



高等院校计算机专业系列规划教材

GAODENG YUANXIAO JISUANJI ZHUANYE XILIE GUIHUA JIAOCAI

Java语言程序设计

朱晓龙 主 编
刘智斌 王 博 朱 辉 副主编

Java YUYAN
CHENGXU SHEJI



北京邮电大学出版社
www.buptpress.com

高等院校计算机专业系列规划教材

Java 语言程序设计

朱晓龙 主 编

刘智斌 王 博 朱 辉 副主编

北京邮电大学出版社
· 北 京 ·

内 容 简 介

本书从类与对象的概念入手,介绍了Java面向对象的程序设计。内容主要包括:类与对象、继承、接口、多态、内部类、异常处理、GUI设计、Applet、多线程、输入输出流、网络编程等。旨在帮助读者建立面向对象的思维,掌握面向对象的编程技术。

本书既注重理论的讲解,又强调实践技能的培养。通过案例介绍理论,说明编程的方法和规范。设置模仿、改写和编写程序的环节,便于读者巩固与提高。并提供案例的源代码和教材的PPT电子教案。

本书可作为大专院校各专业的Java教材,也可作为计算机等级考试和Java语言自学者的参考用书。

图书在版编目(CIP)数据

Java语言程序设计/朱晓龙主编. --北京:北京邮电大学出版社,2011.4

ISBN 978-7-5635-2555-3

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

中国版本图书馆CIP数据核字(2011)第007219号

书 名: Java语言程序设计

主 编: 朱晓龙

责任编辑: 陈岚岚 张珊珊

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路10号(邮编:100876)

发 行 部: 电话:010-62282185 传真:010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京忠信诚胶印厂

开 本: 787 mm×1 092 mm 1/16

印 张: 21

字 数: 548千字

印 数: 1—3 000册

版 次: 2011年4月第1版 2011年4月第1次印刷

ISBN 978-7-5635-2555-3

定 价: 38.50元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前 言

面向对象的程序设计方法是目前软件开发的主流技术。Java 语言是由美国 Sun 公司开发的,是一种完全面向对象的程序设计语言。它具有面向对象、跨平台、安全、健壮、分布式等特点。由于 Java 语言从根本上解决了 Internet 的异质、代码交换以及安全性等诸多问题,因此,Java 语言非常适合于企业网络和 Internet 环境下的应用开发,自 1995 年诞生以来,很快就风靡全球,现已成为 Internet 中最受欢迎、最有影响力的程序设计语言。Java 语言已经成为最卓越的程序设计语言之一,并进入了主流计算模式,对传统的计算模式提出了新的挑战,对整个计算机软件行业的发展产生了极其重大而深远的影响。

1999 年,Sun 公司把 Java 技术分成 J2SE、J2EE 和 J2ME,2005 年分别更名为 Java SE、Java EE 和 Java ME。其中,Java SE 为创建和运行 Java 程序提供了最基本的环境;Java EE 和 Java ME 建立在 Java SE 基础上,Java EE 为分布式的企业应用提供开发和运行环境,Java ME 为嵌入式应用提供开发和运行环境。可见,不论是从事 Java EE 开发,还是从事 Java ME 开发,要想成为 Java 方面的卓越工程师,就必须具备扎实的 Java SE 知识结构。

Java 语言的简单性、完全面向对象等特点,使得它非常适合大学中面向对象程序设计的教学。本书就是提供了一本零起点、全面介绍 Java SE 的知识结构体系和讲解面向对象程序设计的教程。读者只需具有一些计算机文化基础知识,就可学习本教程,而不必具有编程经验。通过学习本教程,可使读者掌握 Java 应用程序和 Java Applet 的开发方法和技术、熟练使用一种开发工具进行 Java 程序设计、学会面向对象的思维方式及其编程风格。为进一步学习各种 Java 技术,从事 Java EE 开发或 Java ME 开发打下坚实的基础。

本书以面向对象的程序设计方法为主线,全面介绍了 Java 技术的基本概念、方法及理论。第 1 章通过几个例子具体说明了类和对象的概念。第 2 章和第 3 章介绍了数据的表达和 Java 语句,强化了类和对象的概念。第 4 章和第 5 章系统地介绍了 Java 面向对象的编程方法和技能。第 6 章介绍了编程中常用的类。第 7 章介绍了数组和字符串的使用。第 8 章介绍了 Java 编程的异常处理技术。第 9 章和第 10 章介绍了基于 Swing 的图形用户界面和 Java Applet 的程序设计技术。第 11 章和第 12 章介绍了 Java 多线程和输入输出流的程序设计技术。第 13 章介绍网络编程的方法。

本书主要有以下特点。

1. 理论与实践相结合

通过精选案例力求做到深入浅出地讲述 Java 技术的基本概念、方法及理论。并通过模仿—改写—编写的实践,理解面向对象的思想,学会面向对象的编程方法。融“教、学、练”于一体,体现“在练中学,学以致用”的教学理念。

2. 注重案例难度的阶梯性

有针对 Java 基本理论而设计的基础性案例,也有针对编程技术而设计的进阶式案例。通过阶梯式案例,力求做到概念准确、结构清晰、逻辑严谨,使读者由浅入深、循序渐进地掌握 Java 编程技术。

3. 注重实用性和启发性

案例力求体现实际需求,并留有拓展空间,启发读者建立面向对象的编程思想,掌握面向对象的编程方法。

4. 注重 Java 最新进展对 Java 知识结构和核心技术的更新

全书以 Java SE6 为基础来讲解 Java 语言的知识结构和核心技术,所有实例程序都在 Java SE6 环境下调试与运行,适应读者对 Java 知识结构更新的需求。并按照适合读者学习的顺序编排内容结构,力求主题鲜明、重点突出、结构合理、深度和广度结合恰当,使读者具备利用 Java 进行程序开发的知识结构。

程序设计是一门实践性很强的课程。读者只有在学习书本内容的同时,注重上机实验环节,才能真正掌握书中介绍的知识和技能。优秀的软件工程师都有大量上机编程的经验,从实践中可学到很多书本上没有的东西。为此,本书引入了大量的例题及习题,只要读者按照书中要求,注重实验,边学边练,一定能够掌握 Java 程序设计的方法和技术。

本书可作为大专院校各专业 Java 语言程序设计课程的教材,也可作为 Java 编程人员的培训教材及自学用书。

本书由朱晓龙、刘智斌、王博和朱辉编写。其中,朱晓龙编写第 1、3、4、5、6、8、12 章;刘智斌编写第 7 章;王博编写第 9、10 章;朱辉编写第 2、11、13 章。

本书的出版得到了北京邮电大学出版社的大力支持,西安邮电学院计算机学院的陈莉君老师也为本书的编写提出了不少建议,在此一并表示深深的谢意。

在本书编写过程中,作者参阅了大量的参考文献与资料,在此谨向诸多学者表示衷心的感谢。由于作者水平有限,不当之处在所难免,敬请广大读者批评和指正。

作者电子邮箱地址:zhuxlf@163.com

朱晓龙

目 录

第 1 章 Java 概述	1
1.1 Java 语言简介	1
1.1.1 Java 语言的起源与发展	1
1.1.2 Java 语言的特点	2
1.2 Java 程序工作原理	3
1.2.1 Java 程序工作流程	3
1.2.2 Java 虚拟机	3
1.2.3 Java 语言的平台无关性	4
1.2.4 Java 平台的分类	5
1.3 Java 开发环境	5
1.3.1 Java SE 的开发工具包	6
1.3.2 Java 集成开发环境	6
1.3.3 Java 学习与帮助资源——JDK 文档	8
1.4 Java 程序概述	9
1.4.1 第一个 Java Application 程序	9
1.4.2 第一个 Java Applet 程序	10
1.5 面向对象的 Java 程序设计举例	11
1.5.1 Java Application 程序设计举例	11
1.5.2 Java JApplet 程序设计举例	15
1.6 Java 程序的结构	16
习题	17
第 2 章 数据类型、运算符和表达式	18
2.1 数据类型与变量	18
2.1.1 关键字和标识符	18
2.1.2 数据类型	19
2.1.3 常量	20
2.1.4 变量	21
2.2 运算符和表达式	22
2.2.1 赋值运算	22
2.2.2 算术运算	23

2.2.3	关系运算	24
2.2.4	位运算	25
2.2.5	逻辑运算	26
2.2.6	其他运算	27
2.3	优先级和基本数据类型的转换	27
2.3.1	运算符的优先级和结合性	27
2.3.2	自动转换	28
2.3.3	强制转换	29
2.4	引用类型和引用变量	30
2.5	Math 类的常量和方法	31
	习题	32
第 3 章	流程控制语句	34
3.1	语句概述	34
3.1.1	Java 语言的语句分类	34
3.1.2	结构化程序设计的三种基本流程	36
3.2	分支语句	36
3.2.1	if 语句	36
3.2.2	if-else 语句	37
3.2.3	switch 语句	40
3.3	循环语句	42
3.3.1	while 循环语句	42
3.3.2	do-while 循环语句	44
3.3.3	for 循环语句	46
3.4	跳转语句	50
3.4.1	break 语句	50
3.4.2	continue 语句	51
3.4.3	return 语句	53
3.5	常用的程序设计方法	53
3.5.1	枚举法(穷举法)	53
3.5.2	递推法	55
3.5.3	递归法	57
3.5.4	简单图形的输出	58
3.5.5	简单游戏的设计	60
3.6	格式化的输入与输出	61
3.6.1	从键盘输入数据	61
3.6.2	格式化的输出	63
	习题	64

第 4 章 类与对象	66
4.1 类与对象的概念	66
4.2 类的定义	67
4.2.1 类定义的基本形式	67
4.2.2 类定义的一般形式	69
4.3 对象	70
4.3.1 对象的声明	70
4.3.2 对象的创建与初始化	71
4.3.3 对象的引用和清除	72
4.4 成员变量	75
4.4.1 成员变量声明的一般形式	75
4.4.2 实例变量和类变量	75
4.4.3 变量的初始化	77
4.4.4 常量	80
4.4.5 变量的作用域	81
4.5 成员方法	82
4.5.1 成员方法声明的一般形式	83
4.5.2 实例方法与类方法	83
4.5.3 构造方法	87
4.5.4 方法重载	88
4.5.5 数据传递	90
4.6 类的组织方式——包	93
4.6.1 包的概念	93
4.6.2 包的创建	94
4.6.3 包的引入	95
4.6.4 静态成员的引入	97
4.7 访问权限	98
4.7.1 类的访问权限	98
4.7.2 类中成员的访问权限	99
4.8 内部类	102
4.8.1 成员内部类	103
4.8.2 局部内部类	106
4.9 可变参数	107
4.9.1 可变参数的定义与使用	107
4.9.2 可变参数的重载	109
4.9.3 可变参数与构造方法	110
习题	110

第 5 章 继承与多态	112
5.1 继承的概念	112
5.2 继承的实现	114
5.2.1 派生子类	114
5.2.2 变量的隐藏	115
5.2.3 方法的覆盖	117
5.3 子类的构造方法	118
5.3.1 有继承时的构造方法	118
5.3.2 关键字 null、this 和 super	122
5.4 抽象类和最终类	124
5.4.1 抽象方法与抽象类	125
5.4.2 最终类	126
5.5 接口	127
5.5.1 接口的定义	128
5.5.2 接口的实现	129
5.5.3 接口的继承与组合	131
5.6 多态	132
5.6.1 引用类型的转换	133
5.6.2 方法覆盖实现的多态	134
5.6.3 方法重载实现的多态	135
5.7 引用类型变量实现的多态	136
5.7.1 引用类型变量是如何实现多态的	136
5.7.2 引用类型转换实现多态	137
5.7.3 引用类型作参数实现多态	139
5.7.4 引用类型作成员变量实现多态	139
5.8 匿名内部类	140
5.8.1 继承类的匿名内部类	141
5.8.2 实现接口的匿名内部类	143
习题	144
第 6 章 Java 常用类	145
6.1 Object 类	145
6.2 System 类	147
6.3 基本类型的包装类	149
6.3.1 包装类对象的创建方式	149
6.3.2 包装类的常用方法	150
6.3.3 自动装箱和自动拆箱	150
6.4 日期类	153
6.4.1 Date 类和 DateFormat 类	153

6.4.2	Calendar 类	154
6.5	随机数类 Random	156
6.6	枚举类型	156
6.6.1	枚举类型的定义及其使用	156
6.6.2	枚举类型继承的方法	158
6.6.3	枚举类型的扩展和方法覆盖	159
6.6.4	枚举类型的构造方法	160
6.7	泛型	161
6.7.1	泛型类和泛型接口	161
6.7.2	泛型方法	164
6.7.3	受限类型参数	165
6.7.4	泛型的子类型与通配符	166
6.7.5	泛型的擦除	168
	习题	169
第 7 章	数组与字符串	170
7.1	数组	170
7.1.1	数组的创建	170
7.1.2	数组元素的赋值及其初始化	171
7.1.3	多维数组	175
7.2	字符串	176
7.2.1	String 类	176
7.2.2	StringBuffer 类	180
7.2.3	StringTokenizer 类	182
	习题	183
第 8 章	异常处理	184
8.1	异常概述	184
8.1.1	异常的概念	184
8.1.2	异常处理机制	185
8.2	异常类	186
8.2.1	异常类的分类	186
8.2.2	系统定义的异常类	187
8.2.3	用户自定义的异常类	188
8.2.4	受检查的异常和不受检查的异常	190
8.3	异常的抛出	191
8.3.1	系统自动抛出的异常	191
8.3.2	throw 语句抛出的异常	191
8.4	异常的捕获与处理	193
8.4.1	try/catch 语句	194

8.4.2 多异常的捕获和处理	196
8.5 异常处理的进一步讨论	198
8.5.1 没有捕获到的异常	198
8.5.2 catch 子句抛出的异常	199
8.5.3 构造方法抛出的异常	200
8.5.4 finally 子句	201
8.6 总结与提高	203
8.6.1 异常处理机制和编程方法总结	203
8.6.2 银行取款的异常处理	204
8.6.3 职员年龄的异常处理	205
习题	206
第 9 章 图形用户界面	208
9.1 图形用户界面概述	208
9.1.1 javax.swing 包	208
9.1.2 类的层次结构	209
9.1.3 一个 GUI 程序	209
9.1.4 容器、组件、布局和观感	210
9.2 常用 Swing 容器和组件	211
9.2.1 常用 Swing 容器	211
9.2.2 常用 Swing 组件	211
9.3 布局管理器	213
9.3.1 边界布局管理器	213
9.3.2 流式布局管理器	214
9.3.3 盒式布局管理器	215
9.3.4 网格布局管理器	216
9.3.5 卡片布局管理器	217
9.3.6 网格包布局管理器	219
9.4 事件处理	220
9.4.1 事件处理机制	220
9.4.2 事件处理的编程方法	221
9.4.3 事件类型和监听器接口	221
9.5 事件处理举例	226
9.5.1 常见低级事件处理举例	227
9.5.2 常见高级事件处理举例	228
9.5.3 典型范例	229
习题	233
第 10 章 小应用程序	234
10.1 小应用程序概述	234

10.1.1	小应用程序的概念	234
10.1.2	JApplet 类	235
10.2	小应用程序的生命周期	235
10.2.1	小应用程序生命周期的概念和方法	235
10.2.2	小应用程序的编写步骤	236
10.3	小应用程序的 AWT 绘制	238
10.3.1	小应用程序的绘制原理	238
10.3.2	图形绘制——Graphics 类	238
10.3.3	字体使用——Font 类	241
10.3.4	颜色选择——Color 类	243
10.4	小应用程序与工作环境的通信	244
10.4.1	小应用程序与浏览器之间的通信	245
10.4.2	同一个网页小应用程序之间的通信	248
10.4.3	小应用程序的网络通信	249
10.5	小应用程序的多媒体应用	250
10.5.1	图像显示	250
10.5.2	声音播放	252
	习题	253
第 11 章	多线程	254
11.1	线程的概念	254
11.2	线程的创建	255
11.2.1	通过扩展 Thread 类创建线程	255
11.2.2	通过实现 Runnable 接口创建线程	258
11.3	线程的控制	259
11.3.1	线程的生命周期	259
11.3.2	线程的优先级	261
11.3.3	线程的调度	263
11.4	互斥与同步	264
11.4.1	临界区与互斥	264
11.4.2	线程同步	267
	习题	270
第 12 章	输入输出流	271
12.1	文件类	271
12.1.1	文件和目录的创建、查看和删除	273
12.1.2	文件和目录的排序	274
12.1.3	文件和目录的过滤	276
12.2	输入输出流概述	277
12.2.1	流的概念	277

12.2.2	字节流、字符流和对象流	277
12.2.3	结点流和过滤流	280
12.2.4	流的编程方法	281
12.2.5	标准输入输出流	281
12.3	结点流	283
12.3.1	二进制文件流	283
12.3.2	文件字符流	286
12.3.3	字节数组流	287
12.3.4	字符数组流	289
12.3.5	字符串流	290
12.4	过滤流	290
12.4.1	InputStreamReader 类和 OutputStreamWriter 类	290
12.4.2	BufferedInputStream 类和 BufferedOutputStream 类	292
12.4.3	BufferedReader 类和 BufferedWriter 类	294
12.4.4	DataInputStream 类和 DataOutputStream 类	296
12.4.5	PrintStream 类和 PrintWriter 类	298
12.4.6	对象流	300
12.4.7	管道流	302
	习题	305
第 13 章	Java 网络编程	306
13.1	计算机网络的概念	306
13.1.1	计算机网络	306
13.1.2	网络编程的概念	306
13.1.3	Java 与网络编程	307
13.2	InetAddress 类	308
13.3	TCP Socket 通信	311
13.3.1	套接字	311
13.3.2	TCP ServerSocket 和 Socket	312
13.4	UDP 套接字	316
13.4.1	UDP 协议	317
13.4.2	UDP 数据	317
	习题	321
	参考文献	322

第1章 Java 概述

随着计算机和 Internet 的发展,迫切需要面向网络的计算机语言。大家想一想,在 PC 上用 Windows 编写的程序,不做修改就直接拿到 UNIX 系统上运行,行吗?显然不行,因为程序必须转换成计算机硬件的机器指令才能执行,而专门为某种计算机硬件和操作系统编写的程序是不能够直接放到另外的计算机硬件上执行的,至少要做移植工作。而在网络环境下,网络上的计算机是五花八门的,即计算机的类型和操作系统是不一样的。例如,SUN 工作站的硬件是 SPARC 体系,软件是 UNIX 中的 Solaris 操作系统;而 PC 的硬件是 INTEL 体系,操作系统是 Windows 或 Linux。因此,要想让程序在网络中各种计算机上都能正常运行,这就要求程序设计语言能够跨越各种软件和硬件平台。Java 就是在这种需求背景下诞生和流行的。微软的比尔·盖茨(Bill Gates)认为:“Java 是长期以来最卓越的程序设计语言。”如今,它已经成为最流行的网络编程语言。

1.1 Java 语言简介

1.1.1 Java 语言的起源与发展

1991 年,美国 Sun Microsystems 公司(以下简称 Sun 公司)的 Jame Gosling、Bill Joy 等为了能够在消费电子产品上开发应用程序,积极寻找合适的编程语言。由于消费电子产品种类繁多,所采用的处理芯片不相同,存在着跨平台的问题。当时,Sun 公司的研究人员采用 C++ 语言来编写消费电子产品的应用程序,但是研究表明,对于消费电子产品而言,C++ 语言过于复杂和庞大,并不适用,安全性也并不令人满意。于是,Bill Joy 先生领导的研究小组就着手设计和开发出一种新语言,称之为 Oak。Oak 是一种与平台无关、可靠性强、小而灵活的编程语言,但在当时,该语言并没有引起人们的注意。

1994 年下半年,随着 Internet 的迅猛发展,人们发现 Oak 这种中性平台及安全性高的面向对象语言恰恰就是全球信息网所期待的语言。为了使之适应网络需求,开发人员对 Oak 进行了一系列的改进,1995 年,面向对象的 Java 语言诞生了。其响亮的口号是“Write Once, Run Anywhere”。Java 的诞生对传统的计算模型提出了新的挑战。如软件 4A 目标要求软件能达到任何人在任何地方和任何时间对任何电子设备都能操作。表 1-1 对 Java 的发展历史作了总结。

表 1-1 Java 的发展历史

时间	Java 的发展历史
1991 年	Sun 以 C++ 为基础开发新的程序设计语言,并将其命名为 Oak
1995 年	Java 语言诞生
1996 年	JDK1.0 发布,Java 的第一个开发包 JDK(Java Development Kit)发布
1998 年	JDK1.2(称 Java2)发布,标志着 Java 已经进入 Java2 时代
1999 年	Java 分成 J2SE、J2EE 和 J2ME,JSP/Servlet 技术诞生
2000 年	JDK1.3 发布,JDK1.4 发布
2002 年	J2SE1.4 发布
2004 年	J2SE1.5 发布,这是一个里程碑事件,从此,J2SE1.5 更名为 J2SE5.0
2005 年	J2SE、J2EE 和 J2ME 更名为 Java SE、Java EE 和 Java ME,公开了 Java SE6
2006 年	SUN 公司发布 JRE6.0
2009 年	Oracle(甲骨文)收购 Sun、Java 并归甲骨文公司

1.1.2 Java 语言的特点

Sun 公司在“Java 白皮书”中对 Java 的定义是:“Java: A simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multi-threaded, and dynamic language.”按照这个定义,Java 语言是一种简单的、面向对象的、分布的、解释的、健壮的、安全的、体系结构中立的、可移植的、高性能的、多线程的以及动态执行的程序设计语言。

1. 简单易学

Java 语言简单易学。其语法与 C 语言和 C++ 语言很接近,但它丢弃了 C++ 中复杂、不安全的特性,如指针与内存管理、操作符重载和多继承等。

2. 面向对象

面向对象的技术是当今软件开发的主流技术之一。Java 语言是一个彻底的纯面向对象的程序设计语言。它具有面向对象的封装、继承和多态三大特点。Java 语言通过类实现封装,在类之间实现单继承,在接口之间实现多继承,并全面支持动态绑定来实现多态。

3. 安全性

除了 Java 语言具有的许多安全特性以外,Java 提供了字节码校验器、文件访问限制机制、类装载器和运行时内存布局四级安全保证机制。

4. 跨平台(体系结构中立)

Java 程序能够在网络上任何地方执行;完全统一的语言版本,实现平台无关性;字节代码与平台无关性;访问底层操作系统功能的扩展类库,不依赖于具体系统等。

5. 多线程

Java 环境本身就是多线程的。特别地,Java 提供了对多线程的语言级支持,程序员能很方便地编写多线程应用程序。

6. 动态性

Java 所需要的类是运行时动态装载的,也可从网络载入。在分布环境中动态地维护应用

程序和类库的一致性,类库的更新,不需重新编译程序,不影响用户程序的执行。

1.2 Java 程序工作原理

1.2.1 Java 程序工作流程

学习 Java 语言,有必要了解 Java 的工作流程,这将更有助于理解 Java 语言的特点。

对于运行在 Internet 上的网络应用程序,需要有良好的可移植性。因为 Internet 是由各种各样不同类型的终端、服务器和 PC 等硬件设备组成的,而且在这些设备上运行的软件系统也是多种多样的,所以 Internet 上的网络应用程序应该具有在各种不同的软硬件平台上都能正常工作的能力。Java 的工作流程使得它具有了这样的能力,如图 1-1 所示。

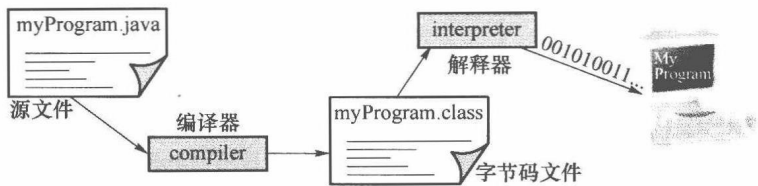


图 1-1 Java 的工作流程

Java 的工作流程是:

- (1) 编辑 Java 源文件。首先,编程人员使用 Java 语言编写好源代码,形成源文件。文件名的后缀为 .java。
- (2) 编译 Java 源文件。然后,使用 Java 编译器来编译 Java 源文件,生成一种二进制的中间码,称为字节码(byte code),形成字节码文件。文件名的后缀为 .class。
- (3) 运行 Java 程序。最后,使用 Java 解释器来解释执行编译生成的字节码,完成 Java 程序要实现的功能。

1.2.2 Java 虚拟机

Java 虚拟机(Java Virtual Machine,JVM)是一个抽象的计算机处理器,负责运行 Java 字节码文件,负责把字节码解释成具体平台上的机器指令进行执行。它也称为 Java 解释器。

Sun 公司对 Java 虚拟机规范进行了严格定义和控制。Java 虚拟机规范规定了:

- (1) Java 虚拟机由 6 个部分组成:一组指令集、一组寄存器、一个类文件格式规定、一个栈、一个无用单元收集堆(Garbage-collected-heap)和一个方法区域。
- (2) 虚拟机能够认识的字节代码以及能实现的功能。

但是 Java 虚拟机规范没有规定 Java 虚拟机组织结构及其功能是如何实现的。而这些必须在真实机器上以某种方式实现,既可以用软件实现,也可以用硬件实现。因此,对于不同的软硬件平台(指处理器和操作系统),Java 虚拟机就要作专门的实现。但是,在不同软硬件平台上的 Java 虚拟机,只要符合 Java 虚拟机规范,将保证字节码文件的正确执行。Java 语言编译执行的过程,如图 1-2 所示。

Java 平台由 Java 虚拟机和 Java 应用编程接口(Application Programming Interface,API)构成。Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口,可分为核心部

分和扩展部分。核心 Java API 中封装了程序设计所需要的主要应用类。在硬件和操作系统平台上安装一个 Java 平台之后,Java 应用程序就可运行。现在 Java 平台几乎已经嵌入到所有的操作系统。这样,Java 程序只编译一次,就可以在各种系统中运行。

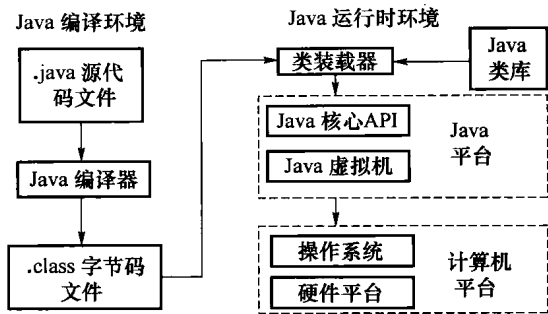


图 1-2 Java 语言编译执行的过程

1.2.3 Java 语言的平台无关性

Java 程序的跨平台主要是指字节码文件可以在任何具有 Java 虚拟机的计算机或者电子设备上正确运行。其原因主要是：

- (1) Java 语言是与平台无关的。它没有“依据机器的不同而不同”或“由编译器决定”等字眼。因此,Java 语言编写的程序在不同平台上都是完全一样的。
- (2) 字节代码是与平台无关的。由 Java 编译器产生的字节代码是二进制码,它与具体的计算机处理器代码无关。
- (3) Java 虚拟机隐藏了不同平台的差异。字节码文件并不是直接运行在计算机平台上,而是运行在 Java 虚拟机上。Java 虚拟机将字节码文件与操作系统及硬件分开,保证了字节码文件在不同平台上运行的正确性。

下面通过一个类比的例子说明 C 语言和 Java 语言跨平台的区别。假如一个外国人想和一个广东人、上海人和陕西人聊天。但这个外国人只懂英语,广东人只懂粤语,上海人只懂上海话,陕西人只懂陕西话。这样,这位老外必须将他的话分别翻译成粤语、上海话和陕西话,这三位中国人才能听懂。在这里做如下类比,英语相当于 C 语言,地方话相当于不同计算机平台的机器指令。在不同计算机平台上,C 语言源程序必须经编译链接,形成该平台所识别的机器码文件后才能运行,如图 1-3 所示。

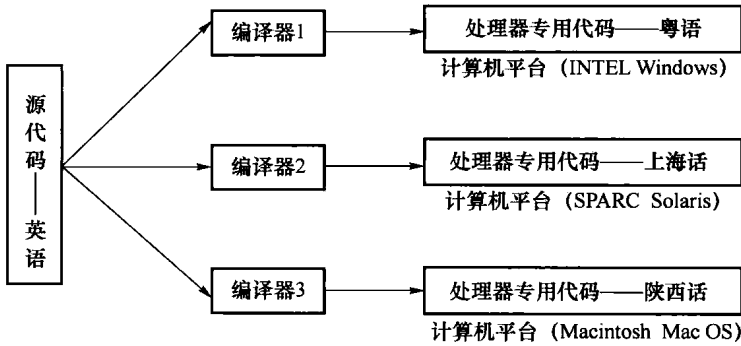


图 1-3 C 语言的机器代码在不同计算机平台上的相关性