



深入浅出

Java

语言程序设计

李云山、吕杰武 / 编著



由国内资深 Java 专家精心编著，融会作者多年的实际编程经验
本书内容全面、系统，知识讲解深入浅出，堪称 Java 语言的权威教材
每个知识点都辅以实例进行讲解，让您迅速掌握 Java 语言的精髓
内容涵盖 Sun Java 认证考试范围，可作为 SCJP 认证考试的参考用书

读者对象：

本书既适合作为初、中级读者学习 Java 语言的入门图书，同时也可作为
大中专院校及培训班学生理想的 Java 语言程序设计教材



随书附赠光盘内含：

书中所讲范例的完整源代码，书中练习题答案及源代码，
相关 Java 开发工具及官方文档



中国青年出版社

计 算 机 应 用 深 入 浅 出 丛 书



深入浅出

Java

语言程序设计

李云山、吕杰武 / 编著



中国青年出版社
CHINA YOUTH PRESS

(京)新登字 083 号

本书由中国青年出版社独家出版。未经出版者书面许可,任何单位和个人不得以任何形式复制或传播本书的部分或全部内容。

图书在版编目(CIP)数据

深入浅出 Java 语言程序设计 / 李云山, 吕杰武编著. - 北京: 中国青年出版社, 2003

ISBN 7-5006-4966-5

I. 深... II. ①李... ②吕... III. JAVA 语言 - 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 104247 号

总 策 划: 胡守文

王修文

郭 光

责任编辑: 曹 建

肖 辉

王学英

责任校对: 王志红

书 名: 深入浅出 Java 语言程序设计

编 著: 李云山、吕杰武

出版发行: 中国青年出版社

地址: 北京市东四 12 条 21 号 邮政编码: 100708

电话: (010) 84015588 传真: (010) 64053266

印 刷: 沈阳铁路局锦州印刷厂

开 本: 787 × 1092 1/16 印 张: 23

版 次: 2003 年 2 月北京第 1 版

印 次: 2003 年 2 月第 1 次印刷

书 号: ISBN 7-5006-4966-5/TP · 292

定 价: 36.00 元 (ICD)

前 言

近年来,随着 Internet 以及信息技术的发展,人们对网络的需求越来越大,同时对网络应用程序的要求也越来越高,特别是一些能在不同的操作系统平台上运行的应用程序。Java 程序语言的出现恰恰迎合了这种跨平台应用程序的需求,并且得到了非常广泛的应用。Java 程序语言是一种完全面向对象的程序语言,它有着强大的连网功能与分布计算能力,以及高度的安全性和健壮性。跨平台的特性更使得 Java 开发出来的程序可以“一次编写,到处运行”。所有这些都是 Java 语言深受众多程序员青睐的原因。

本书对 Java 2 (标准版) 程序设计语言作了较为全面的介绍,并结合实际的编程经验,对一些编程技巧以及编程中应该注意的问题也做了详细的说明。整本书的内容遵循 Sun 公司 Java 标准教材的内容范围编写,让读者看完本书之后,除能全面掌握 Java 语言的基本知识外,还可以进一步参加 Java 语言的认证考试 (SCJP)。

本书前 10 章为 Java 编程的基础知识,包括 Java 编程简介、Java 语法以及面向对象的程序设计;第 11 章~第 13 章为图形编程部分,主要介绍如何应用 Swing 工具包编写 Windows 风格的程序;第 14 章~第 20 章为一些较为高级的编程技术,包括异常处理、多线程、SQL 数据库编程、网络编程、输入输出、文件操作等。第 21 章为 Java 程序设计规范;第 22 章则介绍了 Java 的各项高级开发技术。另外,本书的前 20 章每章后面都有精心设计的练习题,这些练习题对读者巩固各章的知识、增加实际的编程经验起着非常重要的作用。

本书中的例题与习题均在 Windows 98/2000 平台下测试通过,并且范例的源程序与习题答案全部保存于附赠的光盘中,同时我们还在光盘中特意为读者准备了学习时所用的一些 Java 开发工具与开发文档。本书既适合作为初、中级读者学习 Java 语言的入门图书,同时也是大、中、专院校及培训班学生理想的 Java 程序设计教材。

另外,本书在编写的过程中,得到了众多朋友的支持。其中本书的第 1、2、3 章由李云山编写,第 4、5 章由李晓斌编写,第 6、7、8 章由吕杰武编写,第 9、10 章由林世勇编写,第 11、12、13 章由舒志、李钧编写,第 14 章由唐金翔编写,第 15、16 章由郭学旭编写,第 17、18 章由卢哲军编写,第 19、20 章由曾平编写,第 21、22 章由王宇编写。在这里笔者本人向所有为编写此书辛勤工作的人们表示衷心的感谢。

限于笔者水平有限,如您对书中的内容或程序有何意见,请不吝指正。

编者

2002 年 9 月

目 录

第1章 Java从哪里来

1.1 Java——一种程序设计语言	2
1.2 跨越各种操作系统的特性	3
1.3 Java 与 Internet	4
练习一	5

第2章 Java 的开发工具

2.1 什么是开发工具	8
2.2 Java 开发工具的分类	8
2.3 Java 虚拟机	9
2.4 Java 开发工具包(JDK)	9
2.5 安装和设置 JDK	11
2.6 使用 JDK 编译、运行 Java 程序	12
练习二	13

第3章 Java 小应用程序(Applet)

3.1 什么是 Applet	16
3.2 第一个例子	16
3.3 第二个例子	18
练习三	19

第4章 Java 应用程序(Application)

4.1 什么是 Application	22
4.2 第一个例子	22
4.3 第二个例子	23
练习四	25

第5章 基本元素

5.1 运算符	28
5.1.1 运算符分类	28
5.1.2 运算符优先级	31
5.2 数据类型	32
5.2.1 基本数据类型	32
5.2.2 基本数据类型的缺省值	34
5.2.3 基本数据类型的转换	34
5.3 变量	35
5.4 常量	35
5.5 表达式	36

5.6 语句	37
5.7 注释	38
练习五	39

第6章 结构控制

6.1 if 结构	44
6.1.1 if	44
6.1.2 嵌套 if	44
6.1.3 阶梯 if	45
6.1.4 短路表达式	46
6.2 for 循环	47
6.2.1 循环结构	47
6.2.2 for 循环	47
6.2.3 特殊的 for 循环	48
6.2.4 嵌套 for 循环	49
6.3 while 循环	49
6.4 do 循环	50
6.5 switch 结构	51
6.6 break 语句	52
6.7 continue 语句	53
6.8 return 语句	54
6.9 try 结构	54
练习六	55

第7章 数组

7.1 一维数组	58
7.1.1 一维数组的声明	58
7.1.2 一维数组的初始化	58
7.1.3 访问数组元素	59
7.1.4 一维数组的地址空间模型	60
7.1.5 数组的复制	61
7.2 基本类型数组	62
7.3 对象数组	63
7.4 二维数组和 multidimensional 数组	64
7.4.1 二维数组的声明	64
7.4.2 二维数组的初始化	64

7.4.3 多维数组的地址空间模型	65
7.4.4 使用多维数组	66
7.5 查找	67
7.6 排序	69
练习七	72
第8章 字符串	
8.1 字符串的构造	76
8.2 字符串的基本操作	77
8.3 字符串的查找	80
8.4 字符串的比较	82
8.5 字符串与基本数据类型的转换	84
8.5.1 将字符串转换成基本数据类型	84
8.5.2 将基本数据类型转换成字符串	85
8.6 StringBuffer 类	86
8.6.1 StringBuffer 对象的构造	86
8.6.2 StringBuffer 对象的基本操作	87
8.6.3 StringBuffer 对象的修改操作	87
8.7 字符串数组	89
练习八	90
第9章 类	
9.1 类的声明	94
9.1.1 类修饰符	94
9.1.2 类体	94
9.2 成员变量	95
9.3 方法	96
9.4 特殊的方法——构造器	96
9.5 类的实例化	97
9.6 方法的调用	98
9.6.1 传递基本类型的参数	98
9.6.2 传递对象参数	99
9.6.3 方法的递归调用	100
9.7 包	101
9.7.1 包的作用	101
9.7.2 包的声明	101
9.7.3 包与目录	102
9.8 标准 Java 包	102
9.9 类的访问和访问控制	103
练习九	104

第10章 面向对象的程序设计

10.1 与面向过程的程序语言的区别	106
10.2 一切都是对象	107
10.3 封装	108
10.4 继承	111
10.4.1 继承的语法	111
10.4.2 继承举例	112
10.5 多态	116
10.6 方法重载	117
10.7 抽象类	118
10.8 静态类成员	120
10.9 接口	122
10.9.1 接口的声明	122
10.9.2 接口的实现	123
10.9.3 接口的应用	123
练习十	125
第11章 图形编程	
11.1 Java 图形坐标系	128
11.2 绘制基本图形	130
11.3 填充图形	134
11.4 字体	137
练习十一	140

第12章 事件

12.1 Swing 简介	142
12.2 事件模型	145
12.3 事件结构	147
12.4 窗口事件和适配器	150
12.5 焦点事件	152
12.6 键盘事件	154
12.7 鼠标事件	157
练习十二	159

第13章 美丽的界面

13.1 概述	162
13.2 布局管理器	162
13.2.1 边框布局管理器	163
13.2.2 流式布局管理器	164
13.2.3 网格布局管理器	165
13.3 容器	166
13.3.1 最底层的容器	167

13.3.2 面板	168
13.4 文本输入组件	169
13.4.1 文本域和密码域	169
13.4.2 文本区	171
13.5 按钮	173
13.5.1 普通按钮	173
13.5.2 复选框	174
13.5.3 单选按钮	175
13.6 列表框和组合框	178
13.6.1 列表框	178
13.6.2 组合框	180
13.7 表格	182
13.8 菜单	183
13.8.1 建立普通菜单	183
13.8.2 复选菜单和单选菜单	185
13.9 简单对话框	187
13.10 高级布局管理器	188
13.10.1 卡片布局管理器	189
13.10.2 网袋布局	190
练习十三	192
第 14 章 异常处理	
14.1 什么是异常	194
14.2 Java 中的异常类	194
14.2.1 异常类层次结构	195
14.2.2 Throwable 类	196
14.3 异常的捕获与处理	196
14.3.1 使用 try-catch 语句	197
14.3.2 多个 catch 结构	197
14.3.3 finally	198
14.4 throws 和 throw	199
14.5 定义自己的异常类	201
练习十四	203
第 15 章 多线程	
15.1 进程和线程	206
15.2 线程的建立	206
15.2.1 从 Thread 继承	207
15.2.2 实现 Runnable 接口	208
15.3 线程的状态及其转换	208
15.3.1 线程的状态	209

15.3.2 状态之间的转换	209
15.4 线程应用——计时器	212
15.5 在 Applet 中使用线程	214
15.6 多线程	215
15.6.1 线程优先级	215
15.6.2 线程组	216
15.6.3 多线程例程	217
15.7 同步	218
练习十五	221
第 16 章 使用数据库	
16.1 什么是数据库(Database)	224
16.1.1 数据库的基本概念	224
16.1.2 分类和流行的数据库系统	225
16.1.3 目前数据库的使用	225
16.2 Microsoft Access	226
16.3 数据库的标准查询语言	231
16.4 JDBC 简介	231
16.4.1 JDBC 的概念	232
16.4.2 JDBC 的基本内容	233
16.4.3 一个 JDBC 所支持的实例	236
16.5 怎样用 JDBC 连接数据库	237
16.5.1 JDBC 连接数据库的基本概念	237
16.5.2 ODBC 数据源的建立	239
16.5.3 一个 Java 连接 ODBC 数据源 数据库的实例	242
16.6 查询	244
16.6.1 SQL 语句的表单查询	245
16.6.2 两个综合查询的例子	250
16.7 在数据库中插入记录	255
16.8 修改记录	255
16.9 删除记录	256
16.10 在数据库中创建表	257
16.11 修改表和删除表	258
16.12 获得数据库的属性-元数据	261
16.13 多表联合查询和嵌套查询	262
16.14 小节	264
练习十六	265
第 17 章 网络编程	
17.1 基本概念与协议	268

17.1.1 IP 地址	268	19.6 临时文件	308
17.1.2 端口	268	19.7 顺序文件访问	309
17.1.3 客户机与服务器	269	19.8 随机文件访问	311
17.1.4 连接与无连接	269	19.9 比较文件的内容	314
17.1.5 协议	269	练习十九	315
17.2 利用 URL 获取 Internet 资源	270	第 20 章 Java 实用语言支持	
17.2.1 URL 类	270	20.1 Vector 类	318
17.2.2 获取 URL 的信息	271	20.2 Stack 类	320
17.2.3 获取网络图片	273	20.3 LinkedList 类	322
17.3 套接字	274	20.4 Arrays 工具类	324
17.3.1 Socket 和 ServerSocket	274	20.5 Collections 工具类	325
17.3.2 Socket 的通讯步骤	276	20.6 高级排序	326
17.4 一个简单的 Socket 通讯程序	277	练习二十	328
17.5 多客户连接	279	第 21 章 良好的程序设计规范	
17.6 发送电子邮件	281	21.1 整洁的程序代码	330
练习十七	283	21.2 断行与缩进	331
第 18 章 Java 的输入/输出		21.3 空行与空格	332
18.1 System 类	286	21.3.1 空行	332
18.2 控制台输入/输出	287	21.3.2 空格	332
18.3 输入字符	289	21.4 注释	333
18.4 输出字符	291	21.5 遵守 Java 的命名规范	334
18.5 数据流	293	21.6 让别人看懂自己的程序	335
18.6 对象流	294	第 22 章 高级 Java 技术	
18.7 缓冲区	297	22.1 Servlet 小服务程序	338
18.8 小应用程序控制台	298	22.2 JSP	340
练习十八	300	22.3 本地方法调用	341
第 19 章 文件操作		22.4 远程方法调用	344
19.1 Java 中的文件管理——File 类	302	22.5 JavaBeans 与 EJB	346
19.2 创建文件	303	22.6 J2EE	347
19.3 查询文件属性	304	附录 1 Internet Java 资源	
19.4 获得文件列表	306	附录 2 关键字	
19.5 删除文件	307	附录 3 Java 文档	



第 1 章

Java 从哪里来

- 1.1 Java —— 一种程序设计语言
- 1.2 跨越各种操作系统的特性
- 1.3 Java 与 Internet

Java 的诞生需追溯到 1991 年。当时, SUN MicroSystem 公司的 Jame Gosling, Bill Joe 等人, 为在电视、控制烤面包箱等家用电子产品上进行交互式操作而开发了一个名为 Oak 的软件。起初, Oak 并没有引起人们的注意, 甚至可以说是默默无闻。到了 1994 年下半年, Internet 的迅猛发展以及 www 的快速增长, SUN 的工程师将 OAK 技术应用于 Web 上, 并且开发出了 HotJava 的第一个版本。在 1995 年下半年, SUN 公司正式以 Java 这个名字推出的时候, 几乎所有的 Web 开发人员都意识到 Java 正是他们想要的, 从而引发了遍及全球、至今未衰的 Java 热潮。

1.1 Java——一种程序设计语言

Java 是一种优秀的程序设计语言。它最大的优点就是与操作系统无关, 在 Microsoft Windows、UNIX、Linux 以及 MacOS 等多种操作系统上, 都可以使用相同的代码。“一次编写, 到处运行”是 Java 的特色所在, 这使得 Java 在互联网上被广泛采用。另外 Java 语言是完全面向对象的。在 Java 语言中, 除极少数基本类型(如数字)外, 任何一个东西都是对象(对于面向对象的程序设计将在第 10 章详细介绍)。此外, Java 语言与 C++ 语言在语法上类似, 这对于熟悉 C++ 的程序员来说, 学起来效率更高。

Java 有以下基本特点:

简单

Java 是一种比较简单的“面向对象”程序设计语言。一方面, Java 语言省略了 C++ 语言中一些难以理解的、容易混淆的特性, 例如头文件、指针、结构、单元、运算符重载、虚拟基础类等; 另外, Java 语言不支持 goto 语句, 去掉了多亲继承。同 C++ 相比, 避免了很多麻烦问题。

另一方面, Java 程序对系统的要求较低。Java 语言采用了特殊的软件构造方法, 可以在很小的机器上独立运行。只要 40K 就可以容下一个最基本的解释器, 即使加上基本标准库和线程支持, 也仅需再加上 175K 就足够了。这也是 Java 语言的一个成功秘诀。

面向对象

面向对象是软件工程的一次革命, 大大提高了软件开发的效率和水平。在过去的 30 多年里, 面向对象的程序设计充分体现了自己的价值, 到现在已经形成了软件工程的一整套体系。

Java 充分利用了面向对象的思想, 将数据封装于类中, 利用类的优点, 实现了程序的简洁性和便于维护性。类的封装性、继承性等有关对象的特性, 使程序代码只需一次编译, 反复利用。程序员只需把主要精力用在类和接口的设计及应用上。

分布式

Java 语言具有功能强大的、易于使用的网络能力。Java 应用程序可以像访问本地文件系统那样通过 URL 访问远程对象, 非常适合于分布式系统的开发。比如: 使用 Java 在基于

Internet 的开发中，就可以将客户端放在中国，而数据库服务器放在美国。

解释型

Java 编译器产生的是字节代码，而不是本地的机器代码（能直接运行）。编译后的 Java 程序由 Java 虚拟机(Java Virtual System)解释成本地机器码执行。虽然，解释性语言的效率比较低，但正因为是解释执行，不同的平台解释成不同的机器码，因此，才有了 Java 的跨平台特性。

健壮性

Java 语言有非常良好的健壮性，因为 Java 采取了“未雨绸缪”的策略，将重点放在早期潜在问题的检查以及运行时检查，这种防范于未然的措施有效地提高了程序的健壮性。另外 Java 自己操纵内存很大程度上减少了内存出错的可能。

安全性

在当今的网络环境下，安全问题越来越成为人们关注的对象。相对于其他编程语言，Java 有着不可比拟的安全性，这也是 Java 倍受青睐的原因之一。Java 编译器并不控制内存的分配，只是在程序运行时，由操作系统分配，有效避免了不友善代码的破坏或病毒程序的入侵。

可移植性

Java 的平台无关性造就了 Java 高度的可移植性。Java 程序可以方便地从一种操作系统不加改动或只做很少的修改移植到另一种系统。Java 语言当初设计的理念就是“一次编写，到处运行”，它的可移植性是一种必然。

多线程

Java 的多线程机制使应用程序能够并发执行，而且，它的同步机制保证了对共享数据的正确操作。Java 提供了对线程的内置语言的支持，使得用它进行多线程编程非常简单。

动态

Java 语言非常适合于变化发展的环境，可以在类库中自由的加入新的方法和成员变量。此外，Java 语言在需要时才装入类，甚至可以通过网络装载，这对于网络程序来说，特别有用。

1.2 跨越各种操作系统的特性

前面已经提到过，“一次编写，到处运行”是 Java 语言的一大特点，事实上，这也正是 Java 语言最大的优点。

对于一个程序员而言，写出来的程序如果不需修改就能够同时在多种不同的操作系统平台上运行，确实是一件梦寐以求的事！Java 高度的可移植性使得这个用其他语言很难实现甚至无法实现的事变得非常的简单。使用 Java 编写的程序，只要做很少的修改，甚至根本不作修改就可以在不同平台上运行了。

Java 的跨越各种操作系统的特性归根于它是一种解释性语言。Java 编译器生成的是一种字节码，而非本地的机器码。不同的平台由于硬件或操作系统不同，对于一些数据类型的表示是不一样的，比如对于整型（int）数据的表示，有的平台可能是用 16 位来表示的，有的平台却是用 32 位来表示的。为了消除平台的差异，Java 采用了一种中性结构，它生成的字节码同任何一种特定的平台都无关。只有到了在具体的某个平台上执行时，才由 Java 虚拟机解释成平台相关的本地机器码来执行。

1.3 Java 与 Internet

从 Java 的诞生到现在不到 10 年的时间里，它能发展到今天的局面，与 Internet 以及 WWW 的迅速发展是分不开的。

早期的 WWW 服务形式十分死板，都是一些乏味的 HTML 文档。随着 Internet 的迅速发展，WWW 服务变得越来越普遍，人们需要一个好的浏览器来阅读生动的 Web 页面，SUN 的工程师抓住这一需求，编写了一个名为 HotJava 的浏览器，在这个浏览器中，提供了对 Java 小应用程序（Applet）的支持，它是第一个支持 Applet 的浏览器，这种设计受到了广大 Web 程序员的欢迎，并取得了巨大的成功，同时也极大地促进了 Java 的发展。1995 年秋，Netscape 公司决定让其浏览器支持 Java，后来一些著名的 IT 公司如 IBM，Novell，Apple，DEC 等都纷纷购买了 Java 语言的使用权，微软公司也在 IE 浏览器中提供了对 Java 的支持。

Java 的 Applet 极大地丰富了网页，它嵌入在 HTML 文件中，从网上下载 Java 字节码到本地机上运行。这种小应用程序有着优良的交互性能，而且十分小巧，便于网络传输，在网页设计中被广泛地应用。虽然，ActiveX 技术也有着强大的交互功能，但需要在本地安装才能执行，这会带来一些不安全因素。另外，Java 有跨越操作系统的特性，这是 ActiveX 所不能相比的。

也许会有人怀疑 Java Applet 的安全性，在这一点上大可放心。前面已经提到过，Java 语言本身有着非常高的安全性，有许多措施防御了不可靠的 Java Applet。但要做到绝对安全，几乎是一件不可能的事，因为当今的网络本身就有许多不安全的因素，没有任何一种语言是绝对安全的，Java 也不例外。与其他程序设计语言相比，Java 有更高的安全性。

虽然，Java 的 Applet 深受人们的青睐，但 Java 的网络功能并不仅仅局限于 Applet。利用 Java 编写客户机/服务器结构的网络程序以及分布式系统要比其他编程语言容易得多。在 Internet 迅速发展的今天，人们对网络的依赖越来越强，因而对网络应用程序的要求也越来越高。那么，用什么语言来开发网络上的程序呢？Java 便是一个很好的选择。由于 Java 的跨平台特性而且适合网络环境，因此，被视为开发服务器端应用程序的最理想的程序语言。

总之，在 Internet 的世界里，到处都可以看到 Java 的身影。

练习一

1. Java 有那些特点?
2. 编译后的 Java 程序能直接运行吗?
3. Java 为什么能跨操作系统平台运行?

原书空白页



第 2 章

Java 的开发工具

- 2.1 什么是开发工具
- 2.2 Java 开发工具的分类
- 2.3 Java 虚拟机
- 2.4 Java 开发工具包(JDK)
- 2.5 安装和设置 JDK
- 2.6 使用 JDK 编译、运行 Java 程序

通过第 1 章的介绍,我们已经了解到 Java 程序设计语言的优点,是不是很让人心动?心动不如行动,那么,怎样开发自己的程序呢?在进行开发之前,首先需要有一个开发的环境。像 C++, 可以用 C++ Builder、也可以用 Visual C++, 当然还有其他工具,Java 也有很多的开发工具。

2.1 什么是开发工具

就像天上不会掉馅饼一样,程序不会凭空跑到你的机器上去。我们要开发程序,首先,要把你的程序输入到计算机里面,要对程序进行编辑,并能保存为文件,因而需要一个能编辑文本的环境或工具。

有了编辑工具还不够,因为编辑的源程序代码是不能执行的。源代码需要被编译后才能解释执行,所以,编译器是必不可少的。

第 1 章中提到过,Java 是一种解释性语言。经 Java 编译器编译后的文件为字节码,不是机器码,因而不能直接执行。要看到程序的执行结果,就要靠 Java 虚拟机来解释执行了。

如果程序运行的结果并不是所期望的,那么就需要调试程序来排除错误。像微软的 Visual Studio 集成开发环境,它的调试功能相当强大,可以很方便地跟踪调试。在这点上,Java 的调试功能可就不尽人意了,需要一些编程经验来弥补。

如果上述条件都具备了,那么,就可以着手开发自己的程序了。说到这里,应该明白什么是开发工具了吧,说穿了,就是编辑、编译、调试以及运行程序所需的一整套环境或工具。

2.2 Java 开发工具的分类

Java 的开发工具有很多,大致可以分为以下几类:

核心开发工具

Java 的核心开发工具指的是 Java 的 JDK (Java Development Kit),它是 Java 开发工具的核心。任何其他开发工具都需要用到 JDK。

集成开发工具

为了方便 Java 程序的开发,一些著名公司推出了自己的 Java 集成开发工具,如:Java WorkShop, Borland JBuilder, Oracle9i JDeveloper, Symantec Cafe 等,这些工具提供了 Java 软件开发的一整套环境,集成了编码、编译、调试到运行所需的环境,而且有着良好的可视化效果,不过,这些工具对机器的配置要求较高。

其他工具

除以上介绍的开发工具外,还有一些其他的开发工具,比如:TextPad, KawaPro 等,这些工具比较小巧,借助于 JDK 软件包,加上自己的编辑器,比较适合于系统配置较低的机器。

2.3 Java 虚拟机

Java 虚拟机是一个假想中的机器,在实际的计算机上通过软件模拟来实现计算功能。Java 虚拟机有自己想像中的硬件,如处理器、堆栈、寄存器等,还具有相应的指令系统。

Java 有一个非常重要的特点:平台无关性。实现这一特点的关键便是 Java 虚拟机。一般的高级语言如果要在不同的平台上运行,需要编译成不同的目标代码。Java 引入虚拟机后,结果只需生成能在 Java 虚拟机上执行的字节码即可,由虚拟机解释执行。由于 Java 虚拟机屏蔽了操作系统平台的相关信息,编译后的字节码能够在不同的操作系统平台上运行,并由虚拟机将其解释成具体平台的机器指令来执行。

Java 虚拟机由 5 个部分组成:一组指令集、一组寄存器、一个栈、一个无用单元收集堆 (Garbage-collected-heap) 和一个方法区域。这 5 部分是 Java 虚拟机的逻辑成份,但它们的功能必须在实际的操作系统上以某种方式实现。

Java 虚拟机是 Java 底层实现的基础,如果你需要在特定的操作系统上实现 Java 虚拟机或用硬件芯片来实现 Java 虚拟机,那你就必须对 Java 虚拟机的规范有深刻地理解。不过,对于一般的 Java 程序员,只要大致了解就可以了。

2.4 Java 开发工具包(JDK)

前面说过, JDK 是 Java 开发工具的核心工具。JDK 包括所有编译、运行 Java 程序所需要的工具: Java 基本组件、库、Java 编译器、Java 解释器、小应用程序浏览器等。这些工具基本上都在 JDK 安装目录下的 bin 子目录里,以下为一些基本的工具:

Java 编译器: javac.exe

Java 编译器将以“.java”为扩展名的 Java 源文件编译成类文件(扩展名为:“.class”)。

语法为: javac [选项] 源程序文件名。如: `Javac HelloWorld.java`

以下为 Java 编译器选项及其说明:

- classpath<path> 覆盖缺省的 CLASSPATH 环境变量并指定新的 Java 类搜索路径;
- d <directory> 指定存放生成的类文件的目录,注意使用绝对路径;
- g 使用该选项可以用类文件生成调试表,供 jdb 使用;
- nowarn 关闭编译器的警告输出;
- O 打开优化处理;
- verbose 打开冗余编译;
- depend 使用该选项使得编译器考虑重新编译从别的类文件中引用的类文件;
- J option 该选项用于向实际运行该程序的 Java 解释器传递一个单独的变量。