



JAVA 语言 编程技术

王克宏 主编

郁欣 王曦东 姜河 编著



清 华 大 学 出 版 社

(京)新登字 158 号

内 容 简 介

本书对如何使用 Java 语言进行程序设计进行了详细的介绍。全书共分为三篇:第一篇简单地介绍了 Java 语法(共 6 章),包括 Java 语言中的数据类型、运算符、表达式、基本语句等,这是编程的基础;第二篇通过十章的内容详细而全面地讨论 Java 语言编程技术,内容涉及字符串处理、例外处理、线程、I/O 处理、GUI 设计、Applet、多媒体支持、网络通讯、实用程序类,以及本机方法,在讲述基本原理的同时,还给出大量的示例,以帮助读者更好地运用 Java API 中提供的各种类和接口;第三篇是附录,附录 A 中根据作者的编程实践列出在进行 Java 程序设计时须注意的问题,希望对读者有所帮助,附录 D 则给出部分常用的 API,其中的类和接口在第二篇中基本上都已涉及,目的是为读者在阅读本书以及进行程序设计时提供方便。

本书的内容基于 JDK(Java Developer Kit) 1.0.2。相信通过学习本书会使读者熟练掌握 Java 语言。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无防伪标签者不得销售。

书 名: Java 语言编程技术
作 者: 郁欣 王曦东 姜河编著
出 版 者: 清华大学出版社(北京清华大学校内,邮政编码:100084)
责任编辑: 徐培忠
印 刷 者: 北京丰华印刷厂
发 行 者: 新华书店总店北京科技发行所
开 本: 787×1092 1/16 印张:17.75 字数:366 千字
版 次: 1997 年 5 月第 1 版 1997 年 5 月第 1 次印刷
书 号: ISBN 7-302-02484-7/TP·1255
印 数: 0001—8000
定 价: 26.00 元

序 言

近年来 Internet 以迅猛之势发展,已成为全世界规模最大的计算机网络,网上资源丰富,为世界各国千万用户所瞩目,我国也已于 1994 年 5 月正式入网,并已有成千上万使用者上网工作,如何充分、合理地利用 Internet 的网络与信息资源,为我国社会的各方面服务,是我们应予以重视和研究的问题。

Java 的出现正迎合了 Internet 发展形势的需要,它所体现的简单、不依附于平台、面向对象、分布式、可靠性、安全性、可移植性、动态性、多线程等特性,为 Internet 的使用提供了一种良好的开发和运行环境,成为 Internet 适用、新型的编程语言。Java 出现后立即为世界各大公司所注目,纷纷购买 Java 使用权,并在剧烈的市场竞争中展开了大规模的研究与产品开发。Java 语言及其相关技术同样也引起了我国学术界、产业界和应用界的广泛关注和兴趣。因此 Java 丛书的出版发行,是符合时宜之举,必将受到大家的欢迎。

根据编著者的计划,Java 丛书将包括:《Java 语言入门》,《Java 虚拟机规范》,《Java 语言编程技术》,《Java 语言 Applet 编程技术》,《Java 语言 API 类库》,《Java 软件包的使用》,《Java 语言 SQL 接口》,《Java 语言调试技术》及《HotJava 使用指南》等十多部书,达几百万字。详细深入地介绍 Java 语言及其相关技术并在短期内出齐,为我国广大读者研究和掌握 Java 提供了一种系统而全面的文献资料,无疑是一种十分有意义的事。Java 作为新生事物,尚在不断发展完善之中,因此我希望我们对 Java 及其相关技术不只停留于被动的学习、跟踪与使用,还应以积极主动的态度,通过应用实践和深入分析研究,参与开发创新,为计算机的网络应用做出我们自己的贡献。

清华大学计算机科学与技术系王克宏教授及其知识工程科研组,在 Internet 上进行多层次的研究开发工作已有一段时间,取得了阶段性成果,并曾在不久前召开的“中国计算机学会第九次全国学术大会”的全体会议上作报告,受到与会者的欢迎。相信他们在结合自己研究工作基础上编写的这套 Java 系列丛书,必将有助于我国信息科学技术的发展。我高兴地为丛书作序并祝他们成功!

张淑祥

1996.8

* 本文作者为中国科学院院士、中国计算机学会理事长

开辟 Java 计算的新时代

Sun 公司发明的 Java 语言风靡全球。据 Forrester Research 公司对《幸福》杂志评出的世界 1000 家大型企业进行的调查,目前在它们中间已有 62%正在使用 Java 从事开发工作;有 42%已将 Java 纳入自己企业一年内的战略开发计划。这个调查结果表明,Java 已为大家所接受。

计算机自从 50 年前诞生以来,经历了几个发展时代:终端/主机计算、PC 计算、客户机/服务器计算,而现在已发展到一个新的阶段,即 Java 计算时代。这个时代的特点就是以 Java 为代表的网络计算。

Java 带来的是一场革命。这是第一个真正独立于平台的计算方案,它能充分发挥 Internet 的作用。Java 计算是实现“一个世界,一个网络”构想的关键。这样一个透明的、全球连接的和信息交换的网络,可将所有最新的计算技术、电话、出版/媒体和娱乐集成一体。过去妨碍这一构想成为现实的是计算平台不能兼容,而 Java 语言却以其许多优秀特性使之成为允许各类系统相互兼容和共享应用环境的桥梁。这使各类软件“一次写成,到处可用(write once/run anywhere)”,这样,相同的软件可在不同器件上运行,无论是 PC 机、苹果机、UNIX 计算机,还是顶置盒、PDA(个人数据助理)、移动电话,乃至智能元器件,无一例外。Java 的这一贡献将使全人类受益,因而给人们的生活方式带来极大的改变。

清华大学王克宏教授是最早研究 Java 语言的中国学者之一。在他的周围,许多热心于 Java 计算的青年学者、博士生、硕士生和本科生组成了一支朝气蓬勃、积极进取的生力军,他们思想活跃、工作勤奋、通过努力把 Java 语言的开发和应用提高到一个新的水平。据我所知,在中国的诸多大专院校和科研院所,包括北京航空航天大学、上海复旦大学,也都有许多学者致力于 Java 的研究和开发。由于他们的辛勤努力和积极探求,在 Java 平台上开发和使用的 Java 应用将给中国的信息业带来新的发展,使 Java 计算在中国进一步得到推广。

《Java 系列丛书》就是 Java 开发园中的一朵奇葩,它不倜傥,以其朴实无华、实用有效的风格展示于人。我们希望由王克宏教授主编的该套丛书能赢得广大读者的喜爱,希望能有更多的中国智者和用户在 Java 计算的时代熟悉 Java,掌握 Java,利用 Java 的诸多优点,编写出更多更美的应用软件,从而造福于人类。

谨以此愿祝贺 Java 系列丛书的出版发行。

李永超

1996.10

* 本文作者为 Sun 公司中国/香港地区副总裁

目 录

第一篇 Java 语言规范

第 1 章 Java 语言概述	1
1.1 Java 语言发展简史	1
1.2 Java 的特点	1
1.2.1 Java 语言的特点	2
1.2.2 Java Applet(小应用程序)的特点	3
1.2.3 Java 和 C、C++ 的比较	3
1.3 Java 程序举例	4
1.3.1 Java 应用程序(Java Application)	4
1.3.2 Java 小应用程序(Java Applet)	5
第 2 章 简单数据类型	7
2.1 数据类型综述	7
2.2 简单数据类型的实例化——变量与常量	7
2.2.1 标识符(Identifiers)	8
2.2.2 常量(Constant variables)	8
2.2.3 变量(Variables)	8
2.3 整数类型	9
2.3.1 整型常量	9
2.3.2 整型变量	9
2.4 浮点型(实型)数据	10
2.4.1 实型常量	10
2.4.2 实型变量	11
2.5 字符类型	11
2.5.1 字符型常量	11
2.5.2 字符型变量	12
2.6 布尔类型	12
2.7 各类型数据间的优先关系和相互转换	12
2.7.1 数据类型的缺省值	12
2.7.2 自动类型转换	13

• III •

2.7.3 强制类型转换	13
--------------------	----

第3章 运算符和表达式 14

3.1 运算符综述.....	14
3.2 算术运算符.....	14
3.3 关系运算符.....	15
3.4 布尔逻辑运算符.....	15
3.5 位运算符.....	15
3.5.1 按位取反运算符(~)	16
3.5.2 按位与运算符(&)	16
3.5.3 按位或运算符()	16
3.5.4 按位异或运算符(^)	16
3.5.5 左移运算符(<<).....	16
3.5.6 右移运算符(>>).....	16
3.5.7 无符号右移运算符(>>>)	16
3.5.8 不同长度的数据进行位运算	16
3.6 赋值运算符.....	17
3.6.1 赋值运算符(=)	17
3.6.2 扩展赋值运算符	17
3.7 条件运算符.....	18
3.8 表达式.....	18

第4章 控制语句 20

4.1 分支语句.....	20
4.1.1 条件语句 (if-else)	20
4.1.2 多分支语句 (switch)	21
4.1.3 break 语句	21
4.1.4 返回语句 (return)	22
4.2 循环语句.....	23
4.2.1 while 语句.....	23
4.2.2 do-while 语句	23
4.2.3 for 语句	24
4.2.4 continue 语句	24
4.3 例外处理语句.....	25
4.4 注释语句.....	25

第5章 复合数据类型(一)——类与接口 26

5.1 面向对象的程序设计基本概念.....	26
------------------------	----

5.1.1	对象	26
5.1.2	消息	26
5.1.3	类	27
5.2	类	27
5.2.1	类声明	27
5.2.2	类体	27
5.2.3	成员变量	28
5.2.4	方法	29
5.2.5	方法重载	30
5.2.6	构造方法	30
5.2.7	finalize()方法	30
5.2.8	成员变量和方法的访问权限	30
5.2.9	实例成员和类成员	31
5.3	对象	32
5.3.1	对象的生成	32
5.3.2	对象的使用	32
5.3.3	对象的清除	33
5.4	面向对象程序设计的技术——继承和多态	33
5.4.1	继承	33
5.4.2	多态	34
5.5	接口	35
5.5.1	接口的定义	35
5.5.2	接口的实现	35
5.5.3	接口类型	36
第 6 章	复合数据类型(二)——数组	37
6.1	一维数组	37
6.1.1	一维数组的定义(声明)	37
6.1.2	一维数组的初始化	37
6.1.3	一维数组元素的引用	38
6.1.4	一维数组在程序中的使用	39
6.2	多维数组	40
6.2.1	二维数组的定义	40
6.2.2	二维数组的初始化	40
6.2.3	二维数组元素的引用	41
6.2.4	二维数组在程序中的使用	41
6.3	数组与对象的关系	42

第二篇 Java 语言编程技术

第 7 章 字符串处理	43
7.1 生成字符串	43
7.1.1 文字串	43
7.1.2 String	43
7.1.3 StringBuffer	44
7.2 访问字符串	44
7.2.1 String	45
7.2.2 类 StringBuffer	46
7.3 修改字符串	47
7.3.1 String	47
7.3.2 StringBuffer	48
7.4 其它处理	49
7.4.1 字符串的比较	49
7.4.2 字符串的转化	51
7.4.3 运算符‘+’的重载	51
7.4.4 词法分析	51
7.5 小结	52
第 8 章 例外处理	54
8.1 例外	54
8.1.1 例外示例	54
8.1.2 例外处理机制	55
8.1.3 例外处理的类层次	56
8.2 例外处理	57
8.2.1 捕获例外	57
8.2.2 声明抛弃例外	61
8.2.3 抛弃例外	61
8.3 使用例外类	61
8.4 小结	62
第 9 章 线程	63
9.1 线程的概念	63
9.2 线程的属性	63
9.2.1 线程体	63

9.2.2	线程的状态	66
9.2.3	线程的调度	68
9.2.4	Daemon 线程	69
9.3	线程组	70
9.3.1	线程和线程组	70
9.3.2	类 ThreadGroup	70
9.4	多线程程序	72
9.4.1	监控器	72
9.4.2	同步控制	74
第 10 章 输入/出处理		82
10.1	I/O 处理的类层次	82
10.2	InputStream 和 OutputStream	83
10.2.1	InputStream	84)
10.2.2	OutputStream	84
10.3	文件处理	85
10.3.1	文件描述	85
10.3.2	文件 I/O 处理	90
10.3.3	随机访问文件	92
10.4	管道流	94
10.5	内存的读/写	96
10.6	连接多个输入流	98
10.7	过滤流	100
10.7.1	BufferedInputStream 和 BufferedOutputStream	100
10.7.2	LineNumberInputStream	101
10.7.3	DataInputStream 和 DataOutputStream	101
10.7.4	PushbackInputStream	101
10.7.5	PrintStream	103
10.7.6	自定义过滤流	103
10.8	StreamTokenizer	106
10.8.1	流解析语法表的初始状态	106
10.8.2	stream Tokenizer 对重设语法表的支持	106
10.8.3	用 stream Tokenizer 重写 10.7.4 中的例 10.9	107
10.9	小结	108
10.9.1	I/O 例外	108
10.9.2	流结束的判断	108

第 11 章 GUI 设计	109
11.1 AWT 概述	109
11.1.1 AWT 组件的类层次	109
11.1.2 类 Component	109
11.1.3 类 Container	111
11.2 AWT 基本组件	111
11.2.1 按钮(Button)	111
11.2.2 画布(Canvas)	112
11.2.3 校验盒(Checkbox)	112
11.2.4 选择框(Choice)	113
11.2.5 标签(Label)	114
11.2.6 列表(List)	114
11.2.7 滚动条(Scrollbar)	115
11.2.8 文本区(TextField 和 TextArea)	116
11.3 AWT 容器	118
11.3.1 面板(Panel)	118
11.3.2 窗口(Window)	119
11.3.3 框架(Frame)	119
11.3.4 对话框(Dialog)	120
11.3.5 文件对话框(FileDialog)	122
11.4 菜单	122
11.4.1 菜单条(MenuBar)	122
11.4.2 菜单项(MenuItem)	122
11.4.3 校验菜单项(CheckboxMenuItem)	122
11.4.4 菜单(Menu)	122
11.4.5 举例	122
11.5 布局	124
11.5.1 FlowLayout	124
11.5.2 BorderLayout	125
11.5.3 GridLayout	126
11.5.4 GridBagLayout	126
11.5.5 CardLayout	130
11.5.6 null 和自定义布局管理器	130
11.6 AWT 事件处理机制	130
11.6.1 事件(Event)	131
11.6.2 事件处理	131
11.7 GUI 设计示例	134

11.7.1	GUI 设计的基本步骤	134
11.7.2	实例 GUIDemo	135
11.7.3	例程中的类	145
11.7.4	组件层次及布局设计	146
11.7.5	事件处理	146
11.7.6	程序运行	147
11.8	AWT 组件的同等类	148

第 12 章 Java Applet 149

12.1	Applet 简介	149
12.2	Applet 与 HTML	150
12.3	Applet 的执行框架	151
12.4	Applet 的 GUI 设计	152
12.4.1	Applet 常用的几种方法	152
12.4.2	Applet 的性质	152
12.5	Applet 通讯	153
12.5.1	同页 Applet 的通讯	153
12.5.2	与浏览器通讯	157
12.5.3	网络通讯	162
12.6	小结	163
12.6.1	安全性限制	163
12.6.2	Applet 设计技巧	163
12.7	Applet 实例——15 子棋游戏	164

第 13 章 多媒体支持 170

13.1	声音	170
13.2	AWT 绘图	170
13.3	图形	171
13.4	文字	174
13.5	图象	177
13.5.1	载入图象	178
13.5.2	显示图象	179
13.5.3	生成图象	181
13.5.4	处理图象	182
13.6	动画	185
13.6.1	动画实例	185
13.6.2	动画技巧	187
13.6.3	动画模板	189

第 14 章 网络通讯	194
14.1 URL	194
14.1.1 URL 的组成和类 URL	194
14.1.2 通过 URL 访问 WWW	196
14.1.3 通过 URLConnection 连接 WWW	197
14.2 socket 通讯	198
14.2.1 socket 通讯的步骤	198
14.2.2 打开 socket	198
14.2.3 打开输入/出流	199
14.2.4 关闭 socket	199
14.2.5 示例	200
14.3 数据报通讯	202
14.3.1 数据报通讯和流式通讯	202
14.3.2 DatagramSocket 和 DatagramPacket	203
14.4 多客户机制	207
14.5 安全性	217
14.5.1 Applet 的安全考虑	218
14.5.2 对安全性的一些建议	222
第 15 章 实用程序类	223
15.1 基本数据类型封装	223
15.1.1 数字	223
15.1.2 字符类 Character	224
15.1.3 布尔类 Boolean	224
15.2 算术运算	224
15.3 对象枚举	224
15.3.1 接口 Enumeration	225
15.3.2 线性表	225
15.3.3 相联表	226
15.4 系统资源	228
15.4.1 类 System	228
15.4.2 Runtime 对象	230
15.5 日期处理	230
15.6 处理程序属性	231
15.6.1 类 Properties	231
15.6.2 系统属性	232
15.6.3 命令行参数	234

15.6.4	Applet 参数	235
第 16 章 集成本机方法		236
16.1	集成本机方法的步骤	236
16.1.1	编写 Java 源文件	236
16.1.2	编译 Java 源文件	237
16.1.3	生成头文件	237
16.1.4	生成存根文件	238
16.1.5	编写 C 语言文件	238
16.1.6	生成动态库	238
16.1.7	运行 Java 程序	239
16.2	实现本机方法	239
16.2.1	贯穿始终的例子	239
16.2.2	方法与函数	242
16.2.3	参数传递	243
16.2.4	返回类型	244
16.2.5	访问 Java 对象	244
16.3	实用函数	245
16.3.1	字符串处理	245
16.3.2	多线程同步处理	247
16.3.3	例外处理	247

第三篇 附录

附录 A	须注意的问题	249
附录 B	保留字	252
附录 C	包	253
附录 D	部分类和接口	254

第一篇 Java 语言规范

第 1 章 Java 语言概述

本书所讲述的是如何使用 Java 语言编程序,因此首先介绍 Java 语言的基础知识。在这一章里,先来回顾一下 Java 语言的发展历程,然后介绍 Java 语言应用于网络计算的优点,并把 Java 与 C/C++ 等传统程序设计语言相对比进一步指出它所具有的优势。最后,通过两个例子来说明 Java 程序的结构。

1.1 Java 语言发展简史

Java 语言的设计可以追溯到 1991 年,SUN MicroSystem 公司为了开发消费电子产品市场,成立了由 Jame Gosling、Bill Joe 等人组成的 Green 开发小组。他们发现现有的语言如 C/C++ 等在安全性和对芯片与操作系统的依赖性方面并不适合家用消费电子产品,于是他们开发了一个名为 Oak (一种橡树的名字)的软件,但当时并没有引起人们的注意。后来 Oak 演变为更实用的系统 Star Seven(* 7),但这些技术上成功的产品并未获得商业上的成功。

直到 1994 年下半年,Internet 的迅猛发展,环球信息网 WWW 的快速增长,使得 SUN 公司把 Oak 的技术应用于网络,并正式命名为 Java。1995 年,SUN 公司正式发布了 Java 语言,Java 在网络上的独特优势使得它逐渐成为 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,如 MicroSoft、IBM、Netscape、Novell、Apple、DEC、SGI 等。1996 年 SUN 公司成立了 Javasoft 分公司来维护和发展 Java 语言。

1.2 Java 的特点

Java 语言的广泛应用是因为它在网络编程方面有传统程序设计语言难以比拟的特点。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、鲁棒性、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机器和操作系统的限制。

下面分别讨论 Java 和 Java Applet 的特点,然后通过把 Java 与 C、C++ 相对比进一步

指出它所具有的优点。

1.2.1 Java 语言的特点

Java 语言有下面一些特点:简单性、面向对象、分布式、解释执行、鲁棒、安全、体系结构中立、可移植、高性能、多线程以及动态性。下面分别加以介绍。

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收集,大大简化了程序设计者的内存管理工作。另外,Java 语言由 C/C++ 发展而来,语言风格十分类似,易于学习。

2. 面向对象

Java 语言区别于传统程序语言的一大特点是支持面向对象的程序设计,使开发者把设计集中于对象及其接口。Java 语言提供了简单的类机制以及动态的接口模型,对象中封装了成员变量和相应的方法,实现了数据封装和信息隐藏;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. 鲁棒性

Java 在编译和运行程序时,都要对可能出现的问题进行检查,以消除错误的产生。它提供自动垃圾收集来进行内存管理,防止程序员在管理内存时容易产生的错误。通过集成的面向对象的例外处理机制,在编译时,Java 提示出可能出现但未被处理的例外,帮助程序员正确地进行选择以防止系统的崩溃。另外,Java 在编译时还可捕获类型声明中的许多常见错误,防止动态运行时不匹配问题的出现。

5. 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,这样就防止了程序员使用“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。

6. 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。

7. 可移植性

与平台无关的特性使 Java 程序可以方便地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 语言实现,这使得 Java 系统也具有可移植性。

8. 解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码携带了许多编译时信息,使得连接过程更加简单,使用起来更加方便。

9. 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。

10. 多线程

多线程机制使应用程序能够并发执行,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。

11. 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。

1.2.2 Java Applet(小应用程序)的特点

Java 语言的特性使它可以最大限度地利用网络,而 Applet 是 Java 语言迅速流行的主要原因。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 从网络上进行传输,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 可以和图象、声音、动画等一样从网络上下载,但它不同于这些多媒体的文件格式,它可以接收用户的输入,动态地进行改变,而不仅仅是动画的显示和声音的播放。

1.2.3 Java 和 C、C++ 的比较

对于变量声明、参数传递、操作符、控制语句等方面,Java 使用了和 C、C++ 相同的风格,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、鲁棒、安全等特性,也摒弃了 C 和 C++ 中许多不合理的内容。下面介绍 Java 语言和 C、C++ 的区别。

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。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、finally 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

Java 不支持 C/C++ 中的指针,因为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。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。

4. 内存管理

在 C 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,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,有效地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. 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,而在 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。

6. 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。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,以使用它们的方法。

7. 结构和联合

C、C++ 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。

1.3 Java 程序举例

1.3.1 Java 应用程序(Java Application)

下面举例说明 Java 应用程序的结构:

```
// filename: HelloWorld.java
public class HelloWorldApp
{
    public static void main (String args[ ])
    {
        System.out.println("Hello World!");
    }
}
```

使用下述命令在 JDK 中编译并运行程序:

```
javac HelloWorldApp.java
java HelloWorldApp
```

该程序运行后输出下面一行信息:

Hello World!

上面程序有如下部分组成:

1. **注释:**用//起始的一行为注释行,在上面程序中,说明了文件名,文件名要与类名一致。