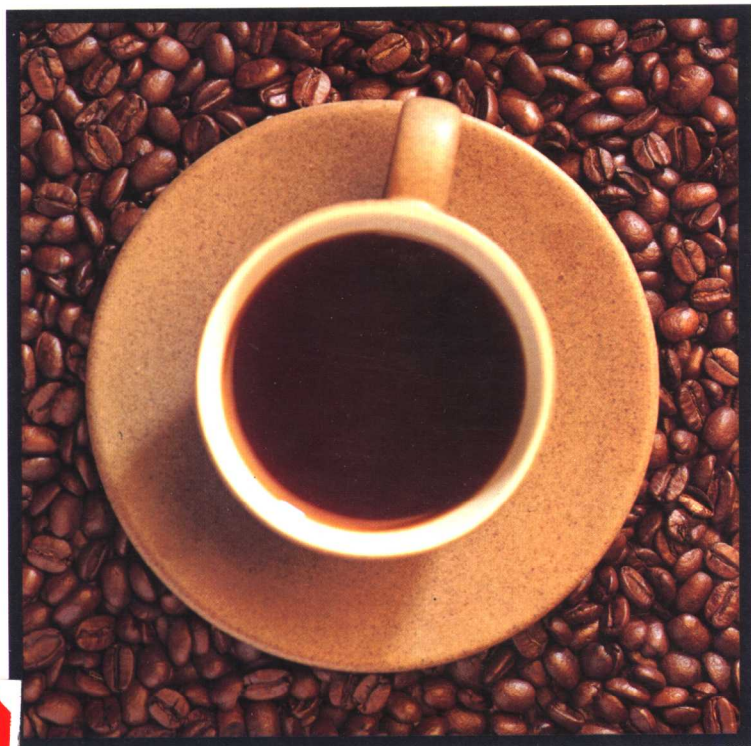


学习经典范例

JAVA 2

捷径教程

▶▶介绍最新的J2SDK



详尽讲解Java的基本语法

探讨AWT组件和Swing组件的用法，以及Java语言与网络、数据库的关系

深入剖析近200个生动实例的完整源代码

河温钊 编著



清华大学出版社

Java 2 捷径教程

柯温钊 编著

清华大学出版社

北京

内 容 提 要

Java 是目前最受欢迎的程序设计语言，在教学与开发中的应用都很广泛。

本书通过近 200 个完整、实用的实例来生动讲解 Java 的语言特性与编程方法，实例覆盖面广，针对性强，每个实例均给出了完整的源代码和相应的剖析，力求让读者透彻理解语言的应用。本书的每一章都提供针对本章内容的习题，并在书末给出习题详解，有助于读者巩固所学的知识。在重点讲解 Java 的基本语法的基础上，还探讨了 AWT 组件和 Swing 组件的用法，并介绍 Java 语言与网络、数据库开发的关系。因此，借助本书，读者不仅可以轻松掌握 Java 语言的精要，还将掌握 Java 编程技能，并用于实际开发中。

本书可作为大专院校相关课程教材，也可作为广大编程爱好者提高编程水平的自学教材，还可供程序开发人员作为编程参考。

版权所有，盗版必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目 (CIP) 数据

Java 2 捷径教程/柯温钊编著. —北京: 清华大学出版社, 2004.7

ISBN 7-302-08917-5

I. J… II. 柯… III. JAVA 语言—程序设计—教材

IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 060934 号

出 版 者: 清华大学出版社

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

社总机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 夏非彼.

文稿编辑: 朱起飞

封面设计: 林陶

版式设计: 科海

印 刷 者: 北京科普瑞印刷有限责任公司

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 24.5 字数: 596 千字

版 次: 2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷

书 号: ISBN 7-302-08917-5/TP·6315

印 数: 1~5000

定 价: 38.00 元

(如有印装质量问题, 我社负责调换)

序

1995年，Sun Microsystems公司正式发布Java语言，Java语言的出现使得网络中的大量难题迎刃而解。笔者在撰写本书时，由于Java语言的内容相当丰富，有欲罢不能的感觉，如线程、AWT组件、Swing组件、异常处理、数据库及Java语言与网络的关系等，再加上Java开发工具可以从网上免费下载，Java语言必将成为未来的主流。从1995年推出以来，Java语言历经多次更新，每次推出新版本，Java的功能就更上一层楼。目前Java 2的最新版本为Java 2 Platform, Standard Edition (J2SE) 1.5 Beta 2版。但由于大多Java程序员还是使用1.4.2版，所以本书内容及程序都以J2SE 1.4.2_4版本为蓝本，程序均在J2SE 1.4.2_4中调试过，请读者特别注意。

Java中的类有很多种，笔者选择比较重要且常用的内容加以介绍。本书除了重点讲解Java的基本语法外，也特别强调AWT组件和Swing组件的用法，另外还有Java语言与网络、数据库的关系。希望读者在读完本书后，能在因特网上一展身手。

本书共有16章，第1~10章为Java基本语法的介绍；第11章与第12章为AWT组件和Swing组件的介绍；第13章介绍Java事件，事件与AWT组件和Swing组件有非常密切的关系；第14章介绍绘图；第15章讲解网络；第16章涉及数据库。希望这些内容能对您有所帮助。

本书承蒙上奇公司的萧进贤先生、徐瑞珠小姐、庄吴行世先生的鼎力相助，提供许多的宝贵意见，使得本书如期付印，在此特别感谢他们。

最后，笔者深感Java语言包容甚广，欲窥其全貌，谈何容易！再加上个人的天赋、学识和取材的局限，相信还有许多地方不尽如人意，有不当之处还请大家多多指教。

柯温钊 于宜兰
2004年4月5日

目 录

第 1 章 如何编写您的第一个 Java 程序 1	第 4 章 Java 运算符 40
1.1 第一个 Java 程序 1	4.1 算术运算符..... 40
1.2 编写 Java 程序应注意的事项..... 3	4.2 关系运算符..... 43
1.3 Java 开发环境 4	4.3 逻辑运算符..... 44
1.4 习题 9	4.3.1 NOT (非) 44
第 2 章 Java 程序结构 10	4.3.2 AND (与) 45
2.1 Java 程序种类 10	4.3.3 OR (或) 46
2.2 Java Application..... 10	4.4 条件运算符..... 47
2.3 Java Applet..... 12	4.5 位运算符..... 49
2.4 习题 14	4.6 赋值运算符..... 51
第 3 章 Java 数据类型 15	4.7 类型转换运算符..... 53
3.1 整数类型 15	4.8 运算符的优先级..... 53
3.1.1 byte 15	4.9 习题 54
3.1.2 short 16	第 5 章 Java 选择语句 56
3.1.3 int 17	5.1 条件选择语句..... 56
3.1.4 long 18	5.1.1 关系表达式..... 57
3.1.5 char 19	5.1.2 算术表达式..... 58
3.2 浮点数类型 21	5.1.3 逻辑表达式..... 59
3.2.1 float..... 21	5.2 continue 跳转语句..... 60
3.2.2 double..... 21	5.3 多重条件选择语句..... 61
3.3 布尔类型 22	5.3.1 if (~else if) ~else 61
3.4 数组类型 23	5.3.2 switch..... 63
3.4.1 一维数组 23	5.4 习题 66
3.4.2 二维数组 27	第 6 章 Java 循环语句 68
3.4.3 魔术方块 28	6.1 for()..... 68
3.4.4 Arrays 类..... 31	6.2 while()..... 72
3.4.5 Object 类..... 33	6.3 do~while()..... 73
3.4.6 Array 类..... 35	6.4 跳转语句..... 76
3.4.7 ArrayList 类..... 36	6.4.1 break..... 76
3.5 习题 38	6.4.2 return 77
	6.5 递归方法..... 78

6.6 习题	81	10.2 接口	144
第 7 章 Java 字符与字符串	83	10.2.1 接口实现	146
7.1 字符	83	10.2.2 接口继承	147
7.2 字符串	88	10.3 习题	150
7.2.1 String 类	91	第 11 章 AWT 窗口组件	151
7.2.2 字符串转换	98	11.1 TextComponent 类	151
7.2.3 StringBuffer 类	99	11.1.1 TextField 类	153
7.2.4 StringTokenizer 类	104	11.1.2 TextArea 类	157
7.3 习题	106	11.2 Label 类	160
第 8 章 Java 异常处理	107	11.3 Button 类	162
8.1 异常处理类	107	11.4 Checkbox 类	166
8.1.1 运行时的异常处理	107	11.5 CheckboxGroup 类	170
8.1.2 检查异常处理	108	11.6 Choice 类	176
8.2 try...catch 语句	108	11.7 List 类	182
8.3 throw 及 throws	109	11.8 Scrollbar 类	188
8.4 finally	111	11.9 菜单类	192
8.5 Error 类	113	11.9.1 MenuComponent 类	193
8.5.1 LinkageError	113	11.9.2 MenuBar 类	194
8.5.2 VirtualMachineError	114	11.9.3 MenuItem 类	194
8.6 习题	116	11.9.4 CheckboxMenuItem 类	200
第 9 章 线程	117	11.9.5 Menu 类	203
9.1 Thread 类	118	11.9.6 PopupMenu 类	204
9.2 Runnable 接口	121	11.9.7 MenuShortcut 类	207
9.3 线程的生命周期	123	11.10 Container 类	210
9.4 ThreadGroup 类	125	11.11 Window 类	213
9.5 习题	128	11.12 Dialog 类	215
第 10 章 面向对象程序设计	129	11.13 Frame 类	221
10.1 类	129	11.14 布局管理器	224
10.1.1 参数传递	134	11.14.1 BorderLayout 类	224
10.1.2 命令行参数	136	11.14.2 CardLayout 类	227
10.1.3 继承	136	11.14.3 FlowLayout 类	230
10.1.4 super	138	11.14.4 GridLayout 类	232
10.1.5 覆盖	139	11.14.5 GridBagLayout 类	234
10.1.6 多态	140	11.14.6 GridBagContainers 类	235
10.1.7 this	142	11.15 习题	238
		第 12 章 Swing 窗口组件	240
		12.1 JButton 类	240

12.2 JLabel 类..... 241	14.4 Color 类..... 310
12.3 JTextField 类..... 243	14.5 Font 类..... 313
12.4 JTextArea 类..... 247	14.6 Canvas 类..... 316
12.5 JCheckBox 类..... 249	14.7 习题..... 319
12.6 JRadioButton 类..... 250	第 15 章 Java 与网络..... 320
12.7 JComboBox 类..... 254	15.1 InetAddress 类..... 320
12.8 JMenuBar 类..... 257	15.2 URL 类..... 324
12.9 JMenuItem 类..... 258	15.3 URLConnection 类..... 327
12.10 JMenu 类..... 260	15.4 Socket 类..... 332
12.11 JCheckBoxMenuItem 类..... 263	15.5 ServerSocket 类..... 335
12.12 JSlider 类..... 267	15.6 习题..... 340
12.13 JTable 类..... 270	第 16 章 Java 与数据库..... 341
12.14 习题..... 275	16.1 使用 Microsoft Access
第 13 章 Java 事件..... 276	创建一个数据库..... 341
13.1 ActionEvent 类..... 278	16.2 ODBC 的设置..... 345
13.2 AdjustmentEvent 类..... 279	16.3 Java 程序连接数据库..... 348
13.3 ComponentEvent 类..... 280	16.4 SQL 语法..... 354
13.4 ContainerEvent 类..... 281	16.4.1 SELECT 语句..... 354
13.5 FocusEvent 类..... 282	16.4.2 INSERT 语句..... 361
13.6 InputEvent 类..... 283	16.4.3 UPDATE 语句..... 362
13.7 ItemEvent 类..... 285	16.4.4 DELETE 语句..... 364
13.8 KeyEvent 类..... 286	16.5 习题..... 365
13.9 MouseEvent 类..... 288	附录 A 如何获得 Java 2 Platform
13.10 PaintEvent 类..... 292	Standard Edition..... 367
13.11 TextEvent 类..... 293	附录 B 如何安装 Java 2 Platform
13.12 WindowEvent 类..... 294	Standard Edition..... 372
13.13 习题..... 295	B.1 安装 Java 2 Platform
第 14 章 绘图..... 296	Standard Edition..... 372
14.1 Graphics 类..... 296	B.2 如何设定 Java 环境变量..... 375
14.2 Graphics2D 类..... 302	附录 C 习题解答..... 377
14.3 Math 类..... 308	

第 1 章

如何编写您的第一个 Java 程序

Java 语言到底长得像什么样子？如何编写 Java 程序？又如何运行 Java 程序？也许您对此充满好奇和疑惑，笔者将为您解开各种疑惑，请听我慢慢道来……

1.1 第一个 Java 程序

下面是本书的第一个 Java 程序：

```
public class kol_1
{
    public static void main(String args[ ])
    {
        System.out.println("欢迎来到 Java 世界!!");
    }
}
```

上述程序到底表示什么意思呢？先别急，这是属于程序结构的部分，将在第 2 章详细介绍。

一般，Java 源代码的编写是利用文本编辑软件完成的，如 Word、记事本等。图 1-1 所示是使用记事本编写的 Java 程序代码。

在程序代码的输入过程中，要注意随时保存输入的内容，方法如下：

第一步：选择“文件”|“另存为”命令，如图 1-2 所示。

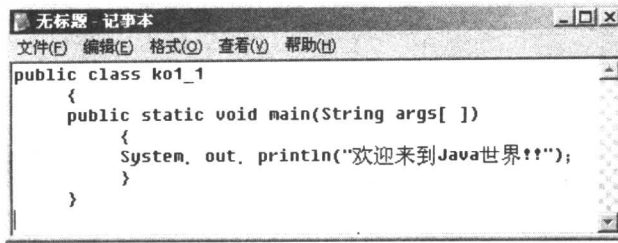


图 1-1

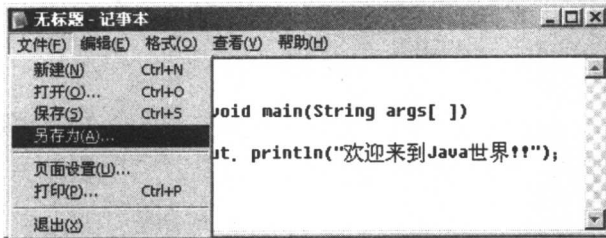


图 1-2

第二步：选择“保存在”目录为“Java 2004”，并在“保存类型”下拉列表框中选择“所有文件”选项，在“文件名”文本框中输入“ko1_1.java”，如图 1-3 所示。最后单击“保存”按钮。

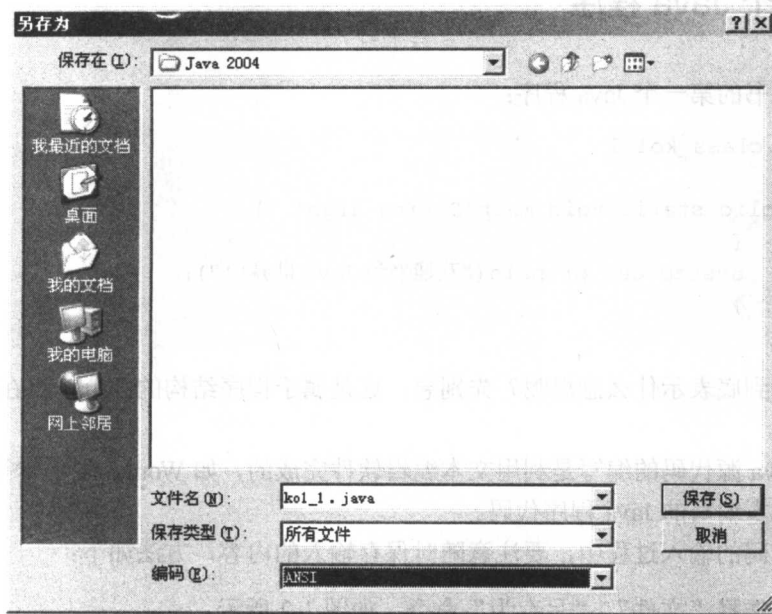


图 1-3

保存后，就可以执行“ko1_1.java”程序。方法如下：

第一步：从“开始”菜单中选择“程序”|“附件”|“命令提示符”命令，打开命令提示符窗口，如图 1-4 所示。

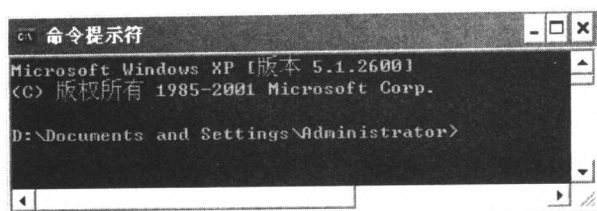


图 1-4

第二步：输入“cd \Java 2004”，并按 Enter 键。

第三步：此时屏幕上出现 C:\Java 2004>，输入 javac kol_1.java，按 Enter 键开始编译程序。

第四步：如果一切 OK，屏幕上再次出现 C:\Java 2004>，输入 java kol_1，按 Enter 键后，将显示“欢迎来到 Java 世界!!”字符串。如图 1-5 所示。

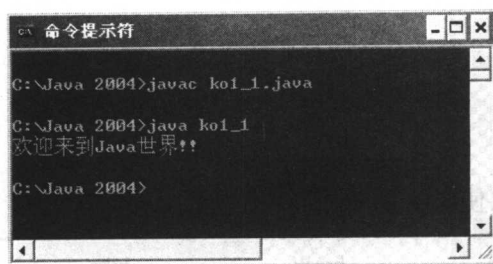


图 1-5

1.2 编写 Java 程序应注意的事项

编写程序时的不成文规定：

第一，编写的程序必须要有层次感。如 kol_1.java 程序：

```
public class kol_1
{
    public static void main(String args[ ])
    {
        System.out.println("欢迎来到 Java 世界!!");
    }
}
```

如果将 kol_1.java 写成：

```
public class kol_1
{
    public static void main(String args[ ])
    {
        System.out.println("欢迎来到 Java 世界!!");
    }
}
```

请比较一下两者是否不一样。两者都能编译运行，但当程序代码比较长时，第二种程序显然不易于阅读，如果程序有问题，也会造成纠错困难。这里建议大家，刚学习程序设计时必须养成第一种编码习惯。

第二，英文字母大小写代表不同的意义。例如，`int total` 与 `int Total` 代表的是两个不同的变量。

第三，使用标识符的原则。标识符 (Identifier) 是用户用来定义变量、数据类型 (data types)、方法名的。标识符的第一个字母必须是英文字母、下划线或\$符号，不可使用其他特殊符号，如!`@# %^&*等。`

正确的标识符	错误的标识符
<code>ko2</code>	<code>2ko</code>
<code>good</code>	<code>%good</code>
<code>_abc</code>	<code>&_abc</code>
<code>ko_pay</code>	<code>ko%pay</code>
<code>\$123</code>	<code>%123</code>
<code>p\$</code>	<code>P#</code>

第四，标识符不可以使用保留字。下表为保留字一览表。

<code>abstract</code>	<code>boolean</code>	<code>break</code>	<code>byte</code>	<code>case</code>
<code>catch</code>	<code>char</code>	<code>class</code>	<code>const</code>	<code>continue</code>
<code>default</code>	<code>do</code>	<code>double</code>	<code>else</code>	<code>extends</code>
<code>final</code>	<code>finally</code>	<code>float</code>	<code>for</code>	<code>goto</code>
<code>if</code>	<code>implements</code>	<code>import</code>	<code>instanceof</code>	<code>int</code>
<code>interface</code>	<code>long</code>	<code>native</code>	<code>new</code>	<code>package</code>
<code>private</code>	<code>protected</code>	<code>public</code>	<code>return</code>	<code>short</code>
<code>static</code>	<code>strictfp</code>	<code>super</code>	<code>switch</code>	<code>synchronized</code>
<code>throw</code>	<code>throws</code>	<code>this</code>	<code>transient</code>	<code>try</code>
<code>void</code>	<code>volatile</code>	<code>while</code>		

1.3 Java 开发环境

当安装好 Java 软件后，在 `j2sdk1.4.2` 目录中有一些子目录及文件，其中 `j2sdk1.4.2\bin` 子目录中放置的是 JDK (Java Developer's Kit) 工具，如图 1-6 所示。

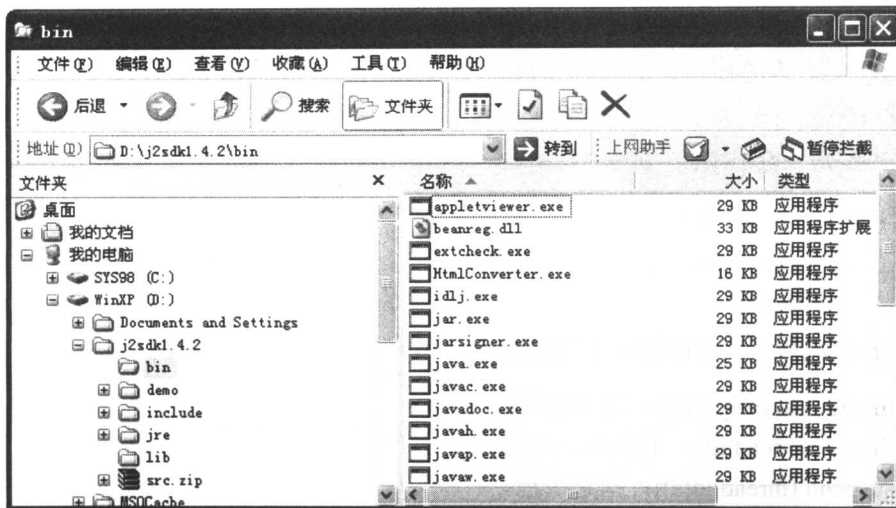


图 1-6

JDK 工具包的内容:

1. javac, Java 语言的编译器

这是 Java 语言最重要的 JDK 工具之一, 用来将 Java 源代码编译成字节码。假设有一个 Java 源代码文件 ko2_1.java, 现在使用 javac 编译器编译它, 方法如下:

```
javac <选项> ko2_1.java
```

其中, <选项>可有可无, ko2_1.java 为源程序代码。

2. java, Java 语言的解释器

java 指令用来运行 javac 编译的字节码, 使用方法如下:

```
java <选项> ko2_1
```

其中, <选项>请参考编译器 javac 的说明。

3. jdb, Java 语言的调试器

jdb 是一个调试程序, 用于调试编译的类。使用方法如下:

```
jdb <选项> <类> <参数>
```

下面来看一看<选项>:

-help	打印帮助信息并退出
-version	打印建立的版本并退出
-host <hostname>	依附解释器的主机名
-password <psswd>	依附解释器的密码
-dbgtrace	打印调试信息

<类>是指开始调试的类名称。

<参数>是指传递给 main() 方法的参数。

例如：

```
C: \Java 2004>jdb -dbgtrace kol_1
```

按 Enter 键后，出现下列信息：

```
Initializing jdb...  
>_ ←jdb 提示符
```

在 jdb 提示符下，可以输入下列指令：

- threads [threadgroup]：列出线程。
- thread <thread id>：设置默认线程。
- suspend [thread id(s)]：挂起线程。
- resume [thread id(s)]：恢复线程。
- where [thread id] | all：倾倒线程堆栈。
- wherei [thread id] | all：倾倒线程堆栈及计算机信息。
- threadgroups：列出线程群组。
- threadgroup <name>：设置当前线程群组。
- print <id> [id(s)]：打印对象或数据成员。
- dump <id> [id(s)]：打印所有对象信息。
- locals：列出当前堆栈所有局部变量 (local variables)。
- classes：列出当前已知类。
- methods <class id>：列出类的方法。
- stop in <class id>.<method>[(argument_type, ...)]：在方法中设置断点。
- stop at <class id>:<line>：在行中设置断点。
- up [n frames]：向上移线程堆栈。
- down [n frames]：向下移线程堆栈。
- clear <class id>.<method>[(argument_type, ...)]：清除方法中的断点。
- clear <class id>:<line>：清除行中的断点。
- step：执行当前的行。
- step up：执行到当前方法返回。
- stepi：执行当前的指令。
- next：下一行。
- cont：从断点继续执行。
- catch <class id>：中止指定异常。
- ignore <class id>：忽略指定异常。
- list [line number|method]：打印源代码。
- use [source file path]：显示或改变源文件路径。
- memory：列出总共的内存和当前没被使用的内存。

- gc: 释放没有用的对象。
- load classname: 载入调试的 Java 类。
- run <class> [args]: 加载的 Java 类开始执行。
- !! : 重复最后指令。
- help (or ?): 列出指令。
- exit (or quit): 退出 jdb 系统, 回到 DOS。

例如:

```
C: \ Java 2004 >jdb      ←输入 jdb
Initializing jdb...
> run kol_1      ←输入 run kol_1
run kol_1
Set uncaught java.lang.Throwable
Set deferred uncaught java.lang.Throwable
>
VM Started: 欢迎来到 Java 世界!!
The application exited
```

4. appletviewer, Java Applet 的浏览器

前面讲过, javac 编译 Java 程序代码成为字节码, 然后, 使用 java 执行字节码类, 这种程序称为 Java Application。如果在 javac 编译 Java 程序代码成为字节码后, 再建立一个 HTML 文件, 使用 appletviewer 执行此 HTML 文件, 这种程序称为 Java Applet。请参考第 2 章关于 Java Applet 的说明。

5. javadoc, Java API 文档管理器

javadoc 可以产生一组 HTML 格式的相关文件来说明编写的 Java 程序代码。例如:

```
C: \ Java 2004 >javadoc kol_1.java
```

按 Enter 键后, 产生如下的信息:

```
Loading source file kol_1.java...
Constructing Javadoc information...
Standard Doclet version 1.4.1

Generating constant-values.html..
Building tree for all the packages and classes...
Building index for all the packages and classes...
Generating overview-tree.html...      ←生成文件
Generating index-all.html...         ←生成文件
Generating deprecated-list.html...    ←生成文件
Building index for all classes...
Generating allclasses-frame.html...   ←生成文件
Generating allclasses-noframe.html... ←生成文件
Generating index.html...              ←生成文件
Generating packages.html...           ←生成文件
Generating kol_1.html...               ←生成文件
```

Generating package-list... ←生成文件
Generating help-doc.html... ←生成文件
Generating stylesheet.css... ←生成文件

6. javah, C/C++ 头文件生成器

javah 用来生成 C/C++ 头文件。javah 用法如下:

```
javah [options] <classes>
```

[options] 包括的指令见下表:

-help	打印 javah 的帮助信息
-classpath <path>	加载类的路径
-d <dir>	输出目录
-o <file>	输出文件
-jni	创建 JNI 样式头文件
-old	创建 JDK1.0 样式头文件
-stubs	创建定义文件
-version	打印版本信息
-verbose	显示输出信息
-force	写入输出文件

7. javap, Java 语言的反编译器

javap(javap_g) 用来反编译 Java 类文件。其用法如下:

```
javap <options> <classes>...
```

<options> 包括的指令见下表:

-c	反编译码
-classpath <pathlist>	搜寻用户类文件
-help	打印 javap 使用方法信息
-J<flag>	直接传递 <flag> 到操作系统
-l	打印行号及局部变量表
-public	只显示 public 类与成员
-protected	显示 protected/public 类与成员
-package	显示 package/protected/public 类与成员
-private	显示所有类与成员
-s	打印内部类型识别标志
-verbose	打印堆栈大小, 方法的参数数量

例如:

```
C: \ Java 2004 >javap -c kol_1
```

按下 Enter 键后, 出现下列信息:

```

Compiled from kol_1.java
public class kol_1 extends java.lang.Object {
    public kol_1( );
    public static void main(java.lang.String[ ]);
}

Method kol_1( )
  0 aload_0
  1 invokespecial #1 <Method java.lang.Object( )>
  4 return

Method void main(java.lang.String[ ])
  0 getstatic #2 <Field java.io.PrintStream out>
  3 ldc #3 <String "欢迎来到 Java 世界!!">
  5 invokevirtual #4 <Method void println(java.lang.String)>
  8 return
    
```

1.4 习题

1. Java 语言的源代码是使用何种软件编写的？
2. 下列变量是否为同一个变量？

sum1 SUM1 Sum1

3. 下列哪些是不合法的标识符？

(a) _8100 (b) 3.14 (c) @abc (d) a□b (e) cat-dog (f) cat
 (g) dog\$ (h) ko_pay (i) p12 (j) a%s (k) int (l) _ace
 (m) T (n) t (o) PI (p) no3 (q) A+B (r) float

4. 连线题：

(a) javap	(1) Java Applet 的浏览器
(b) appletviewer	(2) Java 语言的解释器
(c) javac	(3) Java 语言的调试程序
(d) jdb	(4) Java API 文档管理器
(e) java	(5) Java 语言的反编译器
(f) javadoc	(6) Java 语言的编译器

第 2 章

Java 程序结构

2.1 Java 程序种类

Java 程序分为两种：Java Application 和 Java Applet。Java Application 必须在所处的操作系统下执行，而 Java Applet 必须与超文本标记语言（Hypertext Markup Language, HTML）配合使用，并在浏览器上执行。

2.2 Java Application

Java Application 程序可以在 DOS、UNIX 等操作系统中执行。下面是一个简单的 Java Application 程序：

```
1   public class ko2_1
2   {
3       public static void main(String args[ ])
4       {
5           int a=3, b=4, c;
6           c=a+b;
7           System.out.println("c="+c);
8       }
9   }
```

声明：ko2_1 程序加入 1, 2, ..., 9 编号是为了方便程序解说，一般编写程序时不需要加入编号。另外，这里使用全角的标点符号是为了整体版面美观，