

在线服务： 视频库、源代码库、专业论坛、专家实时支持

Java 编程

网络大课堂

闫迎利 王伟平 等编著



204段全程配音语音教学视频
全书实例源代码，使学习、分析、调试程序更方便

在线服务方式

在线服务网站：www.itzcn.com
QQ群在线服务：45368980、33925615、107423140

清华大学出版社



在线服务： 视频库、源代码库、专业论坛、专家实时支持

Java 编程

网络大课堂

闫迎利 王伟平 等编著



204段全程配音语音教学视频
全书实例源代码，使学习、分析、调试程序更方便



在线服务网站：www.itzcn.com
QQ群在线服务：45368980、33925615、107423140

清华大学出版社
北 京

内 容 简 介

本书全面介绍 Java 编程知识。全书共分 4 篇 19 章, 内容包括: Java 基础篇(第 1~5 章), 介绍 Java 的基础知识; Java 面向对象篇(第 6~10 章), 介绍 Java 面向对象编程知识; Java 应用篇(第 11~17 章), 本篇是该书的重点, 介绍 Java 中常用的工具类, Java 中的输入/输出流, 各种集合, 图形编程, Java 中使用 Swing 组件绘制图形以及创建音频文件播放器等知识; Java 综合案例篇(第 18~19 章), 本篇通过两个综合实例介绍 Java 编程在实际中的应用。本书配套网站 www.itzcn.com 提供了配套学习资源和在线互动学习平台, 帮助读者实现交互式学习模式。

本书可以作为 Java 的基础入门学习书籍, 也可以帮助中级读者提高编程技巧, 掌握面向实践的应用技能。

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 编程网络大讲堂 / 闫迎利等编著. —北京: 清华大学出版社, 2011.1

ISBN 978-7-302-23970-3

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

中国版本图书馆 CIP 数据核字(2010)第 205128 号

责任编辑: 夏兆彦

责任校对: 徐俊伟

责任印制: 何 芊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62795954, jsjic@tup.tsinghua.edu.cn

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

印 装 者: 北京鑫海金澳胶印有限公司

经 销: 全国新华书店

开 本: 190×260 印 张: 41.75 字 数: 1042 千字

附光盘 1 张

版 次: 2011 年 1 月第 1 版

印 次: 2011 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 79.00 元

前 言

Java 是当今非常流行的一种面向对象编程语言。该语言由 Sun 公司推出, 具有跨平台、可移植、分布式、简单、可扩展等诸多特性。Java 语言发展到今天, 已不仅仅表示一门程序语言, 更是一种软件开发平台, 目前已经演化出了 J2SE、J2EE 和 J2ME 3 个版本。凭借其易学易用和功能强大的特点, 可以进行桌面应用、Web 应用、分布式系统及嵌入式系统等应用程序开发, 并且在信息技术、科学研究、军事工业、航空航天等各个领域得到了非常广泛的应用。

1. 本书内容

第一篇: Java 基础篇 (第 1~5 章)。本篇首先介绍 Java 的基础知识, 包括 Java 的工作环境、Java 的基本语法、对数组的操作和对字符串的各种操作。

第二篇: Java 面向对象篇 (第 6~10 章)。本篇主要介绍面向对象的知识, 首先介绍对象和类的一些基本知识, 然后介绍面向对象的继承与多态特性, 详细地介绍内部类, 接着介绍异常处理 (如何捕获异常并处理异常、如何抛出异常和自定义异常, 以及异常种类), 最后介绍了多线程运用。

第三篇: Java 应用篇 (第 11~17 章)。本篇是该书的重点之一, 首先介绍 Java 中常用工具类 (Runtime、System 等) 的使用, 另外介绍正则表达式的使用。接着介绍 Java 中的输入输出流, Java 中各种集合 (List、Set 和 Map 集合), Java 中的图形编程 (AWT 组件、GUI 事件和 Swing 常用事件), Java 中使用 Swing 组件绘制图形及创建音频文件播放器。最后介绍 JDBC 数据库编程过程中的一些常见问题。

第四篇: Java 综合案例篇 (第 18、19 章)。本篇通过两个综合实例介绍 Java 编程在实际中的应用, 分别是蜘蛛纸牌和超市管理系统。

2. 本书特色

本书引用大量来自一线论坛的问题来进行讲解, 力求通过读者实际操作时的问题方法使读者更容易地掌握 Java 编程技术。本书难度适中, 内容由浅入深, 实用性强, 覆盖面广, 条理清晰。

- **结构独特** 通过“问题描述→解决方法→知识扩展→触类旁通”形式将每个知识与实际应用中的问题相结合。
- **形式新颖** 用准确的语言总结概念、用直观的图示演示过程、用详细的注释解释代码、用形象的比喻帮助记忆。
- **技术文档** 将一些非常简单的知识点或者理论性的内容安排在这里。通常这些文档没有具体的实际问题, 但又是读者必须要了解的, 像一些概念和术语。
- **内容丰富** 涵盖了实际开发中 Java 所遇到的基础知识、面向对象、多线程、Java 输



入输出流、Swing 程序开发、Java 绘图与音频播放等方面的热点问题。

- **随书光盘** 本书为实例配备了视频教学文件,读者可以通过视频文件更加直观地学习 Java 的使用知识。
- **网站技术支持** 读者在学习或者工作的过程中,如果遇到实际问题,可以直接登录 www.itzcn.com 与我们取得联系,作者会在第一时间内给予帮助。
- **贴心的提示** 为了便于读者阅读,全书还穿插着一些技巧、提示等小贴士,体例约定如下。
 - 提示** 通常是一些贴心的提醒,让读者加深印象,或者提供建议或解决问题的方法。
 - 注意** 提出学习过程中需要特别注意的一些知识点和内容,或者相关信息。
 - 技巧** 通过简短的文字,指出知识点在应用时的一些小窍门。

3. 读者对象

本书具有知识全面、实例精彩、指导性强的特点,力求以全面的知识性及丰富的实例来指导读者透彻地学习 Java 各方面的知识。本书可以作为 Java 的基础入门学习书籍,也可以帮助中级读者提高编程技能。

本书适合以下人员阅读学习。

- 没有编程基础的 Java 入门人员。
- 有一些编程基础,并且希望全面学习 Java 编程的读者。
- 各大中专院校的在校学生和相关授课老师。
- 相关社会培训班的学员。

除了封面署名人员之外,参与本书编写的还有于永军、张秋香、李乃文、张仕禹、夏小军、赵振江、李振山、李文才、吴越胜、李海庆、何永国、李海峰、陶丽、吴俊海、安征、张巍屹、崔群法、王咏梅、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、郭磊、徐铭、李大庆、王蕾、张勇、郝安林、郭新志、牛丽平、唐守国等。在编写过程中难免会有疏漏,欢迎读者与我们联系,帮助我们改正、提高。

编著者

2010 年 5 月

目 录

绪论	1	0.4 Java 的应用领域	5
0.1 Java 的发展史	1	0.5 Java 的版本	5
0.2 Java 的特征	1	0.6 Java API 文档	6
0.3 Java 的工作原理	3	0.7 Java 开发工具	6

第一篇 Java 基础篇

第 1 章 Java 简介	16	2.6 Java 转义符是什么, 起什么用, 怎么用 ...	51
1.1 Java 到底好在哪里, 高分求教	16	2.7 谁给我写一个 Java 中关于 boolean 类型的例子	53
1.2 Java 只安装了 JRE, 没有安装 JDK, 能不能运行代码	17	2.8 Java 里面 int 类型是不是不能自动转换为 char 类型	55
1.3 javac 不是内部命令或外部命令	18	2.9 Java 中 float 数据转换的小问题	56
1.4 net start mysql 不能启动服务	21	2.10 如何声明 Java 变量并给变量赋值	61
1.5 运行时报 NoSuchMethodError	24	2.11 如何理解 Java 中的变量和常量	62
1.6 仅当显式请求注释处理时才接受类名称 HelloWorld	28	2.12 什么是自加、自减运算	63
1.7 运行时报 NoClassDefFoundError	31	2.13 Java 中扩展赋值运算符“*= ”的问题	67
1.8 什么是 Java Applet	34	2.14 Java 软考中 float 的问题	69
1.9 关于 javadoc	38	2.15 Java 求余	71
1.10 如何提高开发效率	39	2.16 Java 中的三元运算符	72
第 2 章 Java 基本语法	43	2.17 Java 中的“ ”是什么意思	75
2.1 String 和 User name 为什么是非法的标识符	43	2.18 Java 中“~”按位取反运算符的一些问题	77
2.2 Java 的整数类型问题	44	2.19 在 Java 中用位运算符“>>”右移时, 左边何时补 0, 何时补 1 啊	79
2.3 Java 的 4 种整数数据类型的取值范围分别是多少	46	2.10 a 初值为 12, a*=2+3 后 a 的值是多少	81
2.4 编写程序显示 Java 浮点型数据的最大值和最小值	47	第 3 章 流程控制语句	84
2.5 Java 中的 char 问题	49	3.1 Java 中的 if 语句	84
		3.2 Java 中无法使用 if...else 语句	87
		3.3 Java 中用 if...else if 语句判断闰年的问题	90



3.4 用 if 语句判断输入的字符是否是字母, 并判断大小写	94	4.6 如何比较两个数组是否相等	132
3.5 switch 的执行过程	96	4.7 如何创建二维数组	134
3.6 Java 中 switch 的 break 问题	101	第 5 章 字符串与日期类	138
3.7 用 for 循环执行顺序	105	5.1 如何创建字符串变量	138
3.8 用 while 循环输出九九乘法表	110	5.2 如何判断字符串相等	141
3.9 do...while 为何出现死循环	114	5.3 字符串的查找问题	144
3.10 Java 中 break 与 continue 的区别	116	5.4 如何分割字符串	146
第 4 章 数组	119	5.5 截取字符串	149
4.1 如何创建一维数组	119	5.6 正则表达式	152
4.2 如何访问一维数组的元素	122	5.7 字符串替换	154
4.3 如何复制数组	124	5.8 修改字符串的内容	157
4.4 如何对数组进行排序	127	5.9 处理日期时间	161
4.5 如何获取数组中某元素的位置	130	5.10 格式化日期/时间	164
 第二篇 Java 面向对象篇 			
第 6 章 类和对象	172	7.6 多态与重载、重写和覆盖的关系	225
6.1 什么是面向对象	172	7.7 什么是对象类型转换	228
6.2 如何声明类	177	7.8 Java 基础父类强制转换成子类 出现异常	231
6.3 如何创建类的成员变量	180	7.9 接口的实际用途是什么	233
6.4 如何创建类的成员方法	183	7.10 一个 Java 类如何实现多个接口	237
6.5 如何创建类的构造方法	189	第 8 章 内部类	240
6.6 如何理解 Java 类中的主方法 main()	194	8.1 创建内部类的实例对象出错	240
6.7 如何创建与使用类的对象	198	8.2 实例内部类中如何访问外部类的成员	242
6.8 如何理解局部变量	203	8.3 实例内部类中为什么不能定义 static 成员	246
6.9 Java 类中的静态成员	206	8.4 实例化静态内部类出错	247
第 7 章 继承与多态	212	8.5 局部内部类为什么不能用 private 修饰	250
7.1 Java 类的继承中, 都继承了哪些东西	212	8.6 局部内部类只能访问所在方法中的 final 参数与变量	252
7.2 Java 类程序想继承多个类该如何实现	215	8.7 实例化的不是接口, 而是匿名类	252
7.3 Test 类中还可以添加哪些 amethod() 方法	218	8.8 为什么要使用内部类	257
7.4 this (参数) 代表调用本类其他的 构造方法	221	8.9 Outer\$Inner.class 这种文件名 怎么理解	261
7.5 this 和 super 有哪些使用规则	223		

第9章 异常处理.....264

- 9.1 Java 中的“例外”是什么意思.....264
- 9.2 try...catch 语句中, catch 捕获到的
Exception 要怎么写.....265
- 9.3 一个 try 中有多个 catch.....268
- 9.4 try...catch...finally 结构中的 finally
有哪些用途.....270
- 9.5 catch(ArithmeticException e)中参数
的含义.....274
- 9.6 Java 异常中 throw 和 throws 的区别.....277
- 9.7 关于自定义异常类的代码如何解释.....280

第10章 多线程.....286

- 10.1 如何实现多线程.....286
- 10.2 如何让一个类既要实现多线程
又要继承其他类.....290

- 10.3 线程的状态.....292
- 10.4 如何设置线程的优先级.....295
- 10.5 为什么要让线程休眠.....298
- 10.6 如何在线程运行过程中等待
其他线程先运行.....303
- 10.7 如何在多个线程间共享数据.....306
- 10.8 什么是线程的安全问题.....310
- 10.9 如何让某个方法一次只让一个
线程访问.....315
- 10.10 构造方法为什么不可以使用
synchronized 修饰.....318
- 10.11 如何让多个方法与代码块一次
只让一个线程访问.....319
- 10.12 多个线程之间怎么会出现死锁.....324
- 10.13 如何在多个线程之间进行通信.....325

第三篇 Java 应用篇

第11章 常用工具类.....330

- 11.1 Java 中创建一个新的类是否默
认继承了 Object 类.....330
- 11.2 关于 Java.lang 包中 System 类的问题.....332
- 11.3 如何用 Java 程序输出五角星.....335
- 11.4 用 math.sqrt()的方法输出 1~100
之间的素数.....338
- 11.5 用 Math.abs()输出等腰三角形.....341
- 11.6 Math.round(11.5)和 Math.round(-11.5)
的值是多少.....342
- 11.7 利用随机数给数组赋值.....344
- 11.8 随机生成一个 4×4 的二维整数数组,
随机数范围为 0~9.....347
- 11.9 Java 中关于 BigDecimal 的一个疑问.....352
- 11.10 为什么要使用包装类.....355
- 11.11 Java 中 int 和 Integer 的区别.....356

第12章 Java 输入输出系统.....360

- 12.1 Java 中的 InputStream 和 OutputStream
指什么.....360
- 12.2 ByteArrayInputStream 读取数据
怎么总是隔一个数.....362
- 12.3 读取文件为什么要使用 byte[] b =
new byte[1024].....366
- 12.4 为何用 PipedInputStream 和
PipedOutputStream.....370
- 12.5 为什么说管道流在单线程中
可能出现死锁.....373
- 12.6 怎么将两个文件中的内容写到
一个文件里面去.....374
- 12.7 ObjectOutputStream 只能序列化
String 对象吗.....379
- 12.8 字符输入输出流.....383
- 12.9 怎么读取字符数组输出流中的内容.....385



12.10	读取文件出现中文乱码怎么解决	387	14.16	事件源、事件和事件处理器之间的工作关系	470
12.11	用 isFile()方法判断是否为文件的问题	393	14.17	如何实现鼠标双击事件	474
12.12	怎么实现文件过滤	396	14.18	怎么用 new ActionListener()实例化接口呢	479
12.13	RandomAccessFile("D:\\a.txt", "w")不能写数据	398	第 15 章	Swing 程序开发	486
第 13 章	集合	401	15.1	Java 的 AWT 和 Swing 有什么区别	486
13.1	两个集合的相加与相减问题	401	15.2	如何为 JFrame 设置背景图片	489
13.2	怎么去掉重复数据	404	15.3	如何设置按钮边框和标签之间的空白	492
13.3	向 TreeSet 中添加对象报 ClassCastException 异常	410	15.4	JTabbedPane 标签文字的显示问题	497
13.4	Set 集合是如何判断重复元素的	412	15.5	JTable 的表头为什么不显示	500
13.5	怎么让 TreeSet 集合中的元素降序排列	414	15.6	如何让 JTree 的节点默认全部展开	503
13.6	new ArrayList<String>(8)能限制集合长度为 8 吗	417	15.7	JFileChooser 如何添加文件类型过滤	507
13.7	Map 中的元素是不是无序的	420	15.8	报 BorderLayout can't be shared 错误	510
13.8	如何获取集合中值最大或最小的元素	424	15.9	如何根据对话框的选项执行相应的操作	512
第 14 章	图形编程基础	428	第 16 章	Java 绘图与音频播放	519
14.1	Frame 与 Panel 是什么关系	428	16.1	为什么要用 Graphics2D g2d = (Graphics2D) g	519
14.2	无法在窗口里显示 Label	434	16.2	绘图时如何设置线条的粗细	522
14.3	AWT 窗口中怎么设置按钮的颜色	437	16.3	怎么修改文本的字体	527
14.4	使用 TextField 如何实现密码框	439	16.4	移动原点对坐标的影响	529
14.5	TextArea 的滚动条的问题	441	16.5	在坐标 (0, 0) 处绘制的文本不显示	534
14.6	如何让复选框默认选中	444	16.6	怎么用彩色填充图形	538
14.7	如何实现单选按钮	445	16.7	Java 绘图设置渐变色, 抛出 IllegalArgumentException Exception 异常	541
14.8	使用 List 组件出现错误	447	16.8	缩放图片, 出现编译错误	544
14.9	使用 Choice 组件出现错误	449	16.9	图片上添加水印	547
14.10	为什么不显示菜单	452	16.10	怎么绘制一张图片中间的一部分	550
14.11	如何实现左对齐	458	16.11	怎么让图片的上方向右倾斜	553
14.12	Panel 上不显示组件	460	16.12	Java 可以播放哪些格式的音频文件	554
14.13	使用 GridLayout 布局能否指定某行某列	462	第 17 章	JDBC 数据库编程	558
14.14	使用 CardLayout 报 cannot add to layout 错误	464	17.1	Java 有几种连接数据库的方法	558
14.15	使用 GridBagLayout 时, 需要“占位符”吗	467	17.2	JDBC 连接数据库的步骤	562
			17.3	关于 createStatement()的参数问题	566



17.4	DriverManager.getConnection()中的 参数问题	568
17.5	statement 和 PreparedStatement 之间的 区别	570

17.6	ResultSet 的 absolute()方法是 做什么用的	574
17.7	execute()、executeQuery()和 executeUpdate()	577

第四篇 Java 综合案例篇

第 18 章 综合案例——蜘蛛纸牌

582

18.1	案例分析	582
18.1.1	游戏菜单栏设计	582
18.1.2	游戏界面设计	582
18.1.3	游戏功能设计	583
18.2	案例准备	584
18.3	案例开发	585
18.3.1	菜单栏	586
18.3.2	为菜单添加事件监听器	588
18.3.3	纸牌	592
18.3.4	为纸牌添加鼠标监听器	596
18.3.5	实现纸牌闪烁	601
18.3.6	窗口	602
18.3.7	为发牌区添加鼠标监听器	604
18.3.8	为游戏窗口添加键盘监听器	605
18.3.9	开始新游戏	606
18.3.10	初始化纸牌	606
18.3.11	随机分配纸牌	607
18.3.12	设置纸牌的坐标值	608
18.3.13	获得上一张纸牌与 下一张纸牌	611
18.3.14	获取某列最后一张 纸牌的坐标	612
18.3.15	获得某列的背景框的坐标	612
18.3.16	发牌	613
18.3.17	显示可行操作	615
18.3.18	判断纸牌是否组合完成	617

第 19 章 综合实例——超市管理系

619

19.1	系统设计	619
19.1.1	需求分析	619
19.1.2	功能设计	619
19.2	数据库设计	620
19.2.1	数据表	621
19.2.2	视图	624
19.3	系统开发前的准备工作	624
19.4	基础类	626
19.4.1	数据库连接类	626
19.4.2	实体类	628
19.4.3	数据库操作类	629
19.5	登录	630
19.5.1	登录界面的背景面板	630
19.5.2	登录界面	631
19.6	主界面	635
19.6.1	主界面中的组件与布局	635
19.6.2	创建主界面的导航标签面板	637
19.7	基础信息管理	642
19.7.1	商品信息管理	642
19.7.2	获得所有商品	644
19.8	商品查询	646
19.8.1	商品信息查询面板	646
19.8.2	显示全部数据	648
19.8.3	显示按条件查询的数据	652
19.9	生成报表	654

绪 论

在捧起这本书之前,相信读者您已经通过许多途径感受到了 Java 语言的强大。不错,Java 无处不在,它可以应用于任何地方、任何领域,其发展速度要快于在它之前的任何一种计算机语言。在将要开始学习 Java 这门强大的编程语言之前,就让我们更深层次地认识一下 Java 吧!

0.1 Java 的发展史

Java 是由 Sun Microsystems 公司的 James Gosling 领导的工作组开发的,其最初的名称为 Oak。Java 起源于 Sun 公司的一个名为 Green 的项目,目的是开发嵌入家用电器的分布式软件系统,使电器更加智能化。Green 项目开始时准备采用 C++ 语言,但是考虑到 C++ 语言太复杂,而且安全性差,于是决定基于 C++ 语言开发一种新的语言,即 Oak 语言。

Oak 语言保留了许多 C++ 语言的语法,但是去除了明确的资源引用、指针算法与操作符重载等潜在的危險特性。当 Oak 语言成熟后,随着 Internet 的迅猛发展, Sun 公司的开发小组用 Oak 语言编写了一个早期的 Web 浏览器,称为 WebRunner (后改名为 HotJava)。

1995 年, Oak 语言正式更名为 Java 语言。据说 Java 这个名称来自咖啡,我们可以看到 Java 语言的图标就是一杯咖啡。

0.2 Java 的特征

Sun 公司对 Java 语言的特征是这样描述的: Java 是简单的、面向对象的、分布式的、解释性的、强大的、安全的、与体系结构无关的、可移植的、高性能的、多线程的及多态的。怎么理解这些修饰语呢?一起来看看吧。

1. Java 是简单的

Java 的简单性是对比其他流行编程语言来说的。Java 是基于 C++ 语言的基础开发出来的一种语言,它有很多部分模仿了 C++ 语言,但是很大程度上对 C++ 进行了简化和改进,例如 Java 用一种称为“接口”的简单语言构造取代了 C++ 语言中的多重继承,并取消了指针(指针和多重继承通常会让编程变得非常复杂)。

另外, Java 使用动态内存分配和无用单元收集,而 C++ 则需要程序员自己分配和收集无用单元,因此对比 C++ 语言来说, Java 让程序员更加“解脱”。

2. Java 是面向对象的

Java 自诞生之时就被设计成面向对象的语言,而 C++ 语言则是一种强制面向对象的语言。面向对象可以说是 Java 最重要的特性,它不支持类似 C 语言那样的面向过程的程序设计技术。

面向对象编程以“对象”模拟现实世界。世界上的每件事物都可以模拟为对象，如一个人、一辆车、一朵花等都是一个对象。

多年来，面向对象技术被认为是杰出的成果，而 Java 语言则帮助面向对象技术进入了计算机的主流。

3. Java 是分布式的

除了支持基本的语言功能以外，Java 核心类库还包括一个支持 HTTP、SMTP 和 FTP 等基于 TCP/IP 协议的类库。因此，Java 应用程序可凭借 URL 打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。在 Java 出现之前，以分布式环境尤其是 Internet 提供动态的内容无疑是一项非常宏伟、难以想象的任务，但 Java 语言的分布式特性则可以让人们很容易地达到这个目标。

4. Java 是解释性的

运行 Java 程序需要解释器，程序被编译成为字节代码的 Java 虚拟机（Java Virtual Machine, JVM）代码。字节代码与机器无关，可以在任何有 Java 解释器的机器上运行。

而通常，编译器（例如 C++ 编译器）将高级语言程序转换为机器代码，这些代码只能在本地计算机上运行，如果需要在其他计算机上运行这些程序，就必须在该计算机上进行重新编译。

5. Java 是安全的

Java 致力于检查程序在编译和运行时的错误，类型检查能够检查出许多早期开发时出现的错误。Java 支持自动内存管理，这不但让程序员减轻了许多负担，也让程序员减少了犯错的机会。Java 自己操纵内存，减小了内存出错的可能性。Java 还能够检测数组边界，避免了覆盖数据的可能。在 Java 语言中，指针和释放内存等功能均被抛弃，从而避免了非法内存操作的危险。

6. Java 是与体系无关的、可移植的

Java 是解释性的，这一特征让 Java 与体系无关，即独立于平台。使用 Java 虚拟机可以编写一个在任何平台上都可以运行的程序，也就是说，Java 开发者只需要编写一个在某个平台可以运行的版本即可，这个版本可以在任何平台上运行。这也正好说明了 Java 的可移植性。

7. Java 是高性能的

因为 Java 是可解释的，其字节代码不是直接由系统执行，而是在解释器中运行，因此它的速度对多数交互式应用程序来说已经足够了。

早期 Sun Java 虚拟机的用户会觉得 Java 运行起来很慢，不过新 Java 虚拟机做了改进。新 JVM 使用称为实时编译的技术，将字节代码编译为本机代码，保存本机代码，并且在字节代码执行时重新调用本机代码。Sun 公司新开发的 Java HotSpot Performane Engine 包含对经常使用的代码进行优化的编译器，它可以插入到 JVM 中，以显著提高其性能。

8. Java 是多线程的

多线程是程序同时执行多个任务的功能，例如在播放视频文件的同时下载视频文件就可以认为是多线程的。多线程编程被完美地集成到 Java 语言中，而在其他语言中必须调用特定于操作系统的过程才能实现多线程。

在网络编程中，一个服务器可以同时为多个客户提供服务，这也是多线程的一种体现。

9. Java 是动态的

Java 的设计适用于进化的环境，新代码可以不用重新编译就能载入。开发人员不需要创建主要的新软件版本，用户也不需要将其安装，新特征可以在需要的时候透明地注入。

0.3 Java 的工作原理

Java 是一种跨平台的语言，同时也是既编译又解释的语言。Java 可以运行在网络不同的计算机上，但运行的计算机上必须装有相应的 Java 解释器。而这种解释器就是 Java 虚拟机。

1. Java 虚拟机介绍

Java 语言的一个非常重要的特点就是与平台的无关性，而使用 Java 虚拟机是实现这一特点的关键。一般的高级语言如果要在不同的平台上运行，至少需要编译成不同的目标代码。而引入 Java 语言虚拟机后，Java 语言在不同平台上运行时不需要重新编译。Java 语言使用模式 Java 虚拟机屏蔽了与具体平台相关的信息，使得 Java 语言编译程序只需生成在 Java 虚拟机上运行的目标代码（字节码），就可以在多种平台上不加修改地运行。Java 虚拟机在执行字节码时，把字节码解释成具体平台上的机器指令执行。

Java 虚拟机是一个虚构出来的计算机，是通过在实际的计算机上仿真模拟各种计算机功能来实现的，可运行 Java 字节码的虚拟计算机系统。Java 虚拟机有自己完善的硬件架构，如处理器、堆栈、寄存器等，还具有相应的指令系统。Java 虚拟机在执行字节码时，实际上最终还是把字节码解释成具体平台上的机器指令执行。即在机器和编译程序之间加入了一层抽象的虚拟机器。这台虚拟机器在任何平台上都提供给编译程序一个共同的接口。编译程序只需要面向虚拟机，生成虚拟机能够理解的代码，然后由解释器来将虚拟机代码转换为特定系统的机器码执行。

Java 虚拟机在计算机上的位置如图 1 所示。

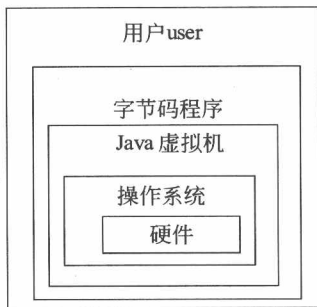


图 1 Java 虚拟机在计算机上的位置

字节码的执行需要经过以下 3 个步骤。

- (1) 首先，由类装载器（Class Loader）负责把类文件（.class 文件）加载到 Java 虚拟机中，在此过程需要检验该类文件是否符合类文件规范。
- (2) 其次，字节码校验器（Bytecode Verifier）检查该类文件的代码中是否存在着某些非

法操作。

(3) 如果字节码校验器检验通过, 由 Java 解释器负责把该类文件解释成为机器码进行执行。Java 虚拟机采用的是“沙箱”运行模式, 即把 Java 程序的代码和数据都限制在一定内存空间里执行, 不允许程序访问该内存空间外的内存。如果是 Applet 程序, 还不允许访问客户端机器的文件系统。

Java 虚拟机解释执行的过程如图 2 所示。

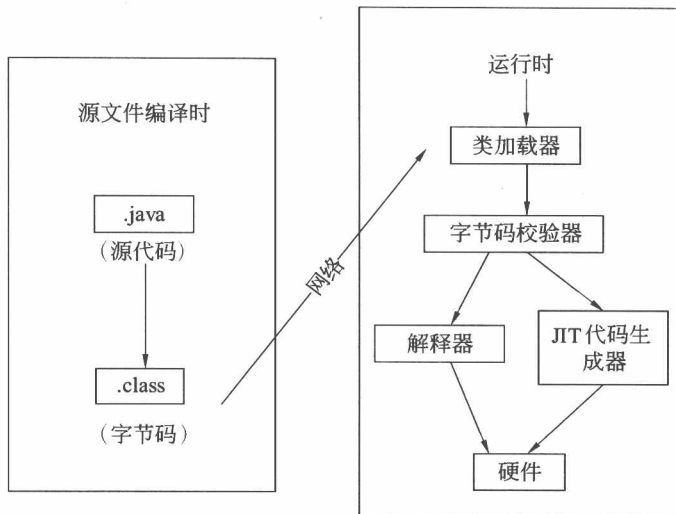


图 2 Java 虚拟机的运行过程

Java 虚拟机的执行引擎有 3 种, 分别如下:

□ 常用的解释器。

□ JIT (Just InTime) 是一个可以将 Java 字节码转换为本地机器码的代码生成器。由 JIT 调用的 Java 程序, 其运行速度通常要比由解释程序执行字节码时的速度高得多。这种情况下, 第一次执行的字节码会被编译成本地的机器代码, 缓存在系统中, 以后可以随时调用。

□ 还有一种自适应的优化器, 这种方法里, 虚拟机开始时解释字节码, 但是会监控运行中程序的活动, 并记录下使用最频繁的代码段。虚拟机会把这些活动最频繁的代码段编译成本地代码。

其实, 以上提到的都是软件实现的虚拟机。还有一种虚拟机是由硬件芯片构成的, 它用本地方法执行 Java 字节码。

2. Java 的运行流程

Java 可以编写各种各样的程序, 如单机应用程序和 Web 程序等。但不管哪种 Java 程序, 其运行的流程大致如图 3 所示。

从图 3 中可以观察到, 首先应该创建源程序, Java 源程序一般用 .java 作为扩展名, 是一个文本文件, 用 Java 语言写成, 可以用任何文本编辑器创建与编辑; 源程序完成后, 就可以进行编译, 需要使用 Java 编译器 javac, 读取 Java 源程序并翻译成 Java 虚拟机能够明白的指令集合, 且以字节码的形式保存在文件中。通常, 字节码文件以 .class 作为扩展名; Java 解释



器（即 java），读取字节码，取出指令并且翻译成计算机能执行的代码，完成运行过程；该字节码运行的平台是 Java 虚拟机。只要计算机上装有 Java 虚拟机，无论采用哪种操作系统，硬件配置情况如何，运行的结果都是一样的。

可以看出，Java 源程序首先被编译器编译成字节码文件，然后被 JVM 加载，解释执行。

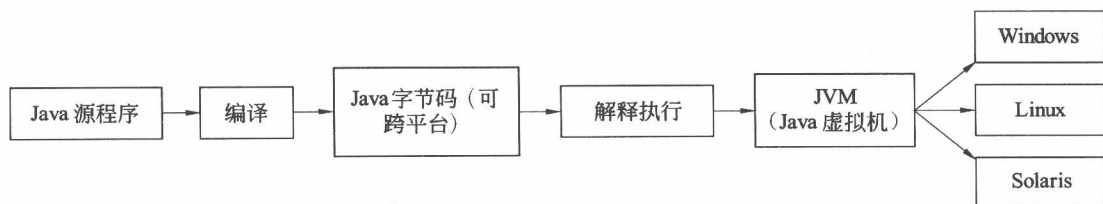


图 3 Java 程序运行的流程

0.4 Java 的应用领域

Java 程序可以在计算机、电视、电话、手机和其他大量设备上运行，其用途数不胜数，功能也非常强大。Java 的应用领域主要包括以下内容。

- ☐ 桌面应用系统开发。
- ☐ 嵌入式系统开发。
- ☐ 电子商务应用开发。
- ☐ 企业级应用开发。
- ☐ 交互式系统开发。
- ☐ 多媒体系统开发。
- ☐ 分布式系统开发。
- ☐ Web 应用系统开发。

其实，我们经常使用的 Java 开发工具，如 Eclipse、NetBeans 和 JBuilder 等都是使用 Java 语言编写的，另外还有 RSSOwl、LimeWire、Azureus、CyberDuck 和 OpenOffice 等优秀软件也都是使用 Java 语言编写的。

0.5 Java 的版本

Java 发展至今，按应用范围可以分为 3 个版本，即 Java SE、Java EE 和 Java ME。

1. Java SE

Java SE 是 Java 的标准版，主要用于桌面应用程序的开发，同时也是 Java 的基础，它包含 Java 语言基础、JDBC 数据库操作、I/O（输入输出）流、网络通信和多线程等技术。

2. Java EE

Java EE 是 Java2 的企业版，主要用于开发企业级分布式的网络程序，如电子商务网站和 ERP 系统，其核心是 EJB。

3. Java ME

Java ME 主要应用于嵌入式系统开发,如掌上电脑和手机等移动通信电子设备,现在大部分手机厂商所生产的手机都支持 Java 技术。

0.6 Java API 文档

Java API (Application Programming Interface, 应用程序编程接口) 文档是 Java 程序开发不可缺少的编程词典,它记录了 Java 语言中海量的 API,主要包括类的继承结构、成员变量、成员方法、构造方法、静态成员的详细说明和描述信息,可以在 Java 官方网站 <http://java.sun.com> 中找到最新的 Java API 文档。Java API 文档的页面效果如图 4 所示。



图 4 Java API 文档

0.7 Java 开发工具

目前有许多优秀的 Java 开发工具,如 Eclipse 和 NetBeans 等,这两种是比较常用的 Java 开发工具,而另一款流行的 Java 开发工具 MyEclipse 则是对 Eclipse 的扩展。下面介绍 Eclipse 与 NetBeans 的安装与使用。

1. Eclipse 开发工具

Eclipse 是一个开放源代码的软件开发项目,专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。它主要由 Eclipse 项目、Eclipse 工具项目和 Eclipse 技术项

目组成。

(1) 下载并安装 Eclipse

由于 Eclipse 是一个开放源代码的项目，可以到其官方网站 www.eclipse.org 上免费下载 Eclipse 的最新版本。虽然 Eclipse 本身用 Java 语言编写，但下载的压缩包中并不包含 Java 运行环境（即安装 Eclipse 平台，应首先安装 JDK 工具包），需要用户自己另行安装 JRE，并且要在操作系统的环境变量中指明 JRE 中 bin 的路径。

安装 Eclipse 的步骤非常简单，将下载的压缩包按原路径直接解压即可。须注意，如果有更新的版本，要先删除老的版本，之后才能重新安装；不能直接解压到原来的路径，从而覆盖老版本。在解压缩之后可以到相应的安装路径运行 Eclipse.exe。启动 Eclipse，会出现一个由日蚀图片构成的闪屏，随后出现一个如图 5 所示的选择工作区路径对话框，Eclipse 会将编辑的所有文件存放在工作区指定的路径下。

确定工作区路径后单击 OK 按钮，打开如图 6 所示的 Eclipse 欢迎主界面。

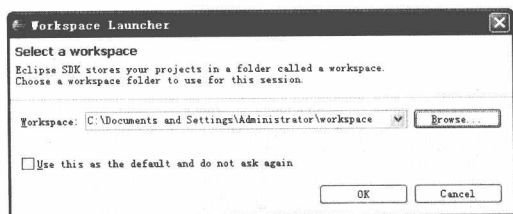


图 5 选择工作区路径

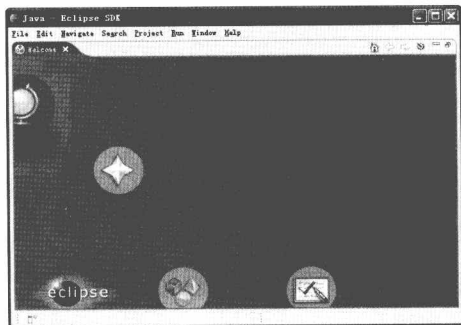


图 6 Eclipse 欢迎界面

如果在未安装 JDK 前启动 Eclipse，系统会出现如图 7 所示的错误信息。提示用户先安装 JDK，并正确配置后再重新启动。

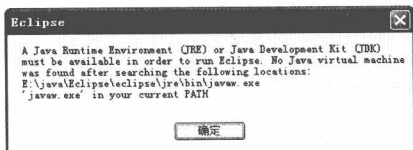


图 7 错误信息提示

(2) Eclipse 开发程序

使用 Eclipse 开发 Java 项目非常简单，首先创建相应的 Java 项目，然后创建 Java 源文件即可。现在我们创建一个 Java 项目，在里面建立一个类，输出“中国，你好”字符串。接下来演示一下 Eclipse 的操作及每个菜单项的功能。

选择 Eclipse 工作台的【文件】|【新建】|【项目】命令，就会弹出一个如图 8 所示的对话框。在这里，双击【创建 Java 项目】选项，就会弹出如图 9 所示的对话框。

在【项目名】文本框中输入 JavaTest，对话框下面的信息主要是用来显示项目的路径、Java 虚拟机的类型及布局，其他值的选项用默认设置就可以。在输入项目名称后，单击【下