

高等院校计算机专业应用技术系列教材

Java
CHENGXU SHEJI
YUYAN

Java 程序设计语言

陈杰华◎编著



北京大学出版社
PEKING UNIVERSITY PRESS

高等院校计算机专业应用技术系列教材

Java 程序设计语言

陈杰华 编著



北京大学出版社
PEKING UNIVERSITY PRESS

图书在版编目 (CIP) 数据

Java 程序设计语言 / 陈杰华编著. — 北京: 北京大学出版社, 2017. 4
(高等院校计算机专业应用技术系列教材)
ISBN 978-7-301-16156-2

I. ① J… II. ① 陈… III. ① Java 语言 - 程序设计 - 高等学校 - 教材 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2009) 第 222776 号

书 名 Java 程序设计语言
著作责任者 陈杰华 编著
责任编辑 王 华
标准书号 ISBN 978-7-301-16156-2
出版发行 北京大学出版社
地 址 北京市海淀区成府路 205 号 100871
网 址 <http://www.pup.cn> 新浪微博: @北京大学出版社
电子信箱 zpup@pup.cn
电 话 邮购部 62752015 发行部 62750672 编辑部 62765014
印 刷 者 北京大学印刷厂
经 销 者 新华书店
787 毫米 × 1092 毫米 16 开本 19 印张 462 千字
2017 年 4 月第 1 版 2017 年 4 月第 1 次印刷
定 价 38.00 元

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究

举报电话: 010-62752024 电子信箱: fd@puppkuedu.cn

图书如有印装质量问题, 请与出版部联系, 电话: 010-62756370

内 容 提 要

本书是学习 Java 程序设计语言的基础教材,共分为 12 章,主要内容包括:绪论,Java 语言目录结构、程序规范和上机操作,数据类型、运算符和表达式、Java 语言的基本语句和控制结构,类、数组和字符串,异常处理,线程与对象串行化,文件和输入输出,图形用户界面程序和事件处理,编写 Applet 程序,图形程序设计,Java 的多媒体应用。为方便讲课与上机实践,每章最后均附有习题,并在书后附有习题答案。

本书内容丰富,讲解简明易懂、循序渐进、深入浅出。本书可作为高等院校本、专科各专业学生学习 Java 语言程序设计课程的教材,也可作为初学者、IT 行业爱好者的辅助学习教材。

前 言

Java 程序设计语言广泛使用面向对象程序设计,它模拟人类对现实世界的思维方式,将数据与操作组合起来,符合现代软件开发的要求,正逐步替代面向过程的程序设计技术,进而成为计算机应用开发领域的主流趋势。同时 Java 程序设计语言具有跨平台运行的优点,从而使软件重用程度非常高。此外,Java 白皮书中还提到:Java 程序设计语言具有简单、分布式、解释、安全、可移植、高性能、多线程、动态等优点。

本书主要特点体现在三个方面:

(1) 适合教师教学。本书按教材编写,内容组织和结构合理,条理清晰。同时,每章后均安排有习题,方便学生学习使用。同时,教师可以利用本书的电子教案、习题参考答案等教学资源,备课、讲课、指导学生上机,方便教学。

(2) 章节结构合理。本书每章均按照基本概念、语句结构、程序思想、代码实现等程序设计思想流程介绍 Java 程序设计语言,有利于学生对照学习,提高学习效率。本书采用循序渐进的学习模式,适合初、中级读者掌握 Java 程序设计语言。

(3) 图文并茂,简明易懂。本书文字通俗,努力做到用通俗语言来解释概念和程序设计思想。对绝大多数 Java 程序都附有计算机运行后的窗口图形,以方便读者阅读。重点介绍 Java 程序设计语言的相关知识,为读者体验式学习奠定基础。

本书最大的特色是按课程教学方式来组织内容,因此适合老师授课,也适合学生阅读。

本书由陈杰华制定全书整体框架和统稿工作,并编写主要文字内容,其他参与文字编写、资料整理、代码调试、图片制作的还有孟宏源、戴丽娟老师等。由于作者水平有限,加之编写出版时间仓促,书中难免不足和谬误之处,恳请广大读者批评和指正。

编者电子邮件地址:cjh028@126.com 和 chenjiehua@scu.edu.cn,如有技术问题、索要案例、习题源程序文件和电子教案,可以发送电子邮件,我们一定准时回复并尽可能为您提供方便。

编者

2016 年 10 月 20 日

目 录

第 1 章 绪论	(1)
1.1 Java 语言起源、特点和现状	(1)
1.1.1 Java 语言的起源	(1)
1.1.2 Java 语言的特点	(1)
1.1.3 Java 语言的现状	(4)
1.2 Java 语言的平台和应用	(5)
1.2.1 Java 语言的平台	(5)
1.2.2 Java 语言的应用	(6)
1.3 Java 语言实现机制	(6)
1.3.1 Java 虚拟机	(6)
1.3.2 代码验证机制	(7)
1.4 简单的 Java 程序介绍	(7)
1.4.1 Java Application 程序	(8)
1.4.2 Java Applet 程序	(9)
1.5 本章知识点	(10)
习题一	(10)
第 2 章 Java 语言目录结构、程序规范和上机操作	(12)
2.1 JDK 目录结构	(12)
2.1.1 JDK 目录结构概述	(12)
2.1.2 JDK 工具集	(13)
2.2 Java 语言的 API 结构	(13)
2.2.1 Java 语言的 API 结构概述	(13)
2.2.2 Java 语言的类库	(13)
2.2.3 Java 语言程序结构	(15)
2.3 Java 语言程序规范	(15)
2.3.1 Java 语言命名约定规则	(15)
2.3.2 Java 语言程序注释规则	(15)
2.3.3 Java 语言源文件结构规则	(16)
2.4 JDK 的下载与安装	(17)
2.4.1 JDK 的下载过程	(17)
2.4.2 安装 JDK 过程	(19)
2.4.3 设置环境参数	(21)
2.5 JDK 的操作命令与程序调试过程	(21)
2.5.1 JDK 的操作命令	(21)
2.5.2 Java 程序调试过程	(23)
2.5.3 Java Application 和 Java Applet	(27)

2.6 本章知识点	(27)
习题二	(28)
第 3 章 数据类型、运算符和表达式	(30)
3.1 常量和变量	(30)
3.1.1 常量	(30)
3.1.2 标识符	(31)
3.1.3 变量	(32)
3.2 基本数据类型及转换	(33)
3.2.1 基本数据类型	(33)
3.2.2 自动类型转换和强制类型转换	(38)
3.2.3 Java 类库中对简单数据类型的类包装	(40)
3.3 运算符和表达式运算	(41)
3.3.1 运算符	(41)
3.3.2 表达式及其运算	(45)
3.3.3 扩展赋值运算符	(46)
3.4 数组和字符串	(46)
3.4.1 数组	(46)
3.4.2 字符串	(50)
3.5 本章知识点	(50)
习题三	(50)
第 4 章 Java 语言的基本语句和控制结构	(55)
4.1 条件语句	(55)
4.1.1 if 语句	(55)
4.1.2 switch 语句	(59)
4.1.3 多分支程序	(60)
4.2 循环语句	(61)
4.2.1 for 循环	(61)
4.2.2 while 循环	(63)
4.2.3 do...while 循环	(63)
4.2.4 多重循环	(65)
4.2.5 循环程序的组成	(67)
4.2.6 跳转控制语句	(67)
4.2.7 循环语句与分支语句的嵌套	(70)
4.3 方法定义与调用	(72)
4.3.1 返回语句	(72)
4.3.2 调用方法	(72)
4.3.3 方法调用中的参数传递	(74)
4.3.4 方法的嵌套调用	(74)
4.4 递归	(75)
4.4.1 递归概念	(75)
4.4.2 递归程序示例	(76)

4.5 本章知识点	(77)
习题四	(78)
第5章 类、数组和字符串	(81)
5.1 面向对象编程的基本概念和特征	(81)
5.1.1 面向对象编程的基本概念	(81)
5.1.2 数据抽象、封装、继承和多态	(82)
5.2 类的基本组成和使用	(84)
5.2.1 类的组成	(84)
5.2.2 类的使用	(95)
5.2.3 继承与关键字 super	(102)
5.3 对象的生成、使用和删除	(105)
5.3.1 对象的生成阶段	(105)
5.3.2 对象的使用阶段	(105)
5.3.3 对象的清除阶段	(110)
5.4 接口与包	(110)
5.4.1 Java 中的接口	(110)
5.4.2 Java 中的包	(113)
5.5 内部类和 Java 类库的常用类	(115)
5.5.1 内部类	(115)
5.5.2 类库	(116)
5.5.3 java.lang 包	(116)
5.5.4 实用工具 Java.util 包	(118)
5.6 数组	(120)
5.6.1 一维数组	(120)
5.6.2 二维数组	(122)
5.7 字符串	(124)
5.7.1 处理字符串的常用方法	(124)
5.7.2 字符串处理程序示例	(125)
5.8 本章知识点	(126)
习题五	(127)
第6章 异常处理	(130)
6.1 异常概念	(130)
6.1.1 一般程序设计语言的错误处理	(130)
6.1.2 异常事件	(132)
6.2 Java 语言的异常处理	(132)
6.2.1 异常处理	(132)
6.2.2 声明异常	(133)
6.2.3 抛出异常	(134)
6.3 自定义异常类	(135)
6.3.1 自定义异常类概念	(135)
6.3.2 自定义异常类	(136)

6.3.3 异常处理的语句模式	(137)
6.4 捕捉异常与 try-catch 语句	(138)
6.4.1 捕捉单个异常	(138)
6.4.2 捕捉多个异常	(139)
6.5 finally 语句	(142)
6.5.1 引言	(142)
6.5.2 finally 语句	(142)
6.6 异常类和错误类	(144)
6.6.1 Exception(异常)类	(144)
6.6.2 Error 类	(145)
6.7 本章知识点	(146)
习题六	(146)
第 7 章 线程与对象串行化	(149)
7.1 线程概念和使用	(149)
7.1.1 线程概念	(149)
7.1.2 线程模型	(150)
7.1.3 创建线程	(150)
7.1.4 线程优先级与线程调度策略	(153)
7.1.5 Thread 类中的线程控制方法	(155)
7.2 多线程操作	(159)
7.2.1 多线程并发操作中存在的问题	(159)
7.2.2 对象的加锁及其操作	(159)
7.2.3 防治死锁	(159)
7.2.4 线程交互与同步	(159)
7.2.5 线程状态与生命周期	(160)
7.2.6 支持线程的类	(160)
7.2.7 线程组	(161)
7.3 对象串行化	(163)
7.3.1 对象串行化	(163)
7.3.2 将对象写到对象输出流中	(163)
7.3.3 从对象输入流中读取对象	(165)
7.4 本章知识点	(167)
习题七	(167)
第 8 章 文件和输入输出	(171)
8.1 文件和输入输出概述	(171)
8.1.1 概述	(171)
8.1.2 Java 中有关输入输出流的类	(171)
8.2 文件	(173)
8.2.1 创建文件	(173)
8.2.2 删除文件	(175)
8.2.3 File 类提供的方法	(176)

8.3	字节输入输出流	(179)
8.3.1	字节输入流	(179)
8.3.2	字节输出流	(180)
8.3.3	读写内存数据	(180)
8.4	字符类输入输出流	(181)
8.4.1	字符类输入流	(181)
8.4.2	字符类输出流	(183)
8.5	本章知识点	(185)
	习题八	(185)
第 9 章	图形用户界面程序和事件处理	(188)
9.1	GUI 与 Swing 特性	(188)
9.1.1	概述	(188)
9.1.2	java.awt 包	(188)
9.1.3	Swing 的特性	(188)
9.2	AWT 库简介	(189)
9.2.1	AWT 库	(189)
9.2.2	常用容器	(190)
9.2.3	布局管理器(LayoutManager)	(195)
9.3	AWT 构件库	(199)
9.3.1	基本构件	(199)
9.3.2	构件与监听器	(202)
9.4	Swing 简介	(203)
9.4.1	用 Swing 编写程序	(203)
9.4.2	Swing 构件分类	(205)
9.4.3	各种容器面板和构件	(207)
9.5	事件处理机制	(215)
9.5.1	AWT 事件处理模型	(215)
9.5.2	AWT 事件类	(216)
9.5.3	事件监听器	(217)
9.5.4	事件适配器	(217)
9.5.5	Swing 的事件处理机制	(219)
9.6	本章知识点	(221)
	习题九	(221)
第 10 章	编写 Applet 程序	(224)
10.1	Applet 程序概念	(224)
10.1.1	Applet 概念	(224)
10.1.2	Applet 生命周期	(225)
10.1.3	Applet 的类层次结构	(226)
10.1.4	Applet 类与 API 概述	(227)
10.1.5	Applet 显示方法	(228)
10.1.6	Applet 类的显示	(228)

10.2	Applet 执行过程	(231)
10.2.1	编写 Applet	(231)
10.2.2	在 HTML 页面中包含 Applet	(231)
10.2.3	Applet 程序的执行过程	(232)
10.3	Applet 类的图形绘制	(232)
10.3.1	Graphics 类中的输出字符串方法	(233)
10.3.2	Graphics 类中的绘图方法	(233)
10.4	本章知识点	(236)
	习题十	(236)
第 11 章	图形程序设计	(239)
11.1	与图形有关的概念	(239)
11.1.1	屏幕坐标	(239)
11.1.2	得到图形的高度和宽度	(240)
11.1.3	paint()、repaint()和 update()方法	(240)
11.2	颜色模型	(241)
11.2.1	RGB 颜色模型	(241)
11.2.2	Color 类中的颜色常数	(241)
11.2.3	Color 类中的方法	(242)
11.3	Graphics 类中的绘图方法	(244)
11.3.1	输出字符	(244)
11.3.2	画线 drawLine()	(246)
11.3.3	绘制矩形	(248)
11.3.4	绘制椭圆	(252)
11.3.5	绘制椭圆弧	(252)
11.3.6	绘制多边形	(254)
11.3.7	设置颜色	(255)
11.3.8	获得字符串	(256)
11.3.9	复制图形	(256)
11.4	字体	(257)
11.4.1	文本文字与图形文字	(257)
11.4.2	Font 类	(258)
11.4.3	FontMetrics 类	(260)
11.4.4	设置字体	(261)
11.4.5	输出字符串	(262)
11.4.6	获取系统字体	(262)
11.5	清除图形	(262)
11.5.1	clearRect()方法	(262)
11.5.2	程序示例	(262)
11.6	异或操作	(264)
11.6.1	setXORMode()方法	(264)
11.6.2	程序示例	(264)

11.7 Java2D 图形	(265)
11.7.1 Java2D 概述	(265)
11.7.2 2D 图形	(265)
11.8 本章知识点	(268)
习题十一	(268)
第 12 章 Java 的多媒体应用	(271)
12.1 图像显示	(271)
12.1.1 Image 类	(271)
12.1.2 URL 类	(272)
12.2 图像装载和显示	(274)
12.2.1 图像装载	(274)
12.2.2 图像显示	(275)
12.3 声音播放	(278)
12.3.1 加载声音文件	(278)
12.3.2 声音播放方法	(278)
12.4 动画	(280)
12.4.1 动画概念	(280)
12.4.2 程序示例	(281)
12.5 本章知识点	(282)
习题十二	(282)
习题参考答案	(284)
参考文献	(289)

第 1 章 绪 论

“Java”在英文中的意思是指印度尼西亚的爪哇岛,该岛以盛产咖啡而闻名于世。但是在计算机领域中,Java 则是一种程序设计语言。它具有与计算机操作平台无关的特性,是当今最流行的网络开发语言之一。本章主要内容包括:Java 语言的特点,Java 语言的三种平台,Java 语言的实现机制和主要应用。

1.1 Java 语言起源、特点和现状

1.1.1 Java 语言的起源

1991 年,美国 Sun Microsystems 公司成立了一个新的计算机语言开发小组,取名为 Green,其目的是研制一种面向家用电器市场的计算机语言。该开发小组首先考虑这一软件产品必须具有与计算机操作系统平台无关的特征,同时还必须具有高度简洁和安全可靠的特征,取名为 Oak 语言,这就是 Java 语言的前身。

同时,美国 Sun Microsystems 公司抓住 WWW 网页都是静态表示的致命弱点,将 Oak 语言重新定位在 WWW 浏览器上。在继续完善 Oak 语言的同时,还用 Oak 语言开发出一个真正的因特网上的 WWW 浏览器——WebRunner,这就是后来的“HotJava”程序设计语言。1995 年 5 月,Oak 语言被重新命名为 Java 语言。

1.1.2 Java 语言的特点

在 Java 白皮书中,它是这样描述的:“Java 语言是一个简单的、面向对象的、分布式的、解释的、健壮的、安全的、独立于平台的、可移植的、高性能的、多线程的、动态的程序设计语言。”

1. 简单性

Java 属于一种简单的程序设计语言,它表现在如下三个方面:

(1) 易学易用。Java 语言的概念不多,而且这些概念大多数程序员都是熟知的。在外部表现形式上,Java 语言与 C++ 语言极为相似,这在一定程度上也就保证了 Java 语言易学易用这一特点。

(2) 取消 C++ 语言中的不良成分。为保证简单性,Java 语言取消了 C++ 语言中的许多不良成分,这些不良成分往往容易导致不良的程序设计风格。

(3) 简单运行系统。Java 语言的运行系统是十分简单的,它的基本解释器(Interpreted)只有 49KB,即使加上全部标准库和线程支持也不过数百 KB。

由此可知,Java 语言是从 C++ 语言发展而来的,原来 C++ 语言的程序开发者很容易转向 Java 语言环境。另外,Java 语言比 C++ 语言更简单易学,所编程序的可读性也得到了大大地增强。

2. 面向对象程序设计

Java 语言完全具有面向对象程序设计技术的四大特点:继承性、封装性、多态性和动态性。Java 语言的封装性比 C++ 语言的封装性要好得多,Java 语言没有全局变量。在 Java 语言中,绝大部分成员变量都是对象,只有属于基本数据类型的整型变量、浮点型变量、字符型变量、布尔型变量才是例外。另外,Java 语言为程序员提供了大量的类,这些类又可以组成各种各样的软件包(Package),从而使程序员从复杂的程序设计中解放出来。Java 语言中的类和 C++ 语言的类一样也是有层次继承关系的,子类可以继承父类中的全部属性和方法,程序员只需要为特定计算任务编写部分代码。

Java 语言是一种纯面向对象 POO(Pure Object Oriented)的程序设计语言,其程序代码以类的形式组成,完全抛弃了 C++ 语言中的非面向对象特性,主要结构特点如下:

(1) 程序一般是由类的声明和类的使用两部分组成的。在主程序中,程序员可以定义全部对象,并规定对象间传递消息的方式。

(2) 程序中的全部操作都是通过向对象发送消息(Message)来实现的,对象接收到消息后,就可以启动有关的方法(Method)完成相应的操作。

注意:C++ 语言中将全部类均处理成虚函数,而 Java 语言中将类处理成是可以动态加载的程序代码。

3. 分布式计算

Java 语言是随着因特网发展起来的一种面向对象语言,它支持网络上的许多应用程序。从本质上看,Java 语言是一种分布式(Distributed)的程序设计语言。Java 语言系统提供了一个 Java.net 的包,通过该包中的许多类,程序员可以完成各种层次上的网络连接。例如:Java 语言系统提供了一个 Socket 类,通过该类可以获取安全的网络连接,这样就方便程序员开发分布式的客户机/服务器(Client/Server)应用程序;另外,Java 语言系统提供了一个 URL 类,它支持 Java 应用程序通过因特网打开并访问远程资源,这样就使得用 Java 语言打开远程文件与打开本地文件是一致的。

4. 健壮性

Java 语言的健壮性(也称为鲁棒性)体现在三个方面:强类型语言、异常处理和内存储器管理。

(1) 强类型语言。Java 语言要求进行完全显式的方法声明、变量声明和类型转换,这就保证了编译器可以发现无效变量、方法调用错误、数据类型不一致等问题,从而保证应用程序更加可靠地运行。

(2) 异常处理。程序员通过使用 try/catch/finally 语句,可以将一组错误处理代码放在一个特定的地方,这样便于简化错误处理工作,同时增强程序可读性。

(3) 内存储器管理,又称为自动垃圾回收机制。Java 语言的无用单元自动收集机制可以防止动态内存储器分配所导致的种种问题,Java 语言不支持 C++ 语言程序开发者推崇备至的指针操作,这就从根本上杜绝了应用程序对内存储器的非法访问。另外,Java 解释器在运行时也进行在线的实时检查,可以发现数组下标越界、字符串访问错误、存储空间不够等问题。

5. 安全性

Java 语言的运行系统具有字节代码(JavaByteCode,JBC)验证机制,从而保证从网络上下载的任何代码不违反 Java 语言的限制;另外,Java 语言的内存储器分配模式是防止有害代码

或病毒破坏用户文件系统的—一个重要手段。由于 Java 语言取消了 C++ 语言中的指针操作，所以—个 Java 应用程序是无法访问它不该访问的内存储器单元的，Java 程序的安全性具体体现在三个层次上，即编译层、解释层和平台层，如表 1-1 所示。

表 1-1 Java 程序安全性的具体体现

层次	说明
编译层	完成词法分析、语法检查等
解释层	包括字节码校验器、测试代码段格式与规则检查、访问权限与类型转换合法性检查、操作数堆栈的上溢或下溢检查、代码参数类型合法性检查等
平台层	通过配置策略可设定访问资源范围而无需区分本地机与远程系统

6. 平台无关性

Java 语言的本质特征就是与平台无关性。任何其他的程序设计语言编译或解释产生的代码，都不能保证在任何操作系统平台上能够有效地运行。Java 语言编译产生的中间代码是字节代码，而不是任何形式的机器语言代码。JBC 本身是一种解释型 (Interpreted) 的代码，Java 程序的字节代码必须运行在一个解释器上。Java 语言的字节代码是一种与操作系统平台无关的面向对象的文件格式，所以 Java 语言可以高效地运行在不同的操作系统平台上。

Java 语言使用完全统一的程序描述机制，它的基本数据类型不会随机器的变化而变化。例如—个整型数据的长度总是 32 位的，不像 C++ 语言，随着机器的不同，其整型数的长度是不相同的，可能是 16 位，也可能是 32 位，甚至是 64 位的。此外，Java 语言环境还定义了一个用于访问底层操作系统功能的扩展类库，从而使 Java 语言应用程序能够完全不依赖于具体的操作系统环境。

7. 可移植性

实际上，Java 语言与平台无关性本身就提供了一种最佳的程序可移植性。Java 语言运行系统本身是采用标准 ANSI C 语言编写的，Java 语言的编译器是采用 Java 语言本身编写的，从而保证 Java 语言程序的可移植性。Java 语言定义出—套自己独有的虚拟机，以及在该虚拟机上使用的机器代码——JBC。Java 通过预先将源代码编译为接近于机器指令的字节代码，有效地克服传统解释型语言的性能缺陷。另外，由于解释执行 JBC 只需要 Java 语言运行系统，而不需要某个特定操作系统平台的支持，所以这就保证了 Java 语言运行环境与操作系统平台的无关性，程序可移植性好。

8. 解释型

Java 语言编译器产生的中间代码是一种字节代码，而不是任何形式的机器语言代码。JBC 本身是一种解释型的代码，Java 语言程序的字节代码必须运行在一个解释器上。所以，Java 语言是一种解释型的程序设计语言。

但是，Java 语言的 JBC 是一种与操作系统平台无关的对象文件格式，所以 Java 语言程序可以高效地运行在各种不同的操作系统平台上。换句话说，只要该操作系统平台上装有 Java 语言的虚拟机 VM (Virtual Machine)，—个 Java 程序就可以运行在任何—种操作系统平台上。

9. 高性能

Java 作为一个现代语言,它的高性能表现在如下五个方面:

(1) Java 语言是一个面向对象的程序设计语言,使用面向对象设计技术本身就可以提高软件系统的可重用性和可维护性。

(2) Java 语言支持多线程机制,提高了软件系统的交互性、并行性和实时性。

(3) 可以进行动态超文本(Dynamic HTML,DHTML)的 Web 页面开发。

(4) 提供数据窗体向导工具(Data Form Wizard,DFW)访问数据库。

(5) 虽然 Java 语言的字节代码执行速度慢,但通过 Java 语言编译器翻译出来的机器代码,执行速度与 C++ 语言编译器翻译产生的机器语言代码几乎一样地快。

10. 多线程

Java 语言是一种多线程的程序设计语言,它可以同时运行多个线程来处理多个任务。Java 语言系统提供内置的多线程机制,这样简化了多线程应用程序的开发过程,Java 程序员可以使用 Java.lang 包中提供的 Thread 类方便地完成多线程应用程序的开发。

注意:通常多个线程之间的切换速度非常快,程序员从宏观上觉得好像全部线程是同时执行的。

11. 动态性

Java 是“动态”程序设计语言,为了实现 Java 语言的动态性,Java 语言的所有类都被设计成具有运行标识码的、同一类中的各个对象实例都具有相应的运行类定义。Java 语言的动态性主要表现在如下两个方面:

(1) 在 Java 语言程序运行期间,可以很容易地确认要使用哪个类,并找到相应的类库。

(2) Java 语言提供使用类名中所含字符串查找类定义的方法,可以根据字符串将数据类型查找出来,并将其动态地链接到运行系统中。

12. Applet 程序

Applet 属于 Java 语言系统中的一种应用程序,主要是嵌入到 HTML 类型文件中,随主页发布到互联网中,最终由浏览器解释执行。Java 语言属于网络编程语言,它充分利用当前软件新技术,从而避免了许多其他编程语言的缺点。Java 语言围绕网络应用开发,最大限度地利用网络资源,其小应用程序(即 Applet)在网络传输时不受 CPU 和环境限制。

注意:Java 语言的执行模式是属于半编译和半解释的,与脚本描述语言 JavaScript 不同,后者的执行模式是使用浏览器解释执行的。

1.1.3 Java 语言的现状

Java 语言从早期的艰难起步到现在的如日中天,历时也不过短短二十几年的时间。不过,应该看到:Java 语言还是一个十分年轻的程序设计语言,它还存在着许多缺点和不足。其中,最主要的是由于 Java 语言是解释运行的,因而用 Java 语言开发的应用程序运行速度比用 C++ 开发的应用程序要慢。另一方面,除了简洁性、安全性、多线程和与操作系统平台无关等优点外,Java 语言与 C++ 语言相比也并没有其他优势。

实际上,Java 语言低下的运行速度可以从 Java 语言与计算机平台无关中得到弥补。现在经常提到的“编写一次,运行多次(write once,run anywhere)”就是 Java 语言与计算机平台无关优点的具体体现。另外,Java 语言提供的高度安全性也是至关重要的。现在,Java 语言已

经成为世界上最流行的一种程序设计语言。

计算机自从 1946 年诞生以来,经历了如下四个发展阶段:终端—主机计算机阶段、微型计算机阶段、客户机—服务器阶段和 Java 语言阶段。其中,Java 语言阶段这一时期的特点就是以 Java 为代表的网络计算(Network Computing,NC),Java 语言阶段实现真正与平台无关的计算机方案,从而能充分发挥因特网的作用。

过去,许多计算机操作系统是完全不能兼容的,而 Java 语言却成为允许各类计算机系统相互兼容和共享应用环境的接口。这将使各类软件可以真正实现“编写一次,运行多次”。这样,同一个软件可以运行在不同的计算机操作系统上,例如在 Windows 操作系统、苹果操作系统和 UNIX 操作系统,也可以在机顶盒、个人数据助理、移动电话上得以运行。Java 语言使我们实现了“轻击鼠标拥抱全球”的梦想,它将使全人类受益,并极大地改变我们的生活方式。

1.2 Java 语言的平台和应用

1.2.1 Java 语言的平台

Sun Microsystems 公司在 1995 年正式发布 Java 1.0 版后,在全球范围内引发 Java 热潮,Java 版本也不断地从 V1.1 更新到 V1.8,相应内容也进行了改进和扩充。在计算机行业中,经常将 Java 1.2 以后的版本称为 Java2。

Java 既是一种程序设计语言,又是一种开发软件的运行平台。目前 Sun Microsystems 公司针对不同的市场需求,将 Java 平台划分为标准版、微型版和企业版。

1. 标准版(Java 2 Platform Standard Edition,J2SE)

该平台主要为台式机和 workstation 提供一个开发软件的运行平台,包含 Java 核心类和 GUI 类,这也是目前开发运用最广泛的 Java 平台。J2SE 平台主要面向企业级应用,属于综合性的标准开发平台,其主要特点包括:

(1) 客户端和服务端均有编译程序;

(2) 支持公共对象请求代理体系结构(Common Object Request Broker Architecture,CORBA)、Java 命名与目录接口(Java Naming and Directory Interface,JNDI)和轻量目录访问协议(Light Weight Directory Access,LDAP);

(3) 提供互操作与安全认证技术,可移植性好;

(4) 拥有支持成组开发的各种工具。

2. 微型版(Java 2 Platform Micro Edition,J2ME)

该平台是为嵌入式系统和移动设备提供的 Java 系统平台,可广泛用于各种消费电子产品中,如个人数据助理、移动电话、机顶盒、汽车导航系统或其他无线设备等。

3. 企业版(Java 2 Platform Enterprise Edition,J2EE)

该平台主要为企业提供一个应用服务器开发的平台,包含开发基于 Web 技术的应用程序的类和接口,如 Servlet、Java Server Pages、Enterprise JavaBeans 等。J2EE 本身是开放的,任何软件厂商都可以推出符合 J2EE 标准的自主产品,如 IBM 公司的 Websphere;另外像甲骨文、惠普等公司都已经推出了相应的产品。

在 Java 语言中,J2EE 的服务功能包括 Java 命名与目录接口服务、Java 事务 API(Java