清华电脑学堂 软件工程师培训系列

Java 程序设计培训教程

孙 燕 等 编著

北京计算机教育培训中心 组编

清华大学出版社

（京）新登字 158 号

内 容 简 介

本书是 Java 电视讲座的配套培训教材，主要介绍 Java 语言的基础知识和编程技巧。书中覆盖了 Java 编程实用技术的各个方面，并通过大量实例对这些内容进行分析和说明。本书对实例代码采用逐行解释的方式，让 Java 语言的初学者快速理解 Java 程序结构和编程技巧。本书内容翔实，重点突出，操作性强，可以帮助初学者在快速入门后，通过实例练习巩固书中的知识点，并达到中级编程水平。本书可作为 Java 的培训班教材，也可作为 Java 爱好者的自学参考书。

书　　名：Java 程序设计培训教程
作　　者：孙燕　等
出 版 者：清华大学出版社（北京清华大学学研大厦，邮编：100084）
　　http://www.tup.tsinghua.edu.cn
责任编辑：林庆嘉
印 刷 者：北京鑫丰华彩印有限公司
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　印张：17.5　字数：428 千字
版　　次：2002 年 6 月第 1 版　2002 年 6 月第 1 次印刷
书　　号：ISBN 7-302-05566-1/TP・3287
印　　数：0001~5000
定　　价：24.00 元

目　录

第1章 Java 概 述

Java 是一种广泛使用的网络编程语言，它最大限度地利用了网络，而且提供了丰富的类库，以满足网络化、多线程、面向对象系统的需要，使程序设计者可以非常方便地创建自己的系统。本章对 Java 语言出现的背景和该语言具有的特点进行简要的说明，并介绍常用的 Java 编程环境 JDK 1.3 的安装和相关环境变量的设置。

本章主要内容：

- 简单介绍 Java 背景知识
- 安装运行环境
- 设置环境变量
- 编译和运行 Java 两种程序类型 Application 和 Applet
- 编写一个简单的程序，在屏幕上显示“How are you”

1.1 Java背景知识

近些年来，计算机界最热门的话题莫过于 Internet，而当前 Internet 上最吸引人的技术莫过于 Java 了。计算机技术的应用正在朝着网络化的方向阔步前进，万维网成为世界上最大的信息中心。但目前万维网上的内容仍多为文本、图像、声音等静态信息，Java 为其提供了简便并且功能强大的编程接口，开发人员利用这些编程接口可以向 Web 增加动态性、交互性内容，这使得 Web 页面翻开了新的一页。

Java 是一种跨平台的面向对象的语言，这就允许开发人员生成独立于平台的应用程序，用 Java 编写的程序可以在许多硬件平台上运行而不需要重新编译。此外，Java 还有一个强大的安全模块，生成用户可以下载的 Applet 而不用担心任何安全性问题。

1.1.1 Java语言的特点

Java 确实称得上新一代编程语言，具有很多优点：简单、面向对象、可移植、与硬件无关、强健安全、具有很高的性能，此外还对多线程、动态性提供了支持。

注意： 为了使大多数程序员能够迅速掌握，Java 语言采取了类似 C++的语法。Java 语言虽然基于 C++，却除去了 C++中复杂不易理解的内容以及潜在的威胁程序安全性的模块，其中包括多重继承、指针等。

Java 的简单性首先体现在精简的系统，它力图使用最小的系统实现足够多的功能。大家可能已经发现，编程工具越来越复杂，占用系统资源也越来越大，通常需要几百兆字节的空间，功能可谓是非常强大了，但对使用者来说，要掌握它们，非得花上一番功夫不可，有时甚至令人望而却步。而对 Java 来说，其基本解释器只有 40KB 左右，加上标准类库和线程的支持也只有 210KB 左右，可谓短小精悍，但功能却毫不逊色，对面向对象、多线程

和多媒体都提供了全面的支持。

和所有新一代程序设计语言一样，Java 也采用面向对象技术，所有的 Java 程序都是对象，通过封装性实现了模块化和信息隐藏，通过继承性实现了代码的复用，使得用户可以根据自己的需要创建自己的类库。

网络上最重要的是安全问题，作为网络语言，Java 必须提供安全保障。在运行应用程序时，Java 严格检查数据的访问权限。不能使用指针，应用程序便不能非法访问对象的私有成员。

网络上充满了各种不同类型的机器和操作系统，怎样使 Java 程序能够在网络的任何地方运行？Java 解释器生成与体系结构无关的字节码结构的文件格式，只需提供相应的 Java 运行系统，程序便能在任何种类的处理器上运行。体系结构中立使得 Java 程序具有可移植性。Java 的类库也具有可移植性，可以在不同的平台上使用。

Java 语言支持多线程。多线程使得应用程序可以同时进行不同的操作，处理不同的事件。网络连接需要时间，如果在 Java 语言中采用事件循环机制，就会造成长时间的等待；采用多线程机制，不同的线程处理不同的任务，不同线程之间互不干涉，不会因为一处等待而影响其他部分，容易实现网络上的实时交互操作。

1.1.2 Java平台结构

图 1-1 给出了 Java 平台结构。Java 程序分两类，Java 应用程序和 Java Applet。

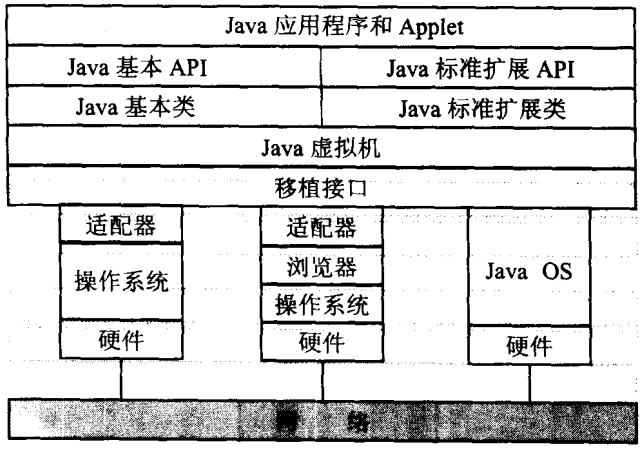

图 1-1 Java 平台结构

在这里，主要介绍 Java 虚拟机及其意义。Java 虚拟机通常定义为：运行经过编译的 Java 目标代码的计算机的实现。它能运行的 Java 程序包括 Java 应用程序和 Java Applet。Java 虚拟机对 Java 的平台独立性和安全性有非常重大的意义。在图 1-1 中，Java 虚拟机处于核心位置，它的下方是移植接口，移植接口由依赖平台和不依赖平台的两部分组成，依赖平台的部分称为适配器，Java 虚拟机通过移植接口在具体的操作系统上实现；不依赖平台的部分为 Java OS，这时不需要依赖于平台的适配器，相关的工作由 Java OS 完成。从图 1-1 中可以看出，对于 Java 虚拟机来说，操作系统和更低层的硬件是透明的，也就是说，对于 Java 虚拟机这一层次来说，操作系统和硬件好像不存在一样，因而也不需要考虑。

在 Java 虚拟机上方，是 Java 的类和 Java API。在 Java API 上，可以编写 Java 应用程序和 Java Applet。对于 Java 应用程序和 Applet 这一层次来说，操作系统和硬件更是透明的，因此编写的 Java 程序可以在任何 Java 平台上运行而不需修改。

1.1.3　Java和C++的比较

Java 语言虽然基于 C++，它们之间还是有许多不同的地方。下面是对这两种语言进行的比较。

1．指针

熟悉 C++语言的朋友都知道，指针是 C++中最灵活的数据类型，同时也是最容易出错的数据类型，通过指针进行内存操作，通常会造成不可预知的错误。此外，由于可以使用指针对内存地址进行显式类型转换，也就可以访问类的私有成员，从而破坏了安全性。为了维护程序的安全性，在 Java 中，不能进行任何指针操作。

2．全局变量

在 C++中，使用全局变量有时会简化程序的设计，但是不加封装的全局变量常会造成系统的崩溃。在 Java 程序中，不能定义全局变量，通常通过类中的公用、静态变量实现全局变量，使得全局变量封装在类中，这样就提供了更好的安全性。

3．条件转移指令

C++中使用 goto 语句处理不同条件下的程序跳转情况，使得程序设计起来比较灵活，但降低了程序的可读性。Java 中不支持 goto 语句，而是通过异常处理语句 try,catch,final 代替 goto 语句，来处理程序遇到异常的情况。

4．内存管理

在 C++中，程序员可以使用运算符 new 和 delete 来分配和释放内存，内存管理是由程序员自己维护的，若忘记释放不再使用的内存，就会逐渐耗尽系统的资源，如果释放未被分配的内存块或再次释放已经释放的内存块，就会造成系统的崩溃。在 Java 中，程序员可以通过运算符 new 分配内存，但 Java 可以自动地进行内存管理并进行自动垃圾收集，这样就有效地防止了因程序员误操作引起的错误，从而更好地利用了系统的资源。

5．数据类型一致性

对 C++程序来说，在不同的平台上，编译器对简单的数据类型分别分配不同的字节数，从而导致了代码的不可移植。Java 对某种数据类型总是分配固定的位数，这样就保证了 Java 的平台无关性和可移植性。

6．类型转换

在 Java 中，进行类型转换时，系统要对对象的处理进行相容性检查，防止进行不安全的类型转换。而在 C++中，由于可以通过指针进行任意的类型转换，从而存在不安全的类型转换。

7．结构和联合

C++中支持结构和联合，由于结构和联合均为公有属性，这就带来了安全性问题。在 Java 中，不支持结构和联合。

8．宏定义

C++中可以用宏定义进行常量的声明，有时可以简化程序的设计，但会影响程序的可读

性。在 Java 中不支持宏定义，使用关键字 final 声明常量。

9. 头文件

C++使用头文件声明类的原型、全局变量以及库函数，但对于一个大的系统，维护这些头文件非常困难。Java 不支持头文件，使用 import 语句与其他类进行通信，以访问其他类中的方法。

1.2 运行环境的介绍和安装

俗语说，工欲善其事，必先利其器。要想学好一门编程语言，选择一个好的运行环境非常重要。在这里建议大家使用 JDK 1.3。JDK 是“Java Development Kit (Java 开发工具包)”的缩写，由 Sun 公司开发。最初的版本是 JDK 1 现在已经发展到 JDK 1.3。JDK 是所有 Java 开发工具的基础，JDK 可以到 java.sun.com 上下载。

JDK 中共有 6 个重要的包，依次为：

java.AWT 提供了容器和众多组件，是图形用户界面设计不可缺少的，也是一个与平台无关、用于图形用户界面编程的类库集合。

java.applet 为创建小应用程序提供必要的元件，包括声音播放资源。

java.io 提供与设备无关的输入、输出流支持。

java.lang 提供支持 Java 的基础类。

java.net 提供支持联网的类。

java.util 提供实用方法和数据结构的类。

Java 开发工具（JDK）是许多 Java 专家最初使用的开发环境。尽管许多编程人员已经使用第三方的开发工具，但 JDK 仍被当作 Java 开发的重要工具。

JDK 由一个标准类库和一组建立、测试及建立文档的 Java 实用程序组成。其核心 Java API 是一些预定义的类库，开发人员需要用这些类来访问 Java 语言的功能。Java API 包括一些重要的语言结构以及基本图形、网络和文件 I/O。一般来说，Java API 的非 I/O 部分对于运行 Java 的所有平台是相同的，而 I/O 部分则仅在通用 Java 环境中实现。

作为 JDK 实用程序，工具库中有七种主要程序。

javac Java 编译器，将 Java 源代码转换成字节码。

java Java 解释器，直接从类文件执行 Java 应用程序字节代码。

appletviewer 小程序浏览器，一种执行 HTML 文件上的 Java 小程序的 Java 浏览器。

javadoc 根据 Java 源码及说明语句生成 HTML 文档。

jdb Java 调试器，可以逐行执行程序，设置断点和检查变量。

javah 产生可以调用 Java 过程的 C 过程，或建立能被 Java 程序调用的 C 过程的头文件。

javap Java 反汇编器，显示编译类文件中的可访问功能和数据，同时显示字节代码含义。

下面介绍 JDK 1.3 的安装过程。

（1）找到 JDK 1.3 软件安装程序，双击运行。等待几秒后，屏幕上就会出现安装画面。从安装画面上可以明显看到 Java（TM）2 SDK，Standard Edition，v1.3 字样，表明是 JDK 1.3 版（基于 Java 2）。

（2）单击安装画面上的 Next 按钮，屏幕上出现 Software License Agreement 对话框。

（3）单击 Yes 按钮，进入 Choose Destination Location 对话框，修改安装目录为自己想要目录或保持默认值，默认路径为 c:\jdk1.3，我们将它安装在 d:\jdk1.3 路径下。

（4）单击 Next 按钮，系统开始安装。在安装结束后，单击 Finish 按钮结束安装。

安装完后，要想检测 JDK 是否安装正确，启动到 DOS 界面下，进入 d:\jdk1.3\bin 路径下，然后键入 javac 或 java 命令，看是否出现错误提示，如果没有错误提示就表示 JDK 的初步安装过程是成功的。倘若出现错误提示，表示安装不成功，原因可能是你下载的 JDK 版本本身存在问题，可以重新下载一个以后再安装试试。

JDK 安装成功后，可以在 d:\jdk1.3\bin 路径下编译和运行 Java 程序，但是要想在其他的路径下也可以编译和运行 Java 程序，就需要设置环境变量，具体操作如下：

首先在桌面上用鼠标右击桌面上的“我的电脑”图标，选择“属性”菜单项，就会出现如图 1-2 所示的界面。

在图 1-2 所示的界面中，单击“高级”选项卡，就会看见如图 1-3 所示的画面。

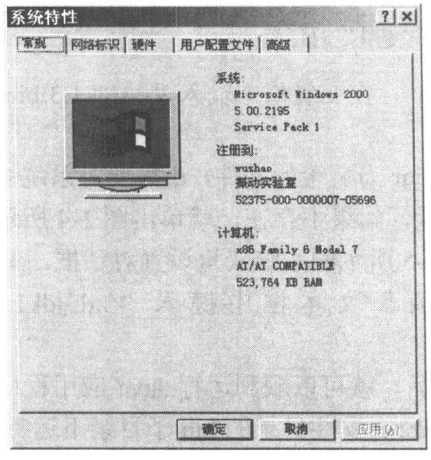

图 1-2　系统属性界面

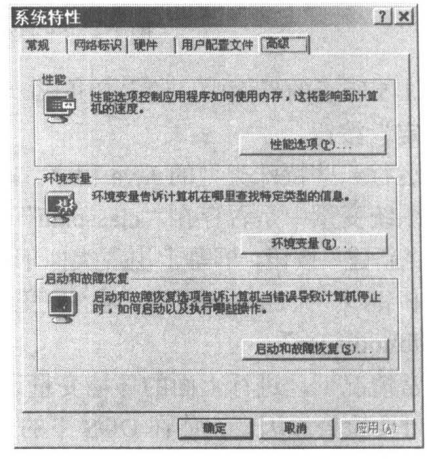

图 1-3　系统的高级属性界面

单击图 1-3 所示界面上的“环境变量”按钮，屏幕上就会出现如图 1-4 所示的窗口。

图 1-4　环境变量设置界面

“环境变量”设置界面上的上半部分是用户变量设置。这是因为 Windows 2000 是多用户操作系统，支持不同用户的个性化系统定制。这里设置的信息只影响当前用户，而不会影响其他用户。“环境变量”设置界面上的下半部分是整个系统的环境变量，改变系统的环境变量，会影响到所有用户。

首先设置用户变量 path。如果用户变量中已经存在 path，双击 path 变量条目，随后出现“编辑用户变量”对话框，在对话框的“变量值”文本框中添加“;d:\jdk1.3\bin”，然后单击“确定”按钮。倘若用户变量中不存在 path 变量，就单击图 1-4 中的“新建”按钮，出现“新建用户变量”对话框，如图 1-5 所示。

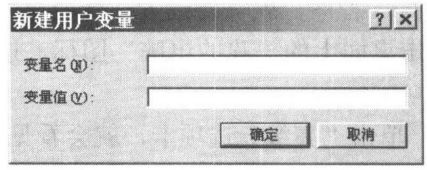

图 1-5 “新建用户变量”对话框

在图 1-5 所示的界面中，输入变量名“path”，在变量值中输入“d:\jdk1.3\bin”，单击图中的“确定”按钮。

为了运行一些特殊类型的 Java 程序，如以.jar 为后缀的文件，还需要对系统变量进行设置。查看系统变量中是否存在“classpath”变量，如果不存在，就单击图 1-4 所示对话框下半部分的“新建”按钮，屏幕上出现类似于图 1-5 所示的系统变量设置对话框，在“变量名”文本框中键入“classpath”，在“变量值”文本框中键入“.;d:\jdk1.3\lib\dt.jar;d:\jdk1.3\lib\tools.jar”。

在正常情况下，进行上面的环境变量设置后，就可以顺利运行 Java 应用程序；如果不能运行，可要检查原因。比如在 DOS 下的非 jdk1.3 安装目录的 bin 子目录下运行 Java 命令时，如果出现下面的提示：

```
Error opening registry key 'Software\JavaSoft\Java Runtime Environment'
Error: could not find java.dll
Error: could not find Java 2 Runtime Environment
```

那么是 DOS 系统无法识别 Java 2 运行环境。可采用下面步骤来建立 Java 2 运行环境：

（1）单击屏幕左下角“开始”菜单，并单击“运行（R）”菜单项。

（2）在弹出的“运行”窗口的“打开（O）”编辑框中键入 regedit，单击“确定”按钮。屏幕上弹出“注册表编辑器”窗口。

（3）找到并选中 HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft 下的“Java 运行时环境”子键，用鼠标右击该子项，从弹出的快捷菜单中选择“重命名”，将该子项名称修改为“Java Runtime Environment”。

接下来需要为编写 Java 程序选择编辑软件。Java 程序可以使用任何文本编辑器，比如 Notepad（记事本）、Word 等，不过推荐使用一些功能强大而又不过多消耗系统资源的文本编辑器，比如 UltraEdit。它可以以不同颜色标志出普通代码，关键字；可以支持多窗口显示，便于比较类似的文件，或在文件之间进行复制、粘贴，等等。可以安装 UltraEdit32 版本的 UltraEdit 编辑器。

1.3　Java程序结构介绍

Java 用户程序分两类，一类是一般形式的应用程序，这类程序通过编译器编译以后，Java 解释器边解释边执行，通常说的 Java 应用程序就是指这类程序。另一类程序是 Applet，该类程序的规模比较小，但值得注意的是，这类程序不能单独运行，Applet 程序必须被嵌入网络的 Web 页中，需要通过与 Java 兼容的浏览器控制执行。下面通过两个简单的例子说明这两类程序各自的编译运行方式。

首先来看看一般形式的 Java 应用程序。

```
class FirstApp{
    public static void main(String args[]){
        System.out.println("How are you");
    }
}
```

上面是一个简单的应用程序，实现的功能是输出一行信息：How are you。让我们一起来看看 Java 应用程序的基本结构。

- 类的定义

值得注意的是，所有的 Java 应用程序都是类，关键字 class 用来声明新类，生成一个 Java 应用程序，也就是需要定义一个类。可以在 class 关键字前面用 public 修饰，表示该类是一个公共类，在 Java 程序设计中，可以定义多个类，但是最多只能有一个公共类。

- main()方法

所有的应用程序都必须有且只有一个 main()方法，而且必须用关键字 public、void 和 static 指定。public 表明所有的类都可以使用这个方法，void 指明该方法不返回任何值，而 static 表明 main()方法是一个类方法，可以直接通过类名调用。Java 解释器在没有生成任何实例时，以 main()方法作为入口来执行程序。

- 程序内容

在上面的程序中，main()方法只有一条语句：

```
System.out.println("How are you");
```

用于实现字符串“How are you”的输出。

需要提醒大家注意的是，在保存编写的 Java 程序时，文件名必须是公共类名，这是因为 Java 解释器要求公共类必须放在同名的文件中，如上面的程序应该保存为 FirstApp.java，并为该文件选择合适的路径，在这里将文件保存在 d:\jdk1.3\tv 路径下。

接着在 DOS 下使用命令 javac 编译程序，如下面的语句所示：

```
javac FirstApp.java
```

查看 FirstApp.java 文件所在的目录下，已经生成字节码文件 FirstApp.class，然后使用 Java 解释器运行它，命令如下：

```
java FirstApp
```

屏幕上随后就会出现“How are you”字符串。

接下来介绍小应用程序 Applet。Applet 就是使用 Java 语言编写的一段代码，它可以在浏览器环境中运行。它与 Application 的区别主要在于其执行方式的不同。Application 是从 main()方法开始运行的，而 Applet 是在浏览器中运行的，因此无法使用命令直接运行，必须创建一个 HTML 文件，通过编写 HTML 语言代码告诉浏览器载入何种 Applet 以及如何运行，使用时，用户只需要在浏览器中给出 HTML 文件的 URL 地址就可以了。

同样编写一个显示 How are you 字符串的程序。程序代码如下所示：

```
import java.awt.Graphics;
import java.applet.Applet;
public class FirstApplet extends Applet{
    public String outStr;
    public void init(){
        outStr = new String("How are you");
        System.out.println(outStr);
    }
    public void paint(Graphics g){
        g.drawString(outStr,60,60);
    }
}
```

从上面这个例子可以看到 Java Applet 的基本结构：

- 类定义

和 Java 应用程序一样，所有的 Java Applet 也都是类，而且必须是 Applet 类的子类，在上面的程序中是使用关键字 extends 实现的。

- 方法

和 Java 应用程序不同，在 Java Applet 中没有 main()方法，由于 Applet 类实现了许多方法，用户只需要重写自己所需的方法就可以了，系统将自动调用。由于 Applet 本身也是一个 AWT 组件，具有图形绘制功能，在上面的程序中，就是通过 paint()方法进行绘图具体操作。

init()方法中声明并初始化了一个字符串 outStr，并输出“How are you”字符串，不过值得注意的是这种方式只是将字符串输出在 DOS 屏幕上。paint()方法在生成的窗口的（60，60）位置处将字符串“How are you”显示出来。

- 类库的使用

在程序中如果要用到 Java 类库中的一些方法，就必须在程序的开头部分进行声明。Java 语言使用 import 来输入类。例如在上面的程序中，通过 import 关键字输入了 java.awt.Graphics 和 java.applet.Applet 类。

下面我们看看如何编译和运行 Applet 程序。

首先使用下面的命令编译上面的程序：

```
javac FirstApplet.java
```

使用上面的命令生成了字节码文件 FirstApplet.class，要想看到 Applet 程序的运行效果，需要编写 HTML 文件，将 Applet 嵌入其中，然后使用 appletviewer 来运行，为上面程序编写的对应 HTML 文件为 FirstApplet.html，内容如下：

```
<html>
<applet code = "FirstApplet.class" height = 160 width = 200>
</applet>
</html>
```

在 FirstApplet.html 文件中，将上面生成的 FirstApplet.class 嵌入其中，height 和 width 分别指定生成的窗口的高和宽。

接着使用下面的命令观看 Applet 的效果：

```
appletviewer FirstApplet.html
```

就可以看到如图 1-6 所示的结果。

图 1-6　Applet 运行效果

同时在 DOS 屏幕上也可以看到“How are you”字符串，只是需要注意的是，DOS 屏幕上的字符串是由 init()方法中的语句实现的，而图 1-6 中显示的字符串是由 paint()方法中的语句实现的。

1.4　小　结

本章对 Java 背景知识和 Java 运行环境及其安装进行了介绍，此外对两类 Java 程序的结构进行了简单的说明。在 Java 背景知识中，介绍了 Java 语言的特点、Java 平台结构以及 Java 和 C++之间的不同点。本章的重点是 JDK 1.3 的安装和相关环境变量的设置，以及 Java Application 和 Java Applet 程序的结构及其编译运行的不同之处。

1.5　练　习　题

1. Java 语言有哪些特点？Java 语言为什么具有跨平台特性？
2. Java 语言与 C/C++语言有哪些不同之处？
3. Java 程序分哪两类？各自的程序结构是什么？各有什么特点？
4. 在 JDK 1.3 运行环境中，如何编译和运行 Java 程序？
5. 下面是两个简单的 Java 程序，指出程序中的错误。

```
(1) class FirstApp
    public static void main(String args[]){
```