	2
1.1.3 Java 应用开发体系	4
1.2 面向对象与程序设计语言	5
1.3 学习 Java 技术可以做什么	6
1.4 Java 核心技术体系	7
1.4.1 Java 核心技术基础部分	7
1.4.2 Java 核心技术应用部分	8
1.5 Java 的开发环境	11
1.5.1 什么是 JDK	11
1.5.2 下载 JDK	12
1.5.3 完成安装 JDK	13
1.5.4 系统环境配置	15
1.5.5 测试 JDK 配置是否成功	17
1.5.6 开发工具 Eclipse 简介	17
1.6 简单的 Java 程序	24
本章总结	29
习题	30
第 2 章 Java 程序设计基础	31
2.1 Java 的基本语法	31
2.1.1 Java 的标识符与关键字	31
2.1.2 Java 中的注释	34
2.1.3 Java 中的常量和变量	37
2.1.4 Java 的数据类型	38

2.2	Java 的运算符与表达式	48
2.2.1	算术运算符和算术表达式	48
2.2.2	赋值运算符和赋值表达式	52
2.2.3	关系运算符和关系表达式	53
2.2.4	逻辑运算符和逻辑表达式	55
2.2.5	位运算符	56
2.2.6	条件运算符和条件表达式	59
2.2.7	表达式中运算符的优先次序	60
2.3	Java 流程控制	61
2.3.1	顺序流程	61
2.3.2	分支流程	61
2.3.3	循环控制流程	70
2.4	项目案例	78
2.4.1	学习目标	78
2.4.2	案例描述	78
2.4.3	案例要点	78
2.4.4	案例实施	78
2.4.5	特别提示	82
	本章总结	82
	习题	82

第 3 章 Java 中数组的应用 85

3.1	什么是数组	85
3.2	一维数组	85
3.3	一维数组的应用	88
3.4	二维数组与多维数组	92
3.5	二维数组的应用	95
3.6	项目案例	98
3.6.1	学习目标	98
3.6.2	案例描述	99
3.6.3	案例要点	99
3.6.4	案例实施	99
3.6.5	特别提示	102
3.6.6	拓展与提高	102
	本章总结	102
	习题	102

第 4 章 面向对象程序设计基础——类和对象	104
4.1 面向对象的基本概念	104
4.1.1 面向对象程序设计思想	104
4.1.2 面向对象程序设计方法特点	105
4.2 对象与类	105
4.2.1 日常生活中看对象与类的关系	105
4.2.2 成员	106
4.3 面向对象的 4 个基本特征	107
4.3.1 继承性	107
4.3.2 抽象性	107
4.3.3 封装性	107
4.3.4 多态性	108
4.4 Java 实现面向对象程序设计	108
4.4.1 类的定义与对象的创建	108
4.4.2 命名的规则	109
4.5 类的成员——变量	110
4.5.1 变量属性的修饰符	110
4.5.2 变量的初始化	112
4.5.3 对成员变量的访问	113
4.6 类的成员——方法	115
4.6.1 方法定义	115
4.6.2 方法的调用及参数传递	116
4.6.3 Java 新特性——可变参数(Varargs)	117
4.6.4 构造方法	118
4.6.5 方法的重载	120
4.7 对象资源的回收	122
4.7.1 垃圾对象	122
4.7.2 finalize()方法	123
4.8 项目案例	124
4.8.1 学习目标	124
4.8.2 案例描述	124
4.8.3 案例要点	124
4.8.4 案例实施	125
4.8.5 特别提示	130
4.8.6 拓展与提高	130
本章总结	131
习题	132

第 5 章 面向对象程序设计高级特性

134

5.1	继承和多态	134
5.1.1	继承的概念	134
5.1.2	继承的实现	135
5.1.3	成员变量隐藏	137
5.1.4	方法覆盖	138
5.1.5	继承中的构造方法调用	139
5.1.6	多态性	142
5.2	抽象方法与抽象类	144
5.2.1	抽象方法	144
5.2.2	抽象类	144
5.2.3	扩展抽象类	145
5.3	接口	146
5.3.1	接口的定义	146
5.3.2	接口的实现	146
5.3.3	引用类型的转换	147
5.4	包	148
5.4.1	包及其使用	149
5.4.2	访问控制	150
5.5	内部类	151
5.5.1	认识内部类	151
5.5.2	成员式内部类——对象成员内部类	152
5.5.3	成员式内部类——静态内部类	154
5.5.4	局部内部类	156
5.5.5	匿名内部类	157
5.6	项目案例	158
5.6.1	学习目标	158
5.6.2	案例描述	159
5.6.3	案例要点	159
5.6.4	案例实施	159
5.6.5	特别提示	163
5.6.6	拓展与提升	163
	本章总结	163
	习题	164

第 6 章 Java 实用类与接口	168
6.1 Object 类	168
6.2 字符串处理	171
6.2.1 String 类	171
6.2.2 StringBuilder	178
6.2.3 StringTokenizer(字符串标记)	184
6.3 基本类型的封装类	186
6.4 System 与 Runtime 类	187
6.4.1 System 类	187
6.4.2 Runtime 类	188
6.5 集合框架	189
6.5.1 Collection 接口	190
6.5.2 Set 接口	191
6.5.3 List 接口	193
6.5.4 Iterator 接口	195
6.5.5 Map 接口	197
6.6 泛型	199
6.7 时间及日期处理	203
6.7.1 Date 类	203
6.7.2 Calendar 类	205
6.7.3 DateFormat 类	207
6.7.4 SimpleDateFormat 类	208
6.8 算术实用类	210
6.8.1 Math 类	210
6.8.2 Random 类	211
6.9 枚举	213
6.10 Annotation	217
6.11 Lamda 表达式	218
6.12 项目案例	220
6.12.1 学习目标	220
6.12.2 案例描述	220
6.12.3 案例要点	220
6.12.4 案例实施	220
6.12.5 特别提示	226
6.12.6 拓展与提高	226
本章总结	226
习题	227

第 7 章 Java 异常处理**229**

7.1 异常处理概述	229
7.1.1 程序中错误	230
7.1.2 异常定义	230
7.2 异常分类	231
7.3 异常处理	233
7.3.1 如何处理异常	233
7.3.2 处理异常的基本语句	233
7.4 自定义异常	239
7.5 项目案例	240
7.5.1 学习目标	240
7.5.2 案例描述	240
7.5.3 案例要点	240
7.5.4 案例实施	240
7.5.5 特别提示	241
7.5.6 拓展与提高	241
本章总结	243
习题	243

第 8 章 Java GUI(图形用户界面)设计**246**

8.1 GUI 程序概述	246
8.1.1 AWT 简介	246
8.1.2 Swing 简介	247
8.2 容器与布局	248
8.2.1 容器	248
8.2.2 布局管理	249
8.3 常用组件	258
8.3.1 AWT 组件	258
8.3.2 Swing 组件	264
8.4 事件处理	272
8.4.1 事件处理的概念	272
8.4.2 监听器和适配器	273
8.4.3 事件处理的编程方法	283
8.5 项目案例	283
8.5.1 学习目标	283

8.5.2 案例描述	283
8.5.3 案例要点	283
8.5.4 案例实施	283
8.5.5 特别提示	286
8.5.6 拓展与提高	286
本章总结	286
习题	287

第 9 章 Java IO(输入输出)流 288

9.1 输入输出流的概述	288
9.1.1 流的概念	288
9.1.2 字节流	289
9.1.3 字符流	290
9.2 java.io 包层次结构	291
9.3 常用输入输出类	295
9.3.1 常用输入类	295
9.3.2 常用输出类	297
9.3.3 转换流	302
9.4 文件和目录的操作	303
9.5 对象流和对象序列化	309
9.5.1 序列化概述	309
9.5.2 序列化实现机制	309
9.6 项目案例	311
9.6.1 学习目标	311
9.6.2 案例描述	312
9.6.3 案例要点	312
9.6.4 案例实施	312
9.6.5 特别提示	315
9.6.6 拓展与提高	315
本章总结	316
习题	316

第 10 章 多线程编程 317

10.1 线程概念	317
10.2 线程的创建及启动	318
10.3 线程状态及转化	323

10.4	线程优先级及调度策略	328
10.5	线程同步与互斥	329
10.5.1	基本概念	329
10.5.2	线程同步	331
10.6	项目案例	335
10.6.1	学习目标	335
10.6.2	案例描述	336
10.6.3	案例要点	336
10.6.4	案例实施	336
10.6.5	特别提示	339
10.6.6	拓展与提高	339
	本章总结	340
	习题	340

第 11 章 Java 网络编程 341

11.1	网络编程概述	341
11.2	理解 TCP/IP 及 UDP/IP 协议	343
11.3	使用 Socket 开发 TCP/IP 程序	343
11.4	使用 Socket 开发 UDP/IP 程序	350
11.5	项目案例	353
11.5.1	学习目标	353
11.5.2	案例描述	353
11.5.3	案例要点	353
11.5.4	案例实施	354
11.5.5	特别提示	362
11.5.6	拓展与提高	362
	本章总结	362
	习题	362

第 12 章 数据库程序设计 363

12.1	关系数据库简介	363
12.2	JDBC 简介	364
12.3	准备数据库环境	364
12.4	JDBC 开发流程	366
12.5	项目案例	370
12.5.1	学习目标	370

12.5.2	案例描述	370
12.5.3	案例要点	370
12.5.4	案例实施	370
12.5.5	特别提示	372
12.5.6	拓展与提高	372
本章总结		372
习题		372
第 13 章	项目开发实战	373
13.1	问题描述	373
13.2	需求分析	373
13.3	概要设计	374
13.3.1	数据库设计	374
13.3.2	接口设计	375
13.4	代码实现	375
13.4.1	PersonVO 类的实现	376
13.4.2	DBConnection 类的实现	377
13.4.3	IPersonDAO 接口的实现	378
13.4.4	PersonDAOImpl 类的实现	379
13.4.5	PersonDAOProxy 类的实现	383
13.4.6	DAOFactory 类的实现	385
13.4.7	MainMenu 类的实现	386
13.4.8	InputHandler 类的实现	387
13.4.9	PersonAction 类的实现	388
13.4.10	StartApp 类的实现	391
本章总结		392
习题		392
附录	Java 编程规范	393
参考文献		405

本章重点

- Java 语言的发展及特点。
- 面向对象概念及 Java 的核心技术体系。
- Java 的开发环境的搭建。
- 简单的 Java 程序的设计与运行。

Java 是面向对象、安全、跨平台、强大稳健、非常流行的高级程序设计语言。它的风格接近 C++ 与 C#, 特别是它的跨平台性, 受到越来越多的程序设计人员的喜爱, 在计算机的各种平台、操作系统, 以及手机、移动设备、智能卡、家用电器等领域均得到了广泛的应用。Java 可运行于多个平台, 如 Windows、Mac OS 以及其他多种 UNIX 版本的系统。

学习 Java 语言首先要了解 Java 语言, 通过本章的学习将能够了解 Java 语言的发展及特点, 了解 Java 语言是纯面向对象的程序设计语言, 了解其技术体系, 熟悉其开发环境, 学会简单 Java 程序的设计与运行。

1.1 初识 Java

1.1.1 Java 语言的诞生与发展

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台的总称。用 Java 实现的 HotJava 浏览器(支持 Java applet)显示了 Java 的魅力: 跨平台、动态的 Web、Internet 计算。因此, Java 被广泛地接受并推动了 Web 的迅速发展。Java 由 Sun Microsystems 公司注册, 是 Sun Microsystems 公司最著名的商标, 也是 IT 行业最著名的商标之一。

1991 年 4 月, Sun Microsystems 公司启动了由 Java 编程语言的创始人——James Gosling(詹姆斯·高斯林, 出生于加拿大, 是一位计算机编程天才, 也被称为 Java 之父, 见图 1-1)等发起的名为 Green 研究项目, 最初研究的目的是创建一种与平台无关的、可用于交互手持式家庭设备控制器(如用于控制嵌入在有线电视交换盒)的语言, 以实现一些家庭娱乐设备和家用电器的控制功能。James Gosling 称这种新语言为 Oak, 后更名为 Java, 应用于网络, 并沿用至今。然而, Green 项目遇

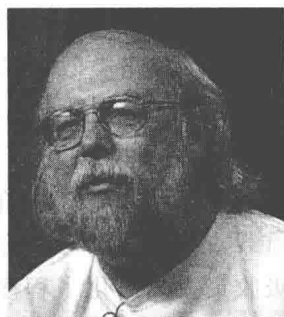


图 1-1 Java 创始人詹姆斯·高斯林

到了困难,市场前景并不乐观。

1994 年,Internet 开始在全球盛行。从此,计算机世界发生了重大的变革。Internet 是世界上最大的客户机/服务器系统,它拥有千万种不同类型的客户机。显然,Web 设计者无法做到对可能访问其页面的每一台计算机编写不同的程序,而 Java 技术正是独立于平台而设计的。这使 Green 项目组的成员意识到,Java 完全符合在 Internet 上编写、发送和使用应用程序的方式。

1995 年,Sun Microsystems 公司正式发布 Java 语言,Microsoft、IBM、NETSCAPE、NOVELL APPLE、DEC 和 SGI 等公司纷纷购买 Java 语言的使用权。

1996 年,Sun Microsystems 公司正式发布了 Java 语言的第一个非试用版本。

1999 年 11 月启用 Java2。

2004 年 9 月,J2SE1.5 发布,这是 Java 语言发展史上的又一里程碑事件。为了表示这个版本的重要性,J2SE1.5 更名为 J2SE5.0。

2005 年 6 月,JavaOne 大会召开,Sun Microsystems 公司公开 Java SE 6。此时,Java 的各种版本已经更名以取消其中的数字 2: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

2009 年 4 月,Oracle(甲骨文)公司宣布收购 Sun Microsystems 公司。

2011 年,Oracle 公司发布 Java7 正式版。

2014 年,Oracle 公司发布了 Java8 正式版。

迄今为止,Java 技术已经非常成熟,同时也不再使用 Java2 的称呼方法,而直接称为 Java。在计算机发展史上,Java 语言的发展速度之快是空前的。

Java 的官方网站是 <http://www.oracle.com/technetwork/java/index.html>。Java 分为 Java 运行时环境 JRE 和 Java 开发工具包 JDK,均可以在其官方网站下载。

1.1.2 Java 语言的特点

Java 程序设计语言是新一代语言的代表,它强调了面向对象的特性,可以用来开发不同种类的软件,它具有支持图形化的用户界面、支持网络以及数据库连接等复杂的功能。Java 语言主要有以下特点。

1. 简单、易于学习

Java 语言很简单。Java 语言的简单性主要体现在以下三个方面:

(1) Java 的风格类似于 C++,因为它的语法和 C++ 非常相似,因而 C++ 程序员是非常熟悉的。从某种意义上讲,Java 语言是 C 及 C++ 语言的一个变种,因此,C++ 程序员可以很快地掌握 Java 编程技术。

(2) Java 摒弃了 C++ 中许多低级、困难、容易混淆、容易出错或不经常使用的功能,例如运算符重载、指针运算、程序的预处理、结构、多重继承以及其他一系列内容,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,这样更有利于 Java 语言初学者的学习。

(3) Java 提供了丰富的类库,为程序设计提供了方便的条件。

2. 面向对象

面向对象可以说是 Java 最重要的特性。Java 语言的设计完全是面向对象的,它不支持类似 C 语言那样的面向过程的程序设计技术。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。

3. 分布式

Java 语言支持 Internet 应用的开发,在基本的 Java 应用编程接口中有一个网络应用编程接口(java.net 包),它提供了用于网络应用编程的类库,包括 URL、URLConnection、Socket、ServerSocket 等。Java 的 RMI(远程方法调用)机制也是开发分布式应用的重要手段。

4. 高性能

用 Java 语言编辑的源程序的执行方法是采用先经过编译器编译、再利用解释器解释的方式来运行的。它综合了解释性语言与编译语言的众多优点,使其执行效率较以往的程序设计语言有了大幅度的提高。如果解释器速度不慢,Java 可以在运行时直接将目标代码翻译成机器指令。Sun 公司用直接解释器一秒钟内可调用 300 000 个过程。翻译目标代码的速度与 C/C++ 的性能没什么区别。

5. 安全性

Java 通常被用在网络环境中,为此,Java 提供了一个安全机制以防恶意代码的攻击。除了 Java 语言具有的许多安全特性以外,Java 对通过网络下载类具有一个安全防范机制(类 ClassLoader),如分配不同的名字空间以防替代本地的同名类、字节代码检查,并提供安全管理机制(类 SecurityManager)让 Java 应用设置安全哨兵。

6. 多线程

Java 的多线程机制使应用程序中的线程能够并发执行,且其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,这样就很容易在网络上实现实时交互行为。Java 的多线程功能使得在一个程序里可同时执行多个小任务。线程有时也称小进程,是一个大进程里分出来的小的、独立的进程。在 Java 语言中,线程是一种特殊的对象,它必须由 Thread 类或其子(孙)类来创建。通常有两种方法来创建线程:一种方法是使用 Thread(Runnable)的构造方法将一个实现了 Runnable 接口的对象包装成一个线程;另一种方法是从 Thread 类派生出子类并重写 run 方法,使用该子类创建的对象即为线程。值得注意的是,Thread 类已经实现了 Runnable 接口,因此,任何一个线程均有它的 run 方法,而 run 方法中包含了线程所要运行的代码。线程的活动由一组方法来控制。Java 语言支持多个线程的同时执行,并提供多线程之间的同步机制(关键字为 synchronized)。

7. 可移植性(与平台无关性)

Java 源程序经过编译器编译,会被转换成一种称为字节码(byte-code)的目标程序。字