

Java 程序设计 循序渐进教程



北京希望电子出版社
程桓松 周晓军 陈垦

总策划
编 著

step

by

- 你想学习风靡全球的主编程语言Java吗? 本书带你轻松进入Java世界!
- 生动的实例: 带你了解从规划、创建资料、编写代码到运行程序的完整流程。
- 精炼的小结: 让你在最短时间内掌握Java语言的基本思想。
- 有趣的练习: 令你不知不觉中领悟多多。
- 广阔的空间: 诠释Java的编程方法与思想, J2EE和J2ME的概念。

step



电子工业出版社



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

Java 程序设计 循序渐进教程

北京希望电子出版社
程桓松 周晓军 陈垦

总策划
编 著

江苏工业学院图书馆
藏书章

step

- 你想学习风靡全球的主流编程语言Java吗? 本书带你轻松进入Java世界!
- 生动的实例: 带你了解从规划、创建资料、编写代码到运行程序的完整流程。
- 精炼的小结: 让你在最短时间内掌握Java语言的基本思想。
- 有趣的练习: 令你不知不觉中领悟多多。
- 广阔的空间: 诠释Java的编程方法与思想, J2EE和J2ME的概念。

step



电子科技大学出版社



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本书是一本介绍 Java 程序设计的书,系统地介绍了 Java 语言的主要功能及用法。

全书由 16 章组成。包括 Java 编程环境、基本语法、面向对象编程,Java 的数据结构,字符串操作,接口与包,异常处理,Java 的 IO 系统,多线程编程,Java 小程序 Applet,操作数据库,网络编程,Servlet 编程基础和 J2EE 与 J2ME 的技术简介。本书从最基本的 Java 基本数据类型和基本语法开始讲起,结合面向对象的编程思想,并详尽地介绍了 Java 的 API,并含有大量精心设计的代码实例。

本书语言生动流畅、没有晦涩的专业术语,能使读者迅速掌握利用 Java 语言的方法和技巧。本书主要面向初中级读者。如果读者具备了 C++语言基础,那么阅读本书更会事半功倍,迅速掌握 Java 语言的开发技巧;如果你是新手上路,那么本书可以作为一本绝佳教程,它让你在了解 Java 语言的同时,掌握面向对象程序设计思想要点及与 Java 语言相关的各项技术。

需要本书或技术支持的读者,请与北京中关村 083 信箱(邮编 100080)发行部联系,电话:010-82702660,010-82702658,010-62978181 转 103 或者 238,传真:010-82702698,E-mail:tbd@bhp.com.cn。

图书在版编目(CIP)数据

Java 程序设计循序渐进教程 / 程桓松,周晓军,陈垦编著.

成都:电子科技大学出版社,2005.7

(新世纪热门软件步步高丛书)

ISBN 7-81094-813-X

I. J... II. ①程...②周...③陈... III. Java 语言—程序—设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 035120 号

Java 程序设计循序渐进教程

程桓松 周晓军 陈垦 编著

-
- 出 版:** 电子科技大学出版社(成都建设北路二段四号 610054)
北京希望电子出版社(北京海淀区上地信息产业基地三街 9 号,金隅嘉华大厦 C 座 610 100085)
网址: www.bhp.com.cn E-mail:lwm@bhp.com.cn zmh@bhp.com.cn
电话: 010-82702660, 82702658, 62978181 转 103 或 238, 传真: 010-82702698
- 责任编辑:** 周凤明 谢晓辉
- 发 行:** 新华书店经销
- 印 刷:** 北京东升印刷厂
- 开 本:** 787mm×1092mm 1/16 印张 24.5 字数 562 千字
- 版 次:** 2005 年 7 月第 1 版
- 印 次:** 2005 年 7 月第 1 次印刷
- 书 号:** ISBN 7-81094-813-X / TP·440
- 印 数:** 0001-5 000 册
- 定 价:** 28.00 元
-

前言

无论你是一位专业软件开发人员，还是一名普通的编程爱好者，Java 对你而言一定耳熟能详。Java 以其简单性、面向对象、分布性、编译和解释性、安全性、可移植性、高性能以及多线程性等特性为设计人员能够编写出完美的应用程序提供了支持，特别是与 Internet 的成功结合，使得 Java 成为风靡全球的主流编程语言。利用 Java 所提供的丰富的类库，程序设计者可以迅速建立自己的系统。

本书力图全面地介绍 Java 语言，并将 Java 语言基础与面向对象程序设计方法结合起来，深入浅出地介绍了 Java 的编程方法和编程思想。本书还通过大量生动的实例详细介绍了 Java 语言各方面的程序编程接口（API），通过这些实例，读者能迅速掌握 Java 语言的实际开发技术细节。

□ 本书读者对象

本书主要面向初中级读者。如果读者具备了 C++ 语言基础，那么阅读本书你会事半功倍，迅速掌握 Java 语言的开发技巧；如果你是新手上路，那么本书可以作为一本绝佳教程，它让你在了解 Java 语言的同时，掌握面向对象程序设计思想的要点以及与 Java 语言相关的各项技术。

□ 本书特色

这是一本关于 Java 的书，它循序渐进地介绍了 Java 语言的编程环境和基本语法，以及面向对象程序开发的基本概念和方法，精辟地讲解了 Java 的数据结构、I/O 系统以及多线程编程，并用实用范例导入了 Java 的网络 API，最后读者将逐步掌握 Java 在 Web 编程中、在企业级应用、在开发无线方面的应用。本书的特色如下：

生动的实例。每一章都包含一个或者数个程序设计的应用实例，你将了解从规划、创建资料、编写代码到运行程序的完整流程。

精炼的小结。在每一章的结尾都有一个小结，这是对全章内容的概括，你可以在最短的时间内掌握每章的基本思想。

有趣的练习。本书提供了大量实用、有趣的练习，希望你能花费些时间完成这些练习，以此起步，以期领悟更多。

广阔的空间。本书不仅仅介绍了 Java 语言本身，还详细讲述了 Java 平台各方面的编程接口，其中包括 Java 2SE1.4 版本才引入的 NIO 等，以及目前 Java 技术应用的两大热点领域 J2EE 和 J2ME，你将充分理解 Java 带给我们的新观念。

□ 本书章节安排

本书章节安排是这样的:

第一章 Java 简介 对 Java 语言的历史、特点和开发平台进行了简明扼要的介绍。

第二章 Java 编程环境 介绍了 Java 的运行环境和开发环境。

第三章 Java 基本语法 介绍了 Java 语言的常量、变量、简单数据类型、运算符和程序流程控制。

第四章 Java 与面向对象编程 讲述了面向对象程序设计的基本概念和特点,介绍了 Java 中的类和对象的使用、继承以及方法和成员变量的权限控制。

第五章 Java 的数据结构 重点讲解了 Java 中的数组和 Java 集合框架。

第六章 字符串操作 对字符串操作的了解可以使编写文本处理程序更加简单、迅速。

第七章 接口和包 主要介绍了 Java 接口和包的概念和使用方法。

第八章 异常处理 主要介绍了异常的定义,介绍了异常的捕捉与处理以及标准异常和自定义异常。

第九章 Java 的 I/O 系统 本章主要介绍了 Java 语言中 IO 流的体系结构,重点说明了 Java 的流,文件系统访问以及 NIO。

第十章 多线程编程 深入介绍了线程的概念,详细探讨了如何实现 Java 的线程以及线程同步的概念和应用。

第十一章 Java 小程序 Applet 介绍 Java Applet 的运行原理,和 Applet 作为一种特殊的 Java 类的特性,以及如何编写自己的 Applet 类并将其应用到网页中。

第十二章 操作数据库 讲述 Java 的 JDBC 技术。

第十三章 网络操作 介绍了网络的基本概念,Java 中流套接字和数据报套接字的使用,以及 Java 的新 IO 系统所提供的网络 API。

第十四章 Servlet 编程基础 讲解了 Servlet 和 JSP 技术。

第十五章 J2EE 技术简介 介绍了 J2EE 的体系结构,J2EE 应用程序的开发流程和开发环境。

第十六章 J2ME 技术简介 本章重点讲述 J2ME 的 MIDP 程序设计方法,并介绍了 MIDP 的图形界面元件。

□ 本书编写人员

本书由程恒松、周晓军、陈垦共同执笔编写。此外,潘家铭、郑任持、蓝荣香、王昊亮、喻波、马天一、魏勇、郝荣福、李光龙、孙明、李大宇、武思宇、牟博超、李彬、付鹏程、高翔、朱丽云、崔凌、张巧玲、李辉、李欣、柏宇、郭强、金春范、程梅、黄霆、钟华、张浩、刘湘和邵蕴秋等同志在整理材料方面给予了编者很大的帮助,在此,编者对他们表示衷心的感谢。

由于时间仓促,加之作者的水平有限,恳请专家和广大读者不吝赐教,批评指正。

编 者

目 录

第1章 Java 简介.....	1	3.3.6 条件运算符.....	31
1.1 Java 语言的历史.....	1	3.3.7 []运算符.....	31
1.2 Java 语言的特点.....	2	3.3.8 点运算符.....	31
1.3 本章小结.....	5	3.3.9 new 运算符.....	31
1.4 习题.....	6	3.3.10 instanceof 运算符.....	31
第2章 Java 编程环境.....	7	3.4 Java 的表达式和语句.....	32
2.1 Java 运行环境和开发环境.....	7	3.4.1 表达式.....	32
2.2 开发环境的安装和使用.....	8	3.4.2 语句.....	32
2.2.1 Java2SDK 简介.....	8	3.4.3 语句块.....	33
2.2.2 安装 Java2SDK.....	8	3.5 程序流程控制.....	33
2.2.3 Java2SDK 的使用.....	10	3.5.1 循环控制.....	33
2.3 编辑器选择.....	11	3.5.2 程序分支.....	35
2.3.1 Ultraedit 编辑器.....	11	3.5.3 break 语句和 continue 语句.....	38
2.3.2 Emacs 编辑器.....	12	3.5.4 返回.....	40
2.4 Java 程序的结构.....	13	3.5.5 例外 (Exception).....	40
2.5 程序调试.....	15	3.6 本章小结.....	41
2.6 本章小结.....	17	3.7 上机练习.....	41
2.7 上机练习.....	18	3.8 习题.....	42
2.8 习题.....	18	第4章 Java 与面向对象编程.....	44
第3章 Java 基本语法.....	19	4.1 面向对象编程.....	44
3.1 Java 的常量和变量.....	19	4.1.1 面向对象编程概述.....	44
3.1.1 常量.....	19	4.1.2 面向对象程序设计思想.....	45
3.1.2 变量.....	19	4.2 类和对象的使用.....	47
3.2 Java 的简单数据类型.....	20	4.2.1 类的定义.....	47
3.2.1 字符和字符串.....	20	4.2.2 成员变量.....	48
3.2.2 整型.....	21	4.2.3 方法.....	49
3.2.3 浮点型.....	22	4.2.4 创建对象和初始化.....	50
3.2.4 布尔型.....	22	4.2.5 成员变量引用和方法调用.....	51
3.2.5 数据类型转换.....	23	4.2.6 this 关键字.....	53
3.3 Java 的运算符.....	23	4.2.7 对象资源回收和 finalize () 方法.....	54
3.3.1 算术运算符.....	23	4.2.8 对象比较.....	56
3.3.2 位运算符.....	26	4.2.9 对象拷贝.....	58
3.3.3 赋值运算符.....	28	4.3 类.....	59
3.3.4 关系运算符.....	28	4.3.1 类的修饰符.....	59
3.3.5 布尔运算符.....	30	4.3.2 嵌套类.....	60

4.4 成员变量.....	64	5.3.2 Set 接口.....	109
4.4.1 成员变量的修饰符.....	64	5.3.3 List 接口.....	119
4.4.2 实例变量和类变量.....	65	5.3.4 Map 接口.....	124
4.4.3 变量初始化.....	67	5.3.5 集合同步.....	130
4.5 方法.....	68	5.4 算法.....	132
4.5.1 方法声明.....	68	5.4.1 排序.....	133
4.5.2 方法的参数.....	68	5.4.2 混排.....	134
4.5.3 方法重载.....	70	5.4.3 查找.....	134
4.5.4 构造方法.....	71	5.4.4 最大值、最小值.....	134
4.5.5 方法的修饰符.....	73	5.4.5 其他.....	135
4.5.6 静态方法.....	73	5.5 早期 Java 的集合.....	135
4.6 类的继承.....	75	5.5.1 Vector.....	135
4.6.1 继承.....	75	5.5.2 BitSet.....	135
4.6.2 成员变量的继承和隐藏.....	75	5.5.3 Stack.....	136
4.6.3 方法继承、方法覆盖和方法隐藏.....	76	5.5.4 Hashtable.....	137
4.6.4 super 关键字.....	78	5.6 本章小结.....	137
4.6.5 抽象类.....	80	5.7 上机练习.....	138
4.6.6 抽象方法.....	82	5.8 习题.....	138
4.6.7 final 关键字.....	85	第 6 章 字符串操作.....	141
4.6.8 单继承和多继承.....	86	6.1 字符和字符串.....	141
4.7 方法和成员变量的权限控制.....	87	6.2 String 类.....	141
4.7.1 public.....	87	6.2.1 String 类的构造函数.....	141
4.7.2 protected.....	88	6.2.2 字符串比较.....	142
4.7.3 private.....	90*	6.2.3 hashCode () 方法.....	143
4.7.4 缺省情况.....	91	6.2.4 定位字符和子串.....	144
4.7.5 存取方法.....	93	6.2.5 抽取子串.....	145
4.8 本章小结.....	94	6.2.6 字符串修改.....	146
4.9 上机练习.....	95	6.2.7 String 类的其他方法.....	146
4.10 习题.....	96	6.3 StringBuffer 类.....	147
第 5 章 Java 的数据结构.....	99	6.3.1 StringBuffer 类的构造函数.....	147
5.1 数组.....	99	6.3.2 length ()、capacity ()、 setLength () 和 ensureCapacity () 方法.....	148
5.1.1 一维数组.....	99	6.3.3 charAt ()、setCharAt ()、 getChars () 和 reverse () 方法.....	149
5.1.2 多维数组.....	102	6.3.4 Append () 方法.....	150
5.2 集合简介.....	104	6.3.5 插入和删除.....	151
5.2.1 Java 集合框架 (Java Collection Framework) 概述.....	105	6.3.6 StringBuffer 类的其他方法.....	152
5.2.2 Java 集合的发展历史.....	106	6.4 StringTokenizer 类.....	152
5.3 集合接口说明.....	106		
5.3.1 Collection 接口.....	106		

6.4.1	StringTokenizer 类的构造函数	152
6.4.2	StringTokenizer 类的常用方法	152
6.4.3	改进的 StringTokenizer 类: BreakIterator 类	154
6.5	使用正则表达式	157
6.5.1	正则表达式	157
6.5.2	Pattern 类	159
6.5.3	Matcher 类	159
6.5.4	操作符优先级	161
6.5.5	正则表达式总结	161
6.6	本章小结	163
6.7	上机练习	163
6.8	习题	164
第 7 章	接口和包	166
7.1	接口概述	166
7.1.1	什么是接口	166
7.1.2	接口定义说明	167
7.2	接口的定义	167
7.2.1	接口的声明	167
7.2.2	基接口	168
7.3	定义接口成员	168
7.4	访问接口	169
7.5	接口与抽象类	171
7.5.1	抽象类与接口的区别	171
7.5.2	使用抽象类和接口的场合	171
7.5.3	抽象类实现接口	173
7.6	包	174
7.6.1	package 语句	174
7.6.2	import 语句	174
7.6.3	编译和运行	175
7.7	本章小结	176
7.8	上机练习	176
7.9	习题	176
第 8 章	异常处理	178
8.1	异常处理概述	178
8.1.1	为什么要使用异常处理	178
8.1.2	Java 中的异常机制	179
8.1.3	异常的产生	179
8.2	异常的捕捉和处理	179

8.2.1	捕捉和处理: try-catch 语句	180
8.2.2	抛出异常 (Throw Exception): throw 语句和 throws 语句	181
8.2.3	缺省处理 (finally 语句)	181
8.3	Java 的标准异常	181
8.3.1	异常类 (Exception 类)	182
8.3.2	标准异常中 Exception 类的 派生类	183
8.4	自定义异常	183
8.4.1	编写自定义异常	183
8.4.2	捕捉并处理自定义异常	184
8.4.3	finally 处理	186
8.5	本章小结	187
8.6	上机练习	187
8.7	习题	187
第 9 章	Java 的 IO 系统	190
9.1	Java 的 IO 系统概述	190
9.1.1	流的概念	190
9.1.2	Java IO 系统介绍	190
9.1.3	字符流 API	191
9.1.4	字节流 API	191
9.2	字符流	191
9.2.1	文件流	191
9.2.2	管道流	192
9.3	字节流	193
9.3.1	文件流	193
9.3.2	管道流	194
9.3.3	过滤器流	195
9.4	文件和目录管理	204
9.4.1	File 类	204
9.4.2	遍历目录	204
9.4.3	创建、删除和重命名文件目录	206
9.4.4	文件或目录信息获取	207
9.5	RandomAccessFile 类	209
9.6	Java 的新 IO 系统: NIO	210
9.6.1	传统 IO 系统的缺点	210
9.6.2	NIO 概述	211
9.6.3	缓存 (Buffer)	211
9.6.4	字符集 (Charset)	214

9.6.5 通道 (Channel)	215	11.3.2 鼠标事件处理	270
9.6.6 选择器 (Selector)	222	11.4 本章小结	273
9.6.7 NIO 应用实例	225	11.5 上机练习	273
9.7 本章小结	227	11.6 习题	273
9.8 上机练习	228	第 12 章 操作数据库	274
9.9 习题	228	12.1 JDBC 技术	274
第 10 章 多线程编程	230	12.1.1 JDBC 技术简介	274
10.1 多线程概述	230	12.1.2 JDBC 技术规范	275
10.1.1 什么是多线程	230	12.2 SQL 语言	276
10.1.2 为什么要使用多线程	231	12.2.1 删除和修改表	276
10.2 建立 Java 线程	231	12.2.2 SELECT 语句	277
10.2.1 计时器	231	12.2.3 插入、删除和修改记录	278
10.2.2 Thread 类	233	12.2.4 聚类函数	279
10.2.3 Runnable 接口	239	12.3 JDBC 基本操作	279
10.3 线程优先级	241	12.3.1 加载 JDBC 驱动	279
10.3.1 线程优先级和线程调度	241	12.3.2 连接数据库	280
10.3.2 用户线程和守护线程	244	12.3.3 Statement 接口	280
10.4 线程同步	244	12.3.4 结果集	282
10.4.1 生产者-消费者问题	244	12.3.5 元数据集	282
10.4.2 对象锁	248	12.3.6 错误处理	283
10.4.3 wait ()、notify () 和 notifyAll () 方法	249	12.4 JDBC 高级操作	284
10.4.4 饥饿和死锁	251	12.4.1 PreparedStatement	284
10.5 线程组	254	12.4.2 使用事务控制	286
10.5.1 创建一个线程组	254	12.4.3 存储过程	286
10.5.2 线程组的方法	256	12.5 本章小结	289
10.6 本章小结	257	12.6 上机练习	289
10.7 上机练习	257	12.7 习题	289
10.8 习题	257	第 13 章 网络编程	290
第 11 章 Java 小程序 Applet	260	13.1 网络初步	290
11.1 Applet 程序基础	260	13.1.1 四层体系模型	290
11.1.1 Applet 小程序	260	13.1.2 TCP 协议和 UDP 协议	291
11.1.2 Applet 程序的生命周期	263	13.1.3 IP 协议	292
11.2 绘制图形	264	13.1.4 域名	293
11.2.1 paint () 方法	264	13.1.5 套接字	293
11.2.2 使用 Graphics 类绘制字符串	265	13.2 流套接字 (Stream Socket)	294
11.2.3 Graphics 类的其他绘图方法	265	13.2.1 InetAddress 类	294
11.3 Applet 中的事件处理	267	13.2.2 Socket 类	296
11.3.1 键盘事件处理	269	13.2.3 客户端例子	297
		13.2.4 ServerSocket 类	298

13.2.5 服务器例子	299
13.2.6 能处理多个客户端的服务器	301
13.3 数据报套接字 (Datagram Socket)	304
13.3.1 DatagramPacket 类	304
13.3.2 DatagramSocket 类	305
13.3.3 多播技术和 MulticastSocket 类	307
13.4 NIO 在网络编程中的应用	310
13.4.1 NIO 回顾	311
13.4.2 套接字通道	311
13.4.3 数据报通道	314
13.5 本章小节	315
13.6 上机练习	316
13.7 习题	316
第 14 章 Servlet 编程基础	317
14.1 CGI 技术简介	317
14.2 Servlet 基础	317
14.2.1 Servlet 的生命周期	319
14.2.2 Servlet 的方法	319
14.3 Servlet 信息交互	320
14.3.1 请求信息	320
14.3.2 表单	321
14.3.3 Session	322
14.3.4 Cookie	323
14.4 JSP 简介	324
14.4.1 一个 JSP 例子	324
14.4.2 JSP 语法提要	325
14.4.3 JSP 的内置对象	333
14.5 本章小结	333
14.6 上机练习	334
14.7 习题	334
第 15 章 J2EE 技术简介	335
15.1 J2EE 技术概述	335
15.2 J2EE 的重要概念	336
15.2.1 J2EE 的四层结构	336
15.2.2 容器	337
15.2.3 J2EE 的核心 API 与组件	339

15.2.4 Enterprise Beans 的分类	340
15.3 J2EE 的开发流程	341
15.3.1 J2EE 应用程序的内容	342
15.3.2 J2EE 应用程序的开发阶段	342
15.4 J2EE 的开发环境	343
15.4.1 J2SDKEE 的安装和配置	343
15.4.2 deploytool 工具	343
15.5 J2EE 开发示例	344
15.5.1 创建 J2EE 应用程序	344
15.5.2 创建 EJB	344
15.5.3 创建 J2EE 应用程序客户端	346
15.5.4 创建 Web 客户端	349
15.5.5 部署和运行	351
15.6 本章小结	352
15.7 上机练习	353
15.8 习题	353
第 16 章 J2ME 技术简介	354
16.1 Java2 Micro Edition 平台简介	354
16.1.1 J2ME 的定义	354
16.1.2 配置 (Configuration) 和 简表 (Profile)	354
16.2 CLDC 和 MIDP 的 API	357
16.2.1 CLDC 的 API	357
16.2.2 MIDP 的 API	357
16.3 MIDP 程序设计	358
16.3.1 MIDlet	358
16.3.2 MIDlet 程序结构	359
16.3.3 开发环境	360
16.3.4 编译、打包和部署	361
16.4 MIDP 程序图形界面元件	364
16.4.1 MIDP 图形界面元件的层次	364
16.4.2 高级 API	365
16.4.3 低级 API	375
16.5 本章小结	378
16.6 上机练习	378
16.7 习题	379

第 1 章 Java 简介

本章内容提要

☆ Java 语言的历史

☆ Java 语言的特点

课前导读

学习一种程序设计语言，应当了解这种语言的诞生背景、发展情况和前景，还应当了解这种语言相对于其他程序设计语言来说有什么优点和缺点。本章主要介绍了 Java 语言的历史，还介绍了 Java 语言在诸多方面的特点。

1.1 Java 语言的历史

Java 的出现最早大概可追溯至 1991 年 4 月，Sun 的绿色计划（Green Project）开始着手于发展消费性电子产品（Consumer Electronics），该小组的成员有：Patrick Naughton, James Gosling 和 Mike Sheridan，设计小组本来是以 C++ 语言作为设计软件的程序语言，但是他们随后发现 C 或 C++ 语言是不够的。所以刚开始的时候他们试着扩展 C++ 的功能，但遇到一些编译器技术上的难题，便决定另起炉灶，于是在经过一番通力合作、存菁去芜下，一种崭新的自行发展的新程序语言—Oak 就出现了，这个程序语言早期的发展并不顺利，尤其在争取电视机上的计算机控制器项目失败后，发觉消费性电子产品的发展无法如当初预期的目标，再加上网络的兴起，绿色计划也因此而改变了发展的方向，将这种计算机技术应用到 CD-ROM 和多媒体上，而 Sun 也将 Oak 改名为 Java。

Java 的名称由来，是 Sun 公司 Java 开发小组历经无数次的脑力激荡会议之后才被选择出来的。生动（Liveliness）、动画（Animation）、速度（Speed）、互动性（Interactivity）为当初选择名字时所欲表达出的特色。Java 是在无数的建议中脱颖而出，而 Java 不是由几个单词的首字所组成，而是从许多程序设计师钟爱的热腾腾、香浓咖啡中产生灵感的。从 Oak 改名到 Java 正代表一种新形态的程序语言的诞生。

在决定要以 Java 为发展方向后，Sun 的发展目标是要 Java 成为一个简单的（Simple）、面向对象的（Object Oriented）、分布式的（Distributed）、解译的（Interpreted）、健壮的（Robust）、安全性的（Secure）、架构中立性的（Architecture Neutral）、可移植性的（Portable）、高效能的（High Performance）、多线程（Multithreaded）、动态性的（Dynamic）程序语言，也就我们现在所见到的 Java 程序语言。而因为 Java 是第一种能在网络（Internet）上具有“硬件/软件中立性”互动能力的程序语言，Java 一经推出即引起世界各种媒体的注意与热烈讨论。



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的版本也不断更新到v1.1、v1.2、v1.3、v1.4,最新的J2SDK版本是1.4,而1.5版本呼之即出,其内容也有了巨大的改进和扩充。随着Java技术的不断发展,Java还分化出三个不同的版本,以满足不同的需求:为针对企业网应用的J2EE(Java 2 Enterprise Edition)、针对普通PC应用的J2SE(Java 2 Standard Edition)和针对嵌入式设备及消费类电器的J2ME(Java 2 Micro Edition)。另外,迅速发展的还有JavaBean,其他的Java编译器和集成开发环境等第三方软件。

1.2 Java语言的特点

Java语言具有如下特点:

1. 简单性

这里所说的编程语言简单性有两层含义:一是语言本身的组成成份较少,结构较小;二是与已有语言类似,用户容易熟悉掌握。用过C或C++语言的开发人员会发现,Java使用了许多与C和C++同样的语言结构。但是又把C和C++中一般程序员很少使用的、容易出错的特征去掉了。Java略去了运算符重载、多重继承等模糊的概念。不支持goto语句,代之以提供break和continue语句以及异常处理。Java不使用主文件,它免去了预处理程序。因为Java是面向对象的,所以C中的结构struct和union也已被删去。Java还通过自动无用单元(垃圾)收集,大大简化了程序员的内存管理工作,使你不必为存储管理问题烦恼。

也许最重要的简化是Java不使用指针。指针在C和C++语言的编程中是最容易出错的部分。Java没有结构,数组和串都是对象,所以不需要指针。从而把程序员从“指针悬空”、无效指针引用和存储漏泄(leak)这些困扰中解脱出来。

另外,Java也适合于在小型机上运行,它的基本解释器及类的支持只有几十KB,加上标准类库和线程的支持也只不过几百KB。

2. 面向对象

Java是一个面向对象的语言,对程序员来说,这意味着要注意应用中的数据和操纵数据的方法,而不是严格地用过程来思考。与C++语言不一样,Java是从基础就设计成面向对象的。Java中的绝大多数成分都是对象,只有简单数、字符和布尔类型是例外。虽然Java看起来设计得很像C++,但是你会发现Java移去了C++的许多复杂性。如果你是一位C++程序员,就需要细心地研究Java中面向对象的结构。虽然Java的语法常常类似于C++,但行为远不是那样近似。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,而类则提供了对象的原型,通过继承机制,子类可以使用父类所提供的方法,这样就实现了代码的复用。



3. 分布性

Java 是面向网络的语言、Internet 上的世界通用语言。Java 从设计上就支持网络应用，它是分布式语言，Java 接口和 Java 类是以包（package）的形式分组存放。Java 语言中专门有一个支持网络的包 `Java.net`，通过这个包中的类，支持各种层次的网络连接。例如，可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他远程对象，就如同存取一个本地（local）文件系统一样简单。Java 也以 `Socket` 类支持可靠的流（Stream）网络连接，所以你可以产生分布式的客户机和服务器，这比原来的客户机/服务器方式进了一步。

4. 编译和解释性

Java 编译程序把 Java 源程序译成字节码，而不是通常的汇编码或机器码，这是 Java 跨平台特点的关键所在。为了实际运行 Java 程序，你还要使用解释程序来执行编译出来的字节码，所以在这种意义上说 Java 是一个解释性语言。实际上 Java 的前一阶段是编译，后一阶段是解释。

Java 字节码提供对机器体系结构中性的目标文件格式，代码设计成可有效地传送程序到多个平台。Java 程序可以在任何实现了 Java 解释程序和运行环境的系统上运行。这个解释程序和运行系统实现了一个称为 Java 虚拟机的虚拟机，简称 JVM。

5. 健壮性

Java 原来是用作编写消费类家用电子产品软件的语言，所以它是被设计成编写高可靠和稳健软件的。

Java 是一个强类型语言。它具有扩展编译时检查潜在的类型不匹配问题的功能。Java 要求显式的方法声明，它不支持 C 风格的隐式声明。这些严格的要求保证了编译程序能捕捉到调用错误，从而保证了程序的可靠性。

可靠性方面最重要的增强之一是 Java 的存储模型。Java 不支持指针，它消除了重写存储和讹误数据的可能性。类似地，Java 自动的无用单元收集功能可预防存储漏泄和其他有关动态存储分配和解除（释放）分配的错误。Java 解释程序也执行许多运行时的检查，诸如验证所有数组和串访问是否在界限之内。

异常（Exception）是某种类类似于错误的异常条件出现的信号。使用 `try-catch-finally` 语句，你可以在一个地方组成你的出错处理代码，一般来说，这就简化了出错处理和恢复的任务。

6. 安全性

尽管仍有用户不放心 Java 的安全性，但它在安全方面还是受到普遍的好评。当 Java 用在网络时，安全性更成为大家关心的焦点。如果得不到安全保证，你当然就不敢从 Internet 上随便什么地方下载小应用程序（Applet），并让它在你的计算机上运行了。

Java 实现了几种安全机制，保护和抵御那些试图产生病毒或损害计算机文件系统的代码。



首先, Java 利用存储分配模型来防御恶意代码。其次, Java 没有指针, 所以程序员不能得到隐蔽起来的内幕和伪造指针去指向存储器。更重要的是 Java 编译程序不处理存储安排决策, 所以程序员不能通过查看声明去猜测类的实际存储安排。编译的 Java 代码中的存储引用在运行时由 Java 解释程序决定实际存储地址。

Java 中还有许多防御措施来确保你免受不可信的小应用程序的侵犯。Java 运行系统使用字节验证过程来保证装载到网络上的代码不违背任何 Java 语言限制。这个安全机制包括类如何从网上装载, 例如, 装载的类是放在分开的名字空间而不是局部类, 预防恶意的小应用程序用它自己的版本来代替标准 Java 类。当然, 安全性并不是一件像在白纸上写字那样简单的事。没有人能保证 Java 程序不包含任何形式的恶意代码。Sun 公司所能保证的是, Java 使得写诈骗软件更加困难。Java 预先考虑和防御了历史上曾用于产生不正当行为软件的大多数技术。要说 Java 或任何系统是绝对安全是不可能的。

下面再介绍一些有关 Java 安全性的情况:

编译的 Java 字节码并不包含绝对对内存引用或地址。Java 虚拟机能处理内存的分配和引用。

为了安全性, 对字节码有验证器可起保护作用。代码首次运行前, 字节码验证器检查字节码中的不匹配参数、超越访问权限等问题。

为了防止无效字节码程序造成的危害, 如删去所有文件、搞乱硬盘或未经用户许可在网上发送本地信息等, Java 虚拟机有用户可配置的访问权限, 如拒绝本地写入、拒绝向其他站点和其他主机和网络访问等。

大多数安全性措施都依靠无病毒的 Java 虚拟机, 所以 Java 虚拟机的取得要慎之又慎。下载一个 Java 虚拟机环境时则更要小心。Sun、IBM 等公司的 Java 虚拟机都经过了彻底检查, 下载和操作通常是安全的。

7. 中性的体系结构

Java 编译程序生成与体系结构无关的字节码, 这种字节码适合各种体系结构。这种方法的优点是允许 Java 应用能在任何系统上运行, 只要该系统实现 Java 虚拟机。用目前 Java 编写的代码经编译后的代码文件可以在 Solaris、Windows、UNIX、Linux 和 MacOS 等不同的操作系统上运行。

8. 可移植性

体系结构中性是可移植性的一大部分。但是 Java 有更进一步的使得语言声明不依赖于实现的方面。例如, Java 显式说明每个基本数据类型的大小和它的运算行为 (这些数据类型由 Java 语法描述)。

Java 环境本身对新的硬件平台和操作系统也是可移植的。Java 编译程序也用 Java 编写, 而 Java 运行系统是用 C 语言编写的。它本质上是遵从 Posix (一种 Unix 标准) 的。Java 的类库中也实现了与不同平台的接口, 使这些类库可以移植。

9. 高性能

Java 是一种先编译后解释的语言, 但和其他纯解释性语言不同, Java 字节码的设计



使之能很容易地直接转换成对应于特定 CPU 的机器码或汇编码,当然这不如全编译性语言快。事实上,Java 平均比 C 语言写的程序执行起来慢 20 倍。不过你要认识这个速度对交互式运行、图形用户界面和基于网络的应用来讲是适当的,因为在这些应用中,常常是有空闲的,等待用户做某些事,或者等待来自网络上的程序或数据。

当然,有些情况下性能是很要紧的。为了支持这些情况,Jit (Just-in-time) 编译程序能在运行时把 Java 字节码翻译成特定 CPU 的机器代码。Java 字节码格式设计时考虑到这些 JIT 编译程序的需要,所以生成机器代码的过程相当简单,它能产生较好的代码。Sun 公司声称字节码转换到机器码的性能和本地机器的 C 或 C++ 的几乎一样。

下面再介绍一些性能降低的原因,分析原因是为了解决问题、提高性能,国外一些公司已在做大量工作。

首先是在主机操作系统上加了虚拟机层,比本地编译码慢 10~20 倍。读入类要在运行时验证。执行类时,Java 字节码还要转换成本地机器指令,多次执行就要多次转换,还有安全检查,如访问数组元素时,Java 虚拟机要确保没有越界访问。

Java 指令都是字节码,但大多数操作的对象超过一个字节长,所以必须读多个字节码来取得不同操作符及其参数,这些也将影响性能。

此外,Java 采用堆栈机理,即指令从操作数堆栈取参数并将结果放回堆栈。而通常编译程序用非常快的寄存器操作,而堆栈在主存内,寄存器操作比堆栈操作要快好几倍。另外,传统编译程序进行多种优化工作。

最后,Java 系统要进行自动内存无用单元的回收,这无疑也会影响性能。

10. 多线程性

Java 程序通过流控制来执行程序流,程序中单个顺序的流控制称为线程,多线程则指的是在单个程序中可以同时运行多个不同的线程执行不同的任务。

多线程程序设计大大提高了程序执行效率和处理能力。对程序员来说,用 C 或 C++ 语言编写处理多线程的代码可能是很麻烦的事。Java 语言使具有线索的程序设计很容易,Java 对线程提供内部语言支持,Java 中的 Thread 类和 Runnable 接口使程序员很容易开发出多线程的应用程序。

如果计算机只有一个 CPU,在给定时刻只有一个执行线程。CPU 迅速地在几个线程之间切换,从而造成了多个线程同时执行的假象。所以说单处理器系统是支持逻辑并发,而不是物理并发。在多处理器系统上几个线程真正同时执行,实现物理并发。多线程程序的重要特征是不管是否真正实现物理并发,多线程程序都支持逻辑并发。

1.3 本章小结

本章主要介绍了 Java 语言的历史和 Java 语言的特点。Java 语言的前身是 Oak,这种语言一开始是作为消费电子产品的开发语言提出的,而 Java 语言却是因网络而发展起来的。

Java 具有简单、面向对象、分布式、解译性、健壮性、安全性、架构中立性、可移



植性、高效能、多线程、动态性等诸多优秀的特性，这使得 Java 语言在诞生之日起就受到人们的广泛关注，并随着技术的进步获得了巨大的发展。Java 目前分化成三个版本：J2SE、J2EE 和 J2ME，而 Java 2SDK 的版本也已经发展到了 1.5 版。

1.4 习 题

- (1) 简述 Java 语言的起源和发展历史。
- (2) 简述 Java 语言的特点，说明 Java 语言的优缺点。
- (3) 为什么说 Java 语言是平台无关的？

第2章 Java 编程环境

本章内容提要

- ☆ Java 运行环境和开发环境
- ☆ 开发环境的安装和使用
- ☆ 编辑器选择
- ☆ Java 程序的结构
- ☆ 程序调试

课前导读

本章重点在于 Java 虚拟机的概念和 J2SDK 的使用。本章首先介绍 Java 的运行环境和开发环境，然后详细介绍 Sun 公司的 J2SDK 开发工具，并且给出两个 Java 程序的源代码以及编译、运行过程，以加深读者对 Java 语言程序开发的直观认识，最后本章介绍 J2SDK 工具包提供的程序调试器。

2.1 Java 运行环境和开发环境

正如前文所言，Java 语言是先编译后解释的。Java 源程序首先被编译成字节码文件，然后由 Java 解释程序解释性地执行。这个 Java 解释程序即 Java 虚拟机，Java 虚拟机在各种常用的操作系统中都已经安装。

Java 虚拟机是一个虚拟机器，在实际的计算机上通过软件模拟来实现。Java 虚拟机有自己虚拟的硬件，如处理器、堆栈、寄存器等，还具有相应的指令系统。

Java 语言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 虚拟机后，Java 语言在不同平台上运行并不需要重新编译。Java 语言使用 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 Java 语言底层实现的基础，对 Java 语言感兴趣的人都应对 Java 虚拟机有个大概的了解。

Java 最基本的开发工具是 Sun 公司发布的 Java 开发工具 Java2SDK。Java2SDK 是免费软件，可以通过 <http://Java.sun.com> 网站或其他网站取得 J2SDK。另外还有一些其他公司也实现了 Java 的开发工具，如 IBM 的 JDK 等。大型的集成开发环境软件有 Borland JBuilder 等，这些软件提供了完整的 Java 开发环境和可视化的开发手段。

Java 源程序可以用任何文本编辑器来编写。然后用 Java2SDK 的 Javac 编译生成扩展