新世纪

高职高专
软件专业系列规划教材

Java语言程序设计实用教程（第三版）

新世纪高职高专教材编审委员会组编
赵从军 编著

大连理工大学出版社

新世纪

高职高专软件专业系列规划教材

Java 语言程序设计实用教程

（第三版）

新世纪高职高专教材编审委员会组编

赵从军　编著

JAVA YUYAN CHENGXU SHEJI SHIYONG JIAOCHENG

大连理工大学出版社
DALIAN UNIVERSITY OF TECHNOLOGY PRESS

图书在版编目(CIP)数据

Java 语言程序设计实用教程 / 赵从军编著. —3 版. —大连:大连理工大学出版社,2008.3
高职高专软件专业系列规划教材
ISBN 978-7-5611-3933-2

Ⅰ.J… Ⅱ.赵… Ⅲ.JAVA 语言—程序设计—高等学校:技术学校—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2008)第 025937 号

大连理工大学出版社出版
地址:大连市软件园路 80 号 邮政编码:116023
电话:0411-84708842 邮购:0411-84703636 传真:0411-84701466
E-mail:dutp@dutp.cn URL:http://www.dutp.cn
大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:185mm×260mm 印张:26.25 字数:588 千字
印数:24001～25000
2002 年 2 月第 1 版 2008 年 3 月第 3 版
2008 年 3 月第 4 次印刷

责任编辑:潘弘喆 彭静静 责任校对:张 艳
封面设计:苏儒光

ISBN 978-7-5611-3933-2 定 价:35.00 元

总　序

我们已经进入了一个新的充满机遇与挑战的时代，我们已经跨入了21世纪的门槛。

20世纪与21世纪之交的中国，高等教育体制正经历着一场缓慢而深刻的革命，我们正在对传统的普通高等教育的培养目标与社会发展的现实需要不相适应的现状作历史性的反思与变革的尝试。

20世纪最后的几年里，高等职业教育的迅速崛起，是影响高等教育体制变革的一件大事。在短短的几年时间里，普通中专教育、普通高专教育全面转轨，以高等职业教育为主导的各种形式的培养应用型人才的教育发展到与普通高等教育等量齐观的地步，其来势之迅猛，发人深思。

无论是正在缓慢变革着的普通高等教育，还是迅速推进着的培养应用型人才的高职教育，都向我们提出了一个同样的严肃问题：中国的高等教育为谁服务，是为教育发展自身，还是为包括教育在内的大千社会？答案肯定而且惟一，那就是教育也置身其中的现实社会。

由此又引发出高等教育的目的问题。既然教育必须服务于社会，它就必须按照不同领域的社会需要来完成自己的教育过程。换言之，教育资源必须按照社会划分的各个专业（行业）领域（岗位群）的需要实施配置，这就是我们长期以来明乎其理而疏于力行的学以致用问题，这就是我们长期以来未能给予足够关注的教育目的问题。

如所周知，整个社会由其发展所需要的不同部门构成，包括公共管理部门如国家机构、基础建设部门如教育研究机构和各种实业部门如工业部门、商业部门，等等。每一个部门又可作更为具体的划分，直至同它所需要的各种专门人才相对应。教育如果不能按照实际需要完成各种专门人才培养的目标，就不能很好地完成社会分工所赋予它的使命，而教育作为社会分工的一种独立存在就应受到质疑（在市场经济条件下尤其如此）。可以断言，按照社会的各种不同需要培养各种直接有用人才，是教育体制变革的终极目的。

随着教育体制变革的进一步深入，高等院校的设置是否会同社会对人才类型的不同需要一一对应，我们姑且不论。但高等教育走应用型人才培养的道路和走研究型（也是一种特殊应用）人才培养的道路，学生们根据自己的偏好各取所需，始终是一个理性运行的社会状态下高等教育正常发展的途径。

高等职业教育的崛起，既是高等教育体制变革的结果，也是高等教育体制变革的一个阶段性表征。它的进一步发展，必将极大地推进中国教育体制变革的进程。作为一种应用型人才培养的教育，它从专科层次起步，进而应用本科教育、应用硕士教育、应用博士教育……当应用型人才培养的渠道贯通之时，也许就是我们迎接中国教育体制变革的成功之日。从这一意义上说，高等职业教育的崛起，正是在为必然会取得最后成功的教育体制变革奠基。

高等职业教育还刚刚开始自己发展道路的探索过程，它要全面达到应用型人才培养的正常理性发展状态，直至可以和现存的（同时也正处在变革分化过程中的）研究型人才培养的教育并驾齐驱，还需假以时日；还需要政府教育主管部门的大力推进，需要人才需求市场的进一步完善发育，尤其需要高职高专教学单位及其直接相关部门肯于做长期的坚忍不拔的努力。新世纪高职高专教材编审委员会就是由全国100余所高职高专院校和出版单位组成的旨在以推动高职高专教材建设来推进高等职业教育这一变革过程的联盟共同体。

在宏观层面上，这个联盟始终会以推动高职高专教材的特色建设为己任，始终会从高职高专教学单位实际教学需要出发，以其对高职教育发展的前瞻性的总体把握，以其纵览全国高职高专教材市场需求的广阔视野，以其创新的理念与创新的运作模式，通过不断深化的教材建设过程，总结高职高专教学成果，探索高职高专教材建设规律。

在微观层面上，我们将充分依托众多高职高专院校联盟的互补优势和丰裕的人才资源优势，从每一个专业领域、每一种教材入手，突破传统的片面追求理论体系严整性的意识限制，努力凸现高职教育职业能力培养的本质特征，在不断构建特色教材建设体系的过程中，逐步形成自己的品牌优势。

新世纪高职高专教材编审委员会在推进高职高专教材建设事业的过程中，始终得到了各级教育主管部门以及各相关院校相关部门的热忱支持和积极参与，对此我们谨致深深谢意；也希望一切关注、参与高职教育发展的同道朋友，在共同推动高职教育发展、进而推动高等教育体制变革的进程中，和我们携手并肩，共同担负起这一具有开拓性挑战意义的历史重任。

新世纪高职高专教材编审委员会

2001年8月18日

前　言

Java 语言自从创建以来，已经风靡全世界，它不仅仅是一种编程语言，也是一种运行平台，渗透到了 IT 业的各个领域，无处不在，无所不能。

从编程角度，Java 是纯的面向对象的编程语言，它避免了 C++中指针操作的复杂性和内存泄漏之类的错误。它同时也是一种跨平台的编程语言，这是因为它针对的是各种操作系统上提供的虚拟机的统一接口，保证代码编写一次，到处运行。

在学习 Java 编程的过程中，需要掌握要领，对初学者来说，不仅仅学习 Java 语言的语法，更要重视 Java 的编程思想，这样才能提纲挈领，触类旁通。

全书的编排如下：

第 1 章　介绍 Java 的运行及开发环境，使学习者熟悉不同操作系统下的 JDK 的安装与配置，熟悉不同操作系统下 eclipse 的开发环境。为 Java 的进一步学习奠定基础。

第 2 章　介绍 Java 的基本知识和编程规范，使学习者遵照编程规范编写简单的 Java 程序。

第 3 章　介绍 Java 的类型和运算符，使学习者了解基本的 Java 知识及语法。

第 4 章　介绍控制结构，使学习者掌握 Java 编程的基本结构与流程。

第 5 章　介绍面向对象解决方案，使学习者掌握运用面向对象编程思想解决问题。

第 6 章　介绍接口与内部类，使学习者掌握其使用，并运用在项目中。

第 7 章　介绍 Java 的基本类库，使学习者掌握 Java API 的使用，理解其系统类的设计思想。

第 8 章　介绍 Java 的字符串的处理，使学习者掌握字符串的编程与处理。

新世纪

第9章 介绍Java的异常处理，使学习者掌握异常处理的设计思想，并运用在项目的编程中。

第10章 介绍数据结构与集合类，使学习者掌握其设计思想，并运用在程序设计中。

第11章 介绍输入与输出流，使学习者掌握其编程，进行文件和网络操作。

第12章 介绍多线程的编程机制，使学习者掌握其设计要领，并运用在项目编程中。

第13章 介绍图形界面的设计，使学习者掌握图形组件的设计和布局，以及事件处理的机制，并运用在项目设计中。

鉴于时间仓促和作者的水平有限，错误之处在所难免，敬请读者批评指正。

所有意见和建议请发往：gzjckfb@163.com

联系电话：0411-84707492 84706104

编 者

2008年3月

目　录

第一章 Java运行与开发环境

本章要点

- Java 引言
- Windows 和 Linux 平台的 Java Development Kit(JDK)的开发环境与配置
- Eclipse IDE 的开发环境与编程

通过第一章的学习，读者能够了解 Java 的历史，掌握 Java 的开发工具 JDK 和 Java 的集成开发环境 Eclipse IDE，利用其开发一个简单的 Java 应用程序，获得对 Java 的初步认识。同时，介绍在 Windows 和 Linux 两种操作系统平台下的 JDK 与 EclipseIDE 的安装配置。

1.1 JDK 的下载、安装与配置

在介绍 Java 程序开发语言之前，有必要了解一下 Java 的历史。Java 在 20 世纪 90 年代诞生于 Sun MicroSystem 公司的一个控制电子设备的软件项目，负责该项目的是 Sun MicroSystem 的首席程序员 James Gosling。在项目的开始，Gosling 采用面向对象的编程语言 C++进行开发，但在进行中发现很多问题都出自于 C++的复杂性，尤其是指针错误和内存泄漏之类的程序错误，因此，Gosling 决定创建一个能够克服 C++程序语言缺陷的编程语言在该项目中代替之。Gosling 沿用了 C++的基本语法和面向对象的特点，剔除了基于指针的编程内容，命名这种新的编程语言为 Oak。

后来，由于互联网的飞速发展，当时在浏览器中看到的页面都是静态的，Sun 公司利用 Oak 可以创建动态页面，并对 Oak 进行了较大的改造，将其命名为 Java，伴随互联网获得了大量的应用和发展，目前的主流浏览器都支持 Java 的运行。

Java 语言是真正的基于面向对象的编程思想，虽然大部分语法与 C++相似，但舍弃了其中面向过程的成分和指针的概念。学习 Java 语言，首先从 Java 的开发工具和运行环境开始，Sun 公司提供了最原始的 Java 开发工具 JDK(Java Development Kit)和 Java 运行环境 JRE(Java Runtime Enviroment)。

Java 语言是一种与操作系统平台无关的编程语言，这是因为 Java 的运行环境 JRE 提供了一种称为 Java 虚拟机(Java Virtual Machine)的应用程序，简称 JVM，Java 程序实际运行在 JVM 上，而 JVM 可以有各种操作系统的版本，例如有 Windows 版、Solaris 版、SCO UNIX 版及 Linux 版等操作系统的 JVM。这样，Java 程序只与 JVM 交互，而 JVM 屏蔽掉所有安装其上的操作系统。Java 程序面向 JVM 的统一接口，只需要编写一次代码，就可以在任意操作系统的 JVM 上运行。

Java 程序是如何在 JVM 上运行的呢？首先，Java 程序通过 JDK 提供的编译程序将

其编译为紧凑的字节代码(bytecode),然后通过 JDK 提供的运行程序将编译后的字节代码调入 JVM,由 JVM 读取这些字节代码并解释运行。由于解释运行比较缓慢,Java 2 引入了 Just In Time 编译器,简称 JIT 编译器,JIT 内置于 JVM 中。这样,Java 程序编译后的字节代码,不同的部分执行运行时检查,JIT 编译器按部分读取字节代码,并将其编译成本地操作系统所在的机器语言,Java 程序以更快的速度运行。

Java 现在不仅是编程语言,还是一个开发平台,Java 技术给程序员提供了许多工具,如编译器、解释器、文档生成器和文件打包工具等等。同时 Java 还是一个程序发布平台,有两种主要的"发布环境"。首先 Java 运行时环境 JRE 包含了完整的类文件包;其次主流浏览器都提供了 Java 解释器和运行时环境。目前 Sun 公司把 Java 平台划分成 J2EE、J2SE、J2ME 三个平台,针对不同的项目。J2EE 是 Java 2 Enterprise Edition,主要目的是为企业级应用提供一个应用服务器的运行和开发平台,J2EE 本身是一个开放标准,任何软件厂商和组织都可以推出自己的符合 J2EE 标准的产品,以便用户可以有多种选择。J2SE 是 Java 2 Standard Edition,主要是为台式机和工作站提供一个开发和运行的平台;学习 Java,首先从 J2SE 开始,也就是从其开发工具 JDK 开始,JDK 工具包含 JRE。J2ME 是 Java 2 Micro Edition,主要是面向嵌入式应用等电子产品,为电子产品提供一个 Java 的运行平台,使得 Java 程序能够在手机、机顶盒、PDA 等产品上运行。J2EE、J2SE 和 J2ME 的表示如图 1-1 所示。

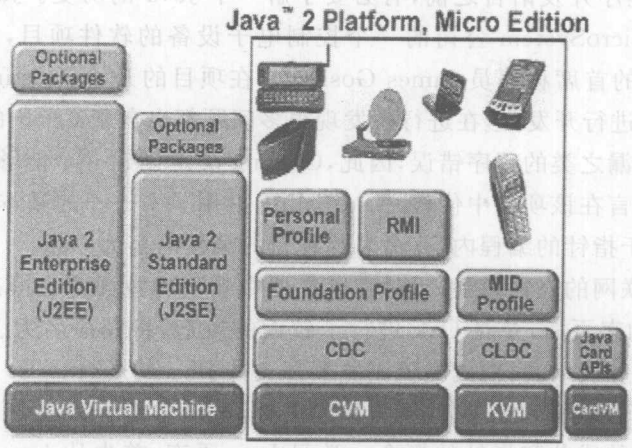

图 1-1 Java 2 Editions

本节以 Windows 和 Linux 两个平台为例,分别介绍 Windows 版和 Linux 版的 JDK 的下载、安装和配置。

1.1.1 JDK for Windows 的下载、安装和配置

首先访问网站 http://java.sun.com,在其上选择 Downloads 标题,其中有 Java SE、Java EE 和 Java ME 等项目,选择 Java SE,进入到 Java Standard Edition 的各种版本的下载页面。选择 JDK 5.0 Update 8 或更高版本,下载 jdk-1_5_0_08-windows-i586-p.exe 到 Windows 操作系统的一个目录下。其次执行 jdk-1_5_0_08-windows-i586-p.exe 文

件，安装JDK到C:\Program Files\Java\jdk1.5.0_08目录，其目录结构如下：

C:\Program Files\Java\jdk1.5.0_08

..\Java\jre1.5.0_08

在jdk1.5.0_08下目录结构为：

..\jdk1.5.0_08\bin

..\demo

..\docs

..\include

..\jre

..\lib

..\sample

最后，当完成安装后，需要配置JAVA_HOME、CLASSPATH和PATH环境变量，选择控制面板→系统→高级→环境变量→系统变量。按[新建]按钮，在弹出对话框的"变量名"栏输入'JAVA_HOME'，在"变量值"栏输入'C:\Program Files\Java\jdk1.5.0_08'，只输入单引号内的内容，以下环境变量值相同，按[确定]按钮完成JAVA_HOME环境变量的设置；重复前面的操作，在"变量名"栏输入'CLASSPATH'，在"变量值"栏输入'.;C:\Program Files\Java\jdk1.5.0_08\lib;C:\Program Files\Java\jdk1.5.0_08\jre\lib'完成CLASSPATH环境变量的设置；在系统变量中选择环境变量PATH，点击[编辑]按钮，在弹出对话框的"变量值"栏的末尾以';'分割添加'C:\Program Files\Java\jdk1.5.0_08\bin'，点击[确定]按钮完成PATH环境变量的修改。最终，点击[确定]按钮退出环境变量的设置。

至此，已经完成了Windows平台的JDK的下载、安装与配置，为了验证JDK的使用是否正常，切换到命令行界面，敲入Java命令，若出现图1-2所示的显示，说明JDK能够正确使用。

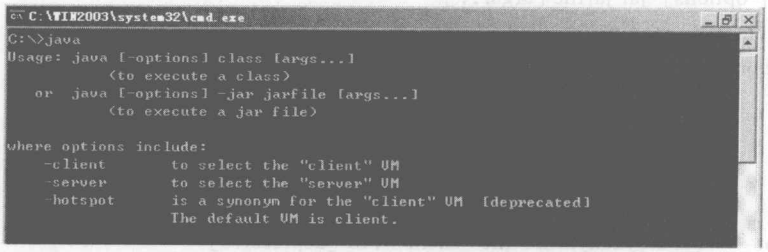

图1-2 Java命令

1.1.2 JDK for Linux的下载、安装与配置

首先在网站http://java.sun.com的Java Standard Edition的各种版本的下载页面，选择JDK 5.0 Update 8或更高版本，下载jdk-1_5_0_08-linux-i586-rpm.bin到Red Hat Linux操作系统的一个目录下，例如在/usr/local目录下。其次以命令cd /usr/local切换到当前目录下，以命令chmod u+x jdk-1_5_0_08-linux-i586-rpm.bin改变执行权

限。然后，以命令./ jdk-1_5_0_08-linux-i586-rpm.bin 执行，按照提示信息完成安装。其目录结构如下：

/usr/java/ jdk1.5.0_08

在 jdk1.5.0_08 下目录结构为：

../jdk1.5.0_08/bin
../demo
../include
../jre
../lib
../man
../sample

同样，当 JDK 安装完成后，需要配置 Linux 系统下的 JAVA_HOME、CLASSPATH 和 PATH 环境变量，以命令 cd ～ 切换到主目录，以命令 ls -ah 列出当前目录的文件，其中可以看到.bash_profile 文件，以命令 vi .bash_profile 编辑文件如下：

```
JAVA_HOME=/usr/java/jdk1.5.0_08
CLASSPATH=./:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
PATH=$JAVA_HOME/bin:$PATH:$HOME/bin
export JAVA_HOME CLASSPATH PATH
```

以上的 JAVA_HOME、CLASSPATH 和 PATH 分别表示 Linux 操作系统的当前用户的环境变量，并且 Linux 操作系统是多用户的系统，要想使得每个用户正确使用 Java 运行环境，则必须编辑每个用户下的.bash_profile 文件，其修改内容同上。然后注销当前用户，重新登录，切换到命令行界面，执行 Java 命令，则出现以下信息：

```
[username@localhost ~]$ java
Usage: java [-options] class [args...]
           (to execute a class)
   or java [-options] -jar jarfile [args...]
           (to execute a jar file)
where options include:
    -d32          use a 32-bit data model if available
    -d64          use a 64-bit data model if available
    -client       to select the "client" VM
    -server       to select the "server" VM
    -hotspot      is a synonym for the "client" VM [deprecated]
                  The default VM is client.
    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                  A ; separated list of directories, JAR archives,
                  and ZIP archives to search for class files.
    -D<name>=<value>
                  set a system property
    -verbose[:class|gc|jni]
                  enable verbose output
```

```
    -version          print product version and exit
    -version:<value>
                      require the specified version to run
    -showversion print product version and continue
    -jre-restrict-search | -jre-no-restrict-search
                      include/exclude user private JREs in the version search
    -? -help          print this help message
    -X                print help on non-standard options
    -ea[:<packagename>... |:<classname>]
    -enableassertions[:<packagename>... |:<classname>]
                      enable assertions
    -da[:<packagename>... |:<classname>]
    -disableassertions[:<packagename>... |:<classname>]
                      disable assertions
    -esa | -enablesystemassertions
                      enable system assertions
    -dsa | -disablesystemassertions
                      disable system assertions
    -agentlib:<libname>[=<options>]
                      load native agent library <libname>, e.g. -agentlib:hprof
                      see also, -agentlib:jdwp=help and -agentlib:hprof=help
    -agentpath:<pathname>[=<options>]
                      load native agent library by full pathname
    -javaagent:<jarpath>[=<options>]
                      load Java programming language agent, see java.lang.instrument
    [username@localhost ~]$
```

这个时候，可以正常使用JDK和Java运行环境了。

1.2 基于JDK1.5的第一个Java应用程序

首先，实现一个简单的功能，Java程序执行后，显示一条“欢迎访问Java应用程序”的信息，显示的是中文信息，为避免出现中文显示乱码，需要采取必要的措施和步骤。在后面的内容中，将会介绍Java程序如何处理中文信息，这是Java开发中文应用系统时必须面对的一个课题。

在介绍集成开发环境之前，为了使Java程序能够在Windows和Linux操作系统下运行，可以使用Windows的记事本(NotePad)或者Linux下的vi编辑器编写Java源程序。代码内容如下，将这些代码放在文件名为WelcomeApp.java的文件中：

```
public class WelcomeApp{
  public static void main(String[] args){
    System.out.println("欢迎访问Java应用程序");
  }
}
```

就是这样一个简单Java应用程序，不管是在Windows下的DOS命令界面，还是在

Linux 下的终端命令界面，编译并运行这个应用程序时，都会输出“欢迎访问 Java 应用程序”的文本信息，只不过要显示的是中文信息，需要对程序的源代码做特殊的处理。

中文处理涉及到汉字的编码，英文的字母编码通常使用 ASCII 编码，它是单字节的；汉字编码使用 GB2312 编码，它是双字节的。如果按照单字节处理汉字，就会出现乱码。为了解决这种字符串的国际化问题，不管是西文字母，还是汉字，或者其他国家的文字，都采用 Unicode 编码机制对其统一编码，而这种编码采用的是双字节。Java 编程的字符串编码默认为 Unicode(UTF-8)，根据编辑器的设置不同，Java 源文件可以为不同的字符编码。在编译 Java 源程序之前，需要将源程序的本地语言(native)编码转换为 Unicode 编码，转换后的中文的 Unicode 编码使用格式\uxxxx(UTF-16)表示。在涉及 Java 的资源文件的后续章节的相关内容时，还会进一步介绍解决资源包的中文问题。通过 JDK 提供的专用工具 native2ascii 完成此功能，运行 JDK 的命令如下：

```
native2ascii -encoding gb2312 WelcomeApp.java temp.java
```

如果省略-encoding 的本地参数选项，native2ascii 自动取当前操作系统的默认字符编码为本地编码参数，运行后将输出文件 temp.java 中的内容复制，然后粘贴覆盖 WelcomeApp.java文件中原来的内容，显示如下：

```
public class WelcomeApp{
  public static void main(String[] args){
    System.out.println("\u6b22\u8fce\u8bbf\u95eeJava\u5e94\u7528\u7a0b\u5e8f");
  }
}
```

从中可以看到，西文字符没有变化，而中文字符变成了\uxxxx 格式。接下来的步骤，将覆盖后的 Java 源文件 WelcomeApp.java 进行编译和运行，在 Windows 的 DOS 命令窗口和 Linux 的终端命令行窗口下的编译和运行分别如图 1-3 和图 1-4 所示。

图 1-3 在 Windows 的 DOS 命令窗口下 Java 程序的编译和运行

图 1-4 在 Linux 的终端命令行窗口下 Java 程序的编译和运行