

● Practical...

Java

程序设计教程

张 曜 张 青 郭立山 编著

Programming Series



冶金工业出版社

Java 程序设计教程

张曜 张青 郭立山 编著

北 京

冶金工业出版社

2002

内 容 简 介

本书介绍了 Java 语言的高级特性及相关的编程技巧,其内容主要包括:Java 语言与面向对象编程基础、Java Applet、字符串、异常处理、线程、数据结构与 Java 工具、高级 Swing、文件与流、图形、JavaBeans、数据库、网络编程、远程方法调用、Servlet、Java 与 XML 等。在全书的编写过程中,采用理论讲解与实例编程相结合的方法,对于每个具体的内容都有相应的实例讲解。

本书结构合理,语言简练,内容深入浅出,主要面向具有一定 Java 编程基础的人员。对于 Java 编程的初、中级读者,本书将会帮助你快速进入 Java 编程高手的行列。本书也可以作为高等院校和培训班相关专业的教材。同时,本书对高级 Java 编程人员也有较高的参考价值。

图书在版编目(CIP)数据

Java 程序设计教程 / 张曜等编著. —北京:冶金工业出版社, 2002.11
ISBN 7-5024-3111-X

I. J... II. 张... III. JAVA 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 075132 号

出版人 曹胜利(北京沙滩嵩祝院北巷 39 号,邮编 100009)

责任编辑 戈兰

中山市新华印刷厂有限公司印刷;冶金工业出版社发行;各地新华书店经销

2002 年 12 月第 1 版, 2002 年 12 月第 1 次印刷

787mm×1092mm 1/16; 34.5 印张; 799 千字; 539 页; 1-2500 册

49.00 元

冶金工业出版社发行部 电话:(010) 64044283 传真:(010) 64027893

冶金书店 地址:北京东四西大街 46 号(100711) 电话:(010) 65289081

(本社图书如有印装质量问题,本社发行部负责退换)

前 言

1. 关于 Java

Java 是面向对象的、支持多线程的、体系结构独立的解释型动态语言，即具有高度的安全性、可移植性和代码可重用性，又具有很高的效率。Java 是目前软件中极为健壮的编程语言，自问世以来，就以其得天独厚的优势，在 IT 业界掀起了研究、开发它的浪潮，显示了“鹏程万里”的美好前景。作为一种与底层硬件无关的、“编写一次，到处运行”的高级语言和计算平台，Java 具有将网络上的各个平台连成一体的能力，真正实现了“网络就是计算机”的理念。以 Java 为代表的网络的成长，改变了人们的联系方式、生活方式、学习方式和经营方式。它的到来使人们目睹了一场类似印刷术的重大变革。

2. 关于本书

本书详细介绍了一个专业的 Java 开发人员所需要掌握的各种 Java 高级特性，阐述了 Java 语言高级编程的相关特性。书中包含了大量的有代表性的实例和精辟的介绍，使读者能快速领会 Java 的精髓。

目前，市面上有很多关于 Java 编程的书，但对 Java 编程技术的介绍大多浅尝辄止，很少全面介绍 Java 的高级技术，更缺少对一些高级应用的深刻剖析，因此很难对读者独立开发专业 Java 应用程序有实质性的帮助。有鉴于此，作者搜集了大量的资料，写成了这本书，来填补这一空缺。

3. 本书结构安排

全书共分为 17 章，各章内容安排如下：

第 1 章：主要介绍了 Java 的起源、Java 的特点以及 Java 的开发和执行环境。

第 2 章：主要介绍了 Java 语言基础。

第 3 章：主要介绍了面向对象编程基础。

第 4 章：主要介绍了 Java Applet 的使用。

第 5 章：主要介绍了 Java 中与字符串相关的 String 类、StringBuffer 类和 String Tokenizer 类。

第 6 章：主要介绍了 Java 的异常处理机制。

第 7 章：主要介绍了线程的基本概念和在 Java 中如何使用多线程。

第 8 章：主要介绍了数据结构以及 Java 所提供的工具。

第 9 章：主要介绍了 Swing 中的主要组件。

第 10 章：主要介绍了 Java 的文件操作以及对象序列化。

第 11 章：主要介绍了图形类（Graphics 类）以及与图形有关的一些实用类。

第 12 章：主要介绍了 JavaBeans 的基本概念以及如何开发自己的 Bean。

第 13 章：主要介绍了 JDBC，即 Java 数据库连接 API。

第 14 章：主要介绍了 Java 的基本网络功能。

第 15 章：主要介绍了远程对象以及远程方法调用的相关知识。

第 16 章：主要介绍了使用 Servlet 开发服务器端程序。

第 17 章：主要介绍了 XML 的基础以及 SAX、DOM 编程。

此外，本书的后面还提供了介绍 JDBC API 的附录和每章习题的参考答案。

4. 本书特点

在介绍 Java 的各种技术时，本书采取“知识点各个击破，整体应用实例剖析”的策略。在内容的介绍上循序渐进，并力求深入浅出，尽量以最易于理解和接受的方式把有关知识呈现给读者。在编写过程中凡是涉及到具体的 Java 编程知识，都会给出相应的实例，并对例子作必要的剖析，力求使读者从实际的例子中领会到相关的知识及编程技巧。对应于不同的具体内容采用了灵活的编写方法，使表达更加简洁明了。

为便于读者阅读，本书在每章的开头都将该章将介绍到的类以表格的形式列出，这也大致反映了该章的主要内容。在介绍的过程中，对于一些作者认为有必要进一步加以说明的地方，会用粗体字“**说明：**”加以标记，提醒读者加以注意。此外，在本书编写的过程中尽量使各章之间内容保持独立，因此读者不一定要按章节的顺序阅读本书，可以按自己需要的次序来阅读各章的内容。

为帮助读者巩固所学的知识，本书每章的结尾都对本章的内容进行了小结，还给出了一定数量的针对该章重点内容的习题，读者可以对照书后的参考答案进行练习。

5. 适用对象

本书主要面向具有一定 Java 编程基础的人员。对于 Java 编程的初、中级读者，本书将会帮助你快速进入 Java 编程高手的行列。本书也可以作为高等院校和培训班相关专业的教材。对高级 Java 编程人员来说，本书也有较高的参考价值。由于在本书编写的过程中尽量降低各章之间内容的联系性，因此，只要是具有某一方面的 Java 编程经历的读者都可以掌握本书的相关内容。

本书由张曜、张青、郭立山编著。此外，余柏山参与编写了第 1、3、5、6、7、10、11、12 章，温瑞星参与编写了第 2、4、8、9、13、14、15、16、17 章，全书由郭立山审校。

本书的附送资料可以在网上下载，**网址：**<http://www.cnbook.net>。此外，读者在学习此书的过程中，如果遇到疑难问题或不妥之处，也可在该网站的相关论坛进行探讨。

虽然作者在编写本书的过程中，对本书进行过反复的审核，但是由于水平有限，加上时间仓促，错漏之处在所难免，希望读者给予批评指正，多提宝贵意见。

编 者

2002 年 10 月

目 录

第1章 Java 概述 1

1.1 Java 的起源	1
1.2 Java 的特点	2
1.3 Java 和 C、C++ 的比较	3
1.4 Java 的开发工具和执行环境	5
1.4.1 Applet Viewer 小应用程序 浏览器	6
1.4.2 Java 解释器	7
1.4.3 Javac 编译器	8
1.4.4 Javah 头文件生成器	8
1.4.5 Javap 反汇编器	9
1.4.6 Java 调试器 jdb	9
1.4.7 Javadoc API 文件产生器	10
1.5 使用 JBuilder 6.0 集成开发环境	11
1.5.1 JBuilder 6.0 的主界面介绍	11
1.5.2 JBuilder 6.0 的使用	13
小结	16
综合练习题一	16
一、选择题（多选）	16
二、填空题	17
三、问答题	17
四、上机题	17

第2章 Java 语言基础 18

2.1 Java 语言的组成元素	18
2.1.1 标识符	18
2.1.2 保留字	18
2.1.3 数据	19
2.1.4 运算符	19
2.1.5 分隔符	19
2.2 变量和常量及数据类型	19
2.2.1 变量和常量	19
2.2.2 数据类型	20
2.3 运算符	21
2.3.1 算术运算符	21

2.3.2 关系运算符和布尔运算符	22
2.3.3 赋值运算符	22
2.4 流程控制	23
2.4.1 分支语句	23
2.4.2 循环语句	24
2.4.3 跳转语句	26
小结	27
综合练习题二	27
一、选择题	27
二、填空题	28
三、问答题	28
四、上机题	28

第3章 面向对象编程基础 29

3.1 面向对象编程的思维方式	29
3.2 面向对象的基本概念和特征	29
3.2.1 对象	30
3.2.2 消息和方法	30
3.2.3 类	31
3.2.4 面向对象的基本特征	32
3.3 面向对象的程序设计方法	33
小结	35
综合练习题三	35
一、选择题	35
二、填空题	36
三、问答题	36
四、上机题	36

第4章 Java Applet 37

4.1 Applet 简介	37
4.1.1 什么是 Applet	37
4.1.2 Applet 的安全机制	37
4.1.3 第一个 Applet 程序	38
4.2 Applet 的创建和执行	38
4.2.1 Applet 的创建	39
4.2.2 Applet 类的继承关系	39

4.2.3 Applet 的执行	39	5.4.1 创建 StringTokenizer 类对象	68
4.2.4 Applet 的生命周期及主要方法	40	5.4.2 使用 StringTokenizer 对象	70
4.2.5 Applet 和 Application	42	小结	72
4.3 Applet 的通信	43	综合练习题五	72
4.3.1 同页 Applet 间的通信	43	一、选择题 (多选)	72
4.3.2 Applet 和浏览器之间的通信	45	二、填空题	73
4.3.3 Applet 的网络通信	46	三、问答题	73
小结	46	四、上机题	73
综合练习题四	47	第 6 章 异常处理	74
一、选择题	47	6.1 异常概述	74
二、填空题	48	6.2 异常分类	75
三、问答题	48	6.3 try 和 catch 语句	75
四、上机题	48	6.4 使用多个 catch 子句	76
第 5 章 字符串	49	6.5 try 语句的嵌套	77
5.1 字符串概述	49	6.6 throw 语句	78
5.2 String 类	49	6.7 throws 语句	79
5.2.1 创建字符常量 String 对象	50	6.8 finally 语句	80
5.2.2 获取字符串长度	52	6.9 创建自己的异常	81
5.2.3 字符串的比较	52	小结	82
5.2.4 字符串中单个字符的查找	53	综合练习题六	82
5.2.5 字符串中子串的查找	55	一、选择题	82
5.2.6 字符串的前缀与后缀的判断	56	二、填空题	83
5.2.7 提取部分字符串	56	三、问答题	83
5.2.8 连接字符子串	58	四、上机题	83
5.2.9 修改字符串	58	第 7 章 线程	84
5.2.10 利用 valueOf() 方法实现		7.1 线程概述	84
数据转换	59	7.2 使用线程的好处	85
5.3 StringBuffer 类	61	7.3 线程的属性	85
5.3.1 创建字符串变量——		7.4 使类线程化的两种途径	87
StringBuffer 对象	61	7.4.1 继承 Thread 类	88
5.3.2 获取 StringBuffer 对象的		7.4.2 实现 Runnable 接口	92
长度与容量信息	61	7.5 改变线程运行状态	94
5.3.3 扩充 StringBuffer 对象的字符串 ..	62	7.5.1 休眠和“同步”访问	94
5.3.4 获取 StringBuffer 对象中的		7.5.2 暂停和恢复	97
字符串	64	7.5.3 等待和通知	99
5.3.5 修改 StringBuffer 对象	65	7.5.4 IO 阻塞	103
5.4 StringTokenizer 类	68		

7.6 线程的优先级	105	二、填空题	157
7.7 线程中断	109	三、问答题	157
7.8 线程同步	110	四、上机题	157
7.8.1 同步访问共享资源	110		
7.8.2 死锁问题	115		
7.9 用户线程与超级线程	118		
小结	120		
综合练习题七	120		
一、选择题（多选）	120		
二、填空题	122		
三、问答题	122		
四、上机题	122		
第8章 数据结构与Java工具	123	第9章 高级Swing	158
8.1 数据结构概述	123	9.1 Swing 概述	158
8.2 集成	123	9.2 布局管理器	159
8.2.1 集成接口	124	9.2.1 BorderLayout	160
8.2.2 List 接口	124	9.2.2 FlowLayout	161
8.2.3 Map 接口	125	9.2.3 GridLayout	162
8.2.4 Iterator 接口	125	9.2.4 GridBagLayout	164
8.2.5 ListIterator 接口	126	9.3 菜单	164
8.3 Vector 类	126	9.3.1 创建菜单	166
8.4 ArrayList 类	130	9.3.2 指定快捷键	167
8.5 Hashtable 类	130	9.3.3 两种特殊的菜单项	167
8.6 Properties 类	132	9.4 工具条	167
8.7 Stack 类	134	9.5 列表框	170
8.8 Random 类	135	9.5.1 实现列表的滚动功能	174
8.9 Date 类	136	9.5.2 列表模型	174
8.10 综合实例——通信录	138	9.6 组合框	175
8.10.1 设计思路	139	9.7 JEditorPane	177
8.10.2 创建项目并设计主界面	139	9.7.1 超链接事件	179
8.10.3 读写文件数据	140	9.7.2 设置新页面	179
8.10.4 实现添加功能	142	9.8 树	179
8.10.5 实现排序功能	143	9.8.1 创建节点	183
8.10.6 完整代码清单	145	9.8.2 树模型	183
小结	156	9.9 表格	184
综合练习题八	156	9.9.1 表格模型	187
一、选择题	156	9.9.2 单元渲染和编辑	188
		9.10 综合例子——一个IE风格的	
		浏览器	189
		9.10.1 设计思路	189
		9.10.2 设计主界面	189
		9.10.3 实现浏览器功能	191
		9.10.4 收藏夹的实现	192
		9.10.5 保存访问过的页面	195
		9.10.6 完整代码清单	196
		小结	208

综合练习题九	209	11.1 paint()、repaint()、update()的 区别	256
一、选择题	209	11.2 图形坐标系统	257
二、填空题	209	11.3 Graphics 类的使用	257
三、问答题	209	11.3.1 画线	257
四、上机题	210	11.3.2 画矩形	258
第 10 章 文件与流	211	11.3.3 画三维矩形	259
10.1 流概述	211	11.3.4 画圆角矩形	260
10.2 基本输入输出流	212	11.3.5 画圆和椭圆	261
10.2.1 InputStream 类	212	11.3.6 画圆弧	262
10.2.2 OutputStream 类	212	11.3.7 画多边形	263
10.2.3 PrintWriter 类	212	11.3.8 画文本	264
10.2.4 System.in 对象和 System.out 对象	213	11.4 其他与图形相关的实用类	266
10.3 文件处理	214	11.4.1 Point 类	266
10.3.1 FileInputStream 类	214	11.4.2 Dimension 类	267
10.3.2 FileOutputStream 类	216	11.4.3 Rectangle 类	269
10.3.3 File 类	217	11.4.4 Polygon 类	272
10.3.4 RandomAccessFile 类	219	11.4.5 Font 类	274
10.4 对象流	223	11.4.6 FontMetrics 类	277
10.4.1 对象序列化	223	11.4.7 Color 类	279
10.4.2 ObjectOutputStream 类	224	11.5 综合实例——一个简单的 画图程序	282
10.4.3 ObjectInputStream 类	225	11.5.1 设计思路	282
10.5 综合实例——文件管理器	227	11.5.2 设计主界面	282
10.5.1 设计思路	227	11.5.3 实现菜单事件的功能	283
10.5.2 设计界面	227	11.5.4 实现具体的画图功能	284
10.5.3 文件夹	229	11.5.5 完整代码清单	286
10.5.4 实现新建删除修改功能	231	小结	293
10.5.5 实现拷贝功能	234	综合练习题十一	293
10.5.6 完整代码清单	236	一、选择题	293
小结	254	二、填空题	295
综合练习题十	254	三、问答题	295
一、选择题	254	四、上机题	295
二、填空题	255	第 12 章 JavaBeans	296
三、问答题	255	12.1 组件模型基础	296
四、上机题	255	12.1.1 组件属性	297
第 11 章 图形	256	12.1.2 组件方法	297

12.1.3 组件事件	297
12.2 JavaBeans	298
12.3 Bean 的基本结构	298
12.4 Bean 开发工具包	299
12.4.1 安装 BDK	299
12.4.2 使用 BDK	299
12.4.3 在 BeanBox 中创建一个 简单应用程序	300
12.4.4 保存和恢复 BeanBox 的状态 ...	302
12.4.5 从 BeanBox 创建一个 Applet ...	302
12.5 JAR 文件	303
12.5.1 清单文件	303
12.5.2 JAR 工具	304
12.6 Bean 属性和事件的命名模式	305
12.7 开发一个简单的 Bean	305
12.7.1 创建 Colors Bean	306
12.7.2 创建一个 Colors Bean 的 应用程序	309
12.8 Bean 的属性类型	309
12.8.1 简单属性	309
12.8.2 索引属性	309
12.8.3 关联属性	310
12.8.4 限制属性	311
12.9 Bean 事件	312
12.10 使用 BeanInfo 接口	314
12.11 Enterprise JavaBeans	315
12.11.1 JavaBeans 和 Enterprise JavaBeans	316
12.11.2 远程接口	316
12.11.3 Enterprise JavaBeans 体系结构	317
12.11.4 Enterprise JavaBeans 的 开发过程	318
小结	319
综合练习题十二	320
一、选择题 (多选)	319
二、填空题	319
三、问答题	320

四、上机题	320
第 13 章 数据库	321
13.1 JDBC 概述	321
13.2 结构化查询语言	322
13.3 准备工作	324
13.4 JDBC 的连接	325
13.4.1 装入 JDBC 驱动程序	326
13.4.2 JDBC URL	326
13.4.3 建立连接	327
13.4.4 创建语句	327
13.5 数据定义与更新	328
13.6 数据查询	331
13.7 其他类型的语句	333
13.7.1 PreparedStatement	334
13.7.2 CallableStatement	336
13.8 综合实例——人事管理系统	336
13.8.1 设计思路	337
13.8.2 设计界面	337
13.8.3 记录的添加和修改	338
13.8.4 搜索功能的实现	341
13.8.5 完整代码清单	343
小结	357
综合练习题十三	357
一、选择题	357
二、填空题	358
三、问答题	359
四、上机题	359
第 14 章 网络编程	360
14.1 网络基本知识	360
14.1.1 TCP/IP 协议	360
14.1.2 套接字	362
14.2 URL 编程	363
14.2.1 URL 类	363
14.2.2 URLConnection 类	365
14.3 TCP 套接字编程	367
14.3.1 InetAddress 类	367

14.3.2 Socket 类.....	368	15.8.9 创建主干并启动注册程序.....	431
14.3.3 ServerSocket 类.....	370	15.8.10 完整代码清单.....	432
14.4 UDP 套接字编程.....	372	小结.....	450
14.4.1 DatagramPacket 类.....	372	综合练习题十五.....	451
14.4.2 DatagramSocket 类.....	373	一、选择题.....	451
14.5 综合实例——类似 FlashGet 的 下载工具.....	376	二、填空题.....	451
14.5.1 设计思路.....	376	三、问答题.....	451
14.5.2 设计界面.....	376	四、上机题.....	451
14.5.3 程序的组织.....	377	第 16 章 Servlet.....	452
14.5.4 HTTP 协议简介.....	379	16.1 Servlet 概述.....	452
14.5.5 下载功能的实现.....	380	16.1.1 什么是 Servlet.....	452
14.5.6 保存未完成的任务的信息.....	389	16.1.2 Servlet 与 CGI.....	453
14.5.7 完整代码清单.....	391	16.1.3 开发工具和环境.....	453
小结.....	413	16.2 Servlet 程序的结构.....	455
综合练习题十四.....	413	16.2.1 HttpServlet 类.....	456
一、选择题.....	413	16.2.2 HttpServletRequest 接口.....	456
二、填空题.....	414	16.2.3 HttpServletResponse 接口.....	457
三、问答题.....	414	16.3 处理 HTTP GET 请求.....	457
四、上机题.....	414	16.4 处理 HTTP POST 请求.....	459
第 15 章 远程方法调用.....	415	16.5 会话状态.....	462
15.1 RMI 概述.....	415	16.5.1 Cookie.....	462
15.2 创建远程对象.....	416	16.5.2 HttpSession.....	466
15.3 扩展远程接口.....	417	小结.....	469
15.4 创建实现类.....	417	综合练习题十六.....	469
15.5 生成主干和框架.....	419	一、选择题.....	469
15.6 创建客户程序.....	419	二、填空题.....	470
15.7 启动注册并运行程序.....	420	三、问答题.....	470
15.8 综合实例——远程数据查询系统.....	421	四、上机题.....	470
15.8.1 设计思路.....	421	第 17 章 Java 与 XML.....	471
15.8.2 设计界面.....	421	17.1 XML 简介.....	472
15.8.3 设计数据库.....	422	17.2 XML 基础.....	475
15.8.4 创建实现了远程接口的类.....	423	17.2.1 处理指令.....	475
15.8.5 启动服务器.....	426	17.2.2 名字空间.....	475
15.8.6 登录服务器.....	427	17.2.3 标签属性.....	476
15.8.7 查询数据.....	429	17.2.4 DTD.....	477
15.8.8 创建 policy 文件.....	430	17.2.5 样式表.....	480

17.2.6 DOM	481	MetaData	504
17.2.7 SAX	482	A.2.7 接口 java.sql.Driver	507
17.2.8 JAXP	482	A.2.8 接口 java.sql.Prepared Statement	507
17.2.9 JDOM	483	A.2.9 接口 java.sql.ResultSet	508
17.3 使用 JAXP 进行 SAX 编程	483	A.2.10 接口 java.sql.ResultSet MetaData	511
17.3.1 解析文档	483	A.2.11 接口 java.sql.SQLData	512
17.3.2 内容管理器	485	A.2.12 接口 java.sql.Statement	512
17.4 使用 JAXP 进行 DOM 编程	489	A.2.13 接口 java.sql.Struct	513
17.4.1 解析文档	489	A.3 类	513
17.4.2 基本的 DOM 类	491	类 java.sql.DriverManager	513
17.4.3 存取数据	493	参考答案	514
17.4.4 修改文件	495	第 1 章	514
小结	499	第 2 章	514
综合练习题十七	499	第 3 章	515
一、选择题	499	第 4 章	516
二、填空题	499	第 5 章	518
三、问答题	500	第 6 章	520
四、上机题	500	第 7 章	521
附录 JDBC API 介绍	501	第 8 章	523
A.1 包 java.sql	501	第 9 章	524
A.1.1 接口	501	第 10 章	526
A.1.2 类	501	第 11 章	528
A.1.3 异常	501	第 12 章	531
A.2 接口	502	第 13 章	535
A.2.1 接口 java.sql.Array	502	第 14 章	536
A.2.2 接口 java.sql.Blob	502	第 15 章	537
A.2.3 接口 java.sql.Callable Statement	502	第 16 章	537
A.2.4 接口 java.sql.Clob	503	第 17 章	539
A.2.5 接口 java.sql.Connection	503	参考文献	540
A.2.6 接口 java.sql.Database			

第 1 章 Java 概述

本章提要

- Java 的起源与特点
- Java 的开发工具和执行环境
- 使用 JBuilder 6.0 集成开发环境

在学习任何一门语言之前,都应该先了解该门语言的背景,包括该门语言的发展历史。再就是在准备更深入地学习一门语言之前,应该对该语言的特点有一个比较全面正确的理解,这样才能使学习更有效。

本章主要介绍 Java 语言的起源,Java 语言的特点,以及 Java 语言的开发工具和执行环境。

1.1 Java 的起源

1991 年,SUN MicroSystem 公司的 Jame Gosling、Bill Joe 等人,为在电视、控制烤面包箱等家用消费类电子产品上进行交互式操作而开发了一个名为 Oak 的软件(即一种橡树的名字),因与当时已有的一种软件名冲突,后来该软件改名为 Java。Java 当时并没有引起人们注意,直到 1994 年下半年,Internet 的迅猛发展,环球信息网 WWW 的快速增长,促进了 Java 语言研制的进展,使得它逐渐成为 Internet 上受欢迎的开发与编程语言,一些著名的计算机公司纷纷购买了 Java 语言的使用权。例如:Microsoft、IBM、Netscape、Novell、Apple、DEC、SGI 等。

因此,Java 语言被美国的著名杂志 PC Magazine 评为 1995 年十大优秀科技产品(计算机类就此一项入选),随之大量出现了用 Java 编写的软件产品,受到工业界的重视与好评,认为“Java 是 20 世纪 80 年代以来计算机界的一件大事”,微软总裁比尔·盖茨在悄悄地观察了一段时间后,不无感慨地说:“Java 是长时间以来最卓越的程序设计语言”,并确定微软整个软件开发战略从 PC 单机时代向网络为中心的计时代转移,而购买 Java 则是他重大战略决策的实施部署。因此 Java 的诞生必将对整个计算机产业产生深远影响,对传统计算模型提出了新的挑战。

SUN MicroSystem 公司的总裁 Scott McNealy 认为 Java 为 Internet 和 WWW 开辟了一个崭新的时代。环球信息网 WWW 的创始人 Berners-Lee 说:“计算机事业发展的下一个浪潮就是 Java,并且将很快会发生的。”

Microsoft 和 IBM 两大公司都计划在 Internet 上销售用 Java 编写的软件。Apple、HP、IBM、Microsoft、Novell、SGI、SCO、Tandem 等公司均计划将 Java 并入各自开发的操作系统,而负责开发并推广 Java 技术的 SunSoft 公司(这是 SUN 下属的一个子公司),将通过颁发许可证的办法来允许各家公司把 Java 虚拟机和 Java 的 Applets 类库嵌入他们开发的操作系统,这样各类开发人员就能更容易地选择多种平台来使用 Java 语言编程,不同的用户

也就可以脱离 Web 浏览器来运行 Java 应用程序,这无疑是很受广大用户欢迎的,也为 Java 语言的应用开拓了极为广阔的前景(当然,各类 Java OS 之间的兼容性必须得到重视,好在 JavaSoft 已保证将监督这种兼容性)。

有人预言:Java 将是网络上的“世界语”,今后所有的用其他语言编写的软件统统都要用 Java 语言来改写。

1.2 Java 的特点

Java 是一个广泛使用的网络编程语言,它是一种新的计算概念。

首先,作为一种程序设计语言,它简单、面向对象、不依赖于机器的结构、具有可移植性、安全性、并且提供了并发的机制、具有很高的性能。其次,它最大限度地利用了网络,Java 的小应用程序(applet)可在网络上传输而不受 CPU 和环境的限制。另外,Java 还提供了丰富的类库,使程序设计者可以很方便地建立自己的系统。Java 有下面一些特点。

1. 简单性

Java 语言是一种面向对象的语言,它通过提供最基本的方法来完成指定的任务,只需理解一些基本的概念,就可以用它编写出适合于各种情况的应用程序。Java 略去了运算符重载、多重继承等模糊的概念,并且通过实现自动垃圾回收,大大简化了程序设计者的内存管理工作。

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 40KB 左右,加上标准类库和线程的支持也只有 215KB 左右。

2. 面向对象

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. 分布性

Java 是面向网络的语言。通过它提供的类库可以处理 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对象。

4. 安全性

用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针,一切对内存的访问都必须通过对象的实例变量来实现,这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员,同时也避免了指针操作中容易产生的错误。

5. 体系结构中立

Java 解释器生成与体系结构无关的字节码指令,只要安装了 Java 运行时系统,Java 程序就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示,Java 解释器得到字节码后,对它进行转换,使之能够在不同的平台运行。

6. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器。同时,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。另外,Java 编译器是由 Java 语言实现的,Java 运行时系统由标准 C 实现,这使得 Java 系统本身也具有可移植性。

7. 解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译信息，使得连接过程更加简单。

8. 高性能

和其他解释执行的语言如 BASIC、TCL 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，从而得到较高的性能。

9. 多线程

多线程机制使应用程序能够并行执行，而且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，这样就很容易地实现网络上的实时交互行为。

10. 动态性

Java 的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。并且 Java 通过接口来支持多重继承，使之比严格的类继承具有更灵活的方式和扩展性。

11. Java Applet

Java 语言的特性使它可以最大限度地利用网络。Applet 是 Java 的小应用程序，它是动态、安全、跨平台的网络应用程序。Java Applet 嵌入 HTML 语言，通过主页发布到 Internet。网络用户访问服务器的 Applet 时，这些 Applet 从网络上进行载入，然后在支持 Java 的浏览器中运行。由于 Java 语言的安全机制，用户一旦载入 Applet，就可以放心地来生成多媒体的用户界面或完成复杂的计算而不必担心病毒的入侵。

虽然 Applet 可以和图像、声音、动画等一样从网络上下载，但它并不同于这些多媒体文件格式，它可以接收用户输入，动态地进行改变，而不仅仅是动画的显示和声音的播放。

12. 丰富的类库

Java 提供了大量的类以满足网络化、多线程、面向对象系统的需要。

(1) 语言包提供的支持包括字符串处理、多线程处理、例外处理、数学函数处理等，可以用它简单地实现 Java 程序的运行平台。

(2) 实用程序包提供的支持包括哈希表、堆栈、可变数组、时间和日期等。

(3) 输入输出包用统一的“流”模型来实现所有格式的 I/O，包括文件系统、网络、输入/输出设备等。

(4) 低级网络包用于实现 Socket 编程。

(5) 抽象图形用户接口包实现了不同平台的计算机的图形用户接口部分，包括窗口、菜单、滚动条、对话框等，使得 Java 可以移植到不同平台的机器。

(6) 网络包支持 Internet 的 TCP/IP 协议，提供了与 Internet 的接口。它支持 URL 连接，WWW 的即时访问，并且简化了用户/服务器模型的程序设计。

1.3 Java 和 C、C++ 的比较

对于变量声明、参数传递、操作符、流控制等，Java 使用了和 C、C++ 相同的传统，

使得熟悉 C、C++ 的程序员能很方便地进行编程。同时, Java 为了实现其简单、安全等特性, 也摒弃了 C 和 C++ 中许多不合理的内容。

1. 全局变量

Java 程序中, 不能在所有类之外定义全局变量, 只能通过在一个类中定义公用、静态的变量来实现一个全局变量。例如:

```
Class GlobalVar{  
    public static global_var;  
}
```

在类 GlobalVar 中定义变量 global_var 为 public static, 使得其他类可以访问和修改该变量。Java 对全局变量进行了更好的封装。而在 C 和 C++ 中, 依赖于不加封装的全局变量常常造成系统的崩溃。

2. Goto

Java 不支持 C、C++ 中的 goto 语句, 而是通过例外处理语句 try, Catch, final 等来代替 C、C++ 中用 goto 来处理遇到错误时跳转的情况, 使程序更可读且更结构化。

3. 指针

指针是 C、C++ 中最灵活, 也是最容易产生错误的数据类型。由指针所进行的内存地址操作常会造成不可预知的错误, 同时通过指针对某个内存地址进行显式类型转换后, 可以访问一个 C++ 中的私有成员, 从而破坏安全性, 造成系统的崩溃。

而 Java 对指针进行完全的控制, 程序员不能直接进行任何指针操作, 例如: 把整数转化为指针, 或者通过指针释放某一内存地址等。同时, 数组作为类在 Java 中实现, 良好地解决了数组访问越界这一 C、C++ 中不作检查的错误。

4. 内存管理

C 语言中, 程序员通过库函数 malloc () 和 free () 来分配和释放内存, C++ 中则通过运算符 new 和 delete 来分配和释放内存。再次释放已释放的内存块或未被分配的内存块, 会造成系统的崩溃。

同样, 忘记释放不再使用的内存块也会逐渐耗尽系统资源。而在 Java 中, 所有的数据结构都是对象, 通过运算符 new 为它们分配内存堆。

通过 new 得到对象的处理权, 而实际分配给对象的内存可能随程序运行而改变, Java 对此自动地进行管理并且进行垃圾收集, 有效防止了由于程序员的误操作而导致的错误, 并且更好地利用了系统资源。

5. 数据类型的支持

在 C、C++ 中, 对于不同的平台, 编译器对于简单数据类型如 int, float 等分别分配不同长度的字节数, 例如: int 在 IBM PC 中为 16 位, 在 VAX-11 中为 32 位, 这导致了代码的不可移植性, 但在 Java 中, 对于这些数据类型总是分配固定长度的位数, 如对 int 型, 它总占 32 位, 这就保证了 Java 的平台无关性。

6. 类型转换

在 C、C++ 中, 可以通过指针进行任意的类型转换, 常常带来不安全性, 而 Java 中, 运行时系统对对象的处理要进行类型相容性检查, 以防止不安全的转换。

7. 头文件

C、C++中用头文件来声明类的原型以及全局变量、库函数等，在大的系统中，维护这些头文件是很困难的。而 Java 不支持头文件，类成员的类型和访问权限都封装在一个类中，运行时系统对访问进行控制，防止对私有成员的操作。同时，Java 中用 `import` 语句来与其他类进行通讯，以便使用它们的方法。

8. 结构和联合

C、C++中的结构和联合中所有成员均为公有，这就带来了安全性问题。Java 中不包含结构和联合，所有的内容都封装在类中。

9. 预处理

C、C++中用宏定义来实现的代码给程序的可读性带来了困难。在 Java 中，不支持宏，它通过关键字 `final` 来声明一个常量，以实现宏定义中广泛使用的常量定义。

1.4 Java 的开发工具和执行环境

Java 不但提供了一个特征丰富的语言和运行环境，而且还为程序员和最终用户提供了开发和使用 Java 的工具。

SUN 公司提供的 Java 开发工具包是 JDK (Java Development Kit)。JDK 提供了调试及运行一个 Java 程序所有必需的工具和类库，在真正进入 Java 程序的开发之前，必须首先安装好 JDK，JDK 的最新版本可以在 SUN 公司的主页里免费得到。

在安装 JDK 时，最好是在根目录下创建一个 JDK 目录，再将 JDK 安装到该目录下的 `java` 目录下。正确安装后，在 `Java` 目录下至少应该可以看到如下目录：`bin`，`demo`，`include`，`lib` 等（不同版本的 JDK 会不太一样）。

需要注意的一点是，在安装好 JDK 后，如果用户希望自己的 Java 程序能在任意的目录下运行，则需要更新系统的 `PATH` 环境变量，修改 `PATH` 环境变量可以使得 Windows 能够从当前目录中找到可执行文件（例如：`javac`、`java`、`javadoc` 等），将如下内容添加到 `PATH` 环境变量中即可：（假设 JDK 是安装在 C 盘的根目录下）

```
C:\jdk\java\bin
```

也就是将 Java 可执行文件所在的目录添加到 `PATH` 路径中。

对于 JDK1.3 以前的版本，还要修改 `CLASSPATH` 环境变量使它包括 Java 核心类库文件的目录。这个类库文件的名字为 `classes.zip`。可在 `CLASSPATH` 环境变量中添加如下文字：

```
C:\jdk\java\lib\classes.zip
```

`CLASSPATH` 环境变量告诉 Java 应用程序在哪里寻找 Java 核心类库文件，即 `classes.zip` 文件。

Java 提供了程序员可用来更好地制作并更快运行其程序的扩展类库。Java 类库是一组预先开发的程序码，它可以与单独的应用程序相链接。Java 类库给程序员提供了一个彻底经过测试的健壮性很好的类集合。通过使用这些类库，程序员就不需要重新编写这些代码，从而节约时间，提高效率。这些类库代表了面向对象计算的核心部分。