



21st CENTURY
规划教材

面向21世纪高等院校计算机系列规划教材

COMPUTER COURSES FOR UNDERGRADUATE EDUCATION

Java 程序设计实验与习题

唐文彬 主 编
陈 华 张若冰 副主编



科学出版社

www.sciencep.com



面向21世纪高等院校计算机系列规划教材
COMPUTER COURSES FOR UNDERGRADUATE EDUCATION

Java 程序设计实验与习题

唐文彬 主 编

陈 华 张若冰 副主编

科学出版社

北 京

内 容 简 介

本书从 Java 语言的基础入手,由浅入深地通过详细的实验步骤和完整的实验程序指导每个实验。基础篇有 10 个实验,其内容主要包括 Java 语言基础实验、数组与控制语句、类与面向对象程序设计、包与接口的使用、异常处理、输入输出和文件处理、图形界面和 Applet 程序,每个实验后面都附有上机实践题,使读者在学习过程中学练结合,从而达到充分掌握每个实验的目的。提高篇有两个实验,包括一个时钟程序和一个记事本程序,通过实用软件的理解和修改提高学习 Java 的兴趣。

本书适合作为 Java 初学者的上机实验指导教材,也可作为自学 Java 语言上机练习的参考用书。

图书在版编目(CIP)数据

Java 程序设计实验与习题/唐文彬主编.—北京:科学出版社,2005

(面向 21 世纪高等院校计算机系列规划教材)

ISBN 7-03-016140-8

I. J… II. 唐… III. JAVA 语言-程序设计-高等学校-教学参考资料
IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 091161 号

责任编辑:吕建忠 韩 洁 / 责任校对:耿 耘

责任印制:吕春珉 / 封面设计:飞天创意

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂印刷

科学出版社发行 各地新华书店经销

*

2005 年 8 月第 一 版 开本:787×1092 1/16

2005 年 8 月第一次印刷 印张:10 1/4

印数:1—3 000 字数:234 000

定价:15.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

销售部电话 010-62136131 编辑部电话 010-62135927-8001 (H101)

前 言

目前,各理工类高校大多开设了 Java 程序设计课程。该课程教学与上机的比例大致为 2:1。市场上关于 Java 程序设计的教材很多,但如何指导学生上机操作的书籍却很少。尽管市场上也有一些 Java 语言的上机指导或解题教材,但这些教材仅仅是介绍如何上机操作,对上机操作过程中经常遇到的问题介绍的不够,其实践性不够强,一些学生往往在上机时不知干什么,或者上机操作时遇到问题在教科书中找不到解决办法。如何指导学生上机实践操作,减轻教师的工作量,是急需解决的问题,这也是编写本书的目的。

本书通过手把手的指导和按图示操作讲解,直观生动,让读者能够快速掌握 Java 编程的基本方法,提高学习效率,快速入门。为巩固和加深对所写内容的理解,每部分还有针对性的设置了实践题目,使读者通过思考能得到进一步的提高。

学习任何一门计算机语言,不动手编制程序是不可能学好的。只有亲自动手输入、编译、运行才能真正掌握 Java 语言,才能发现编制 Java 程序是一件很有趣的事情。

本书共十二个实验,其中唐文彬编写了实验一、实验二、实验十一、实验十二,并对全书进行了审校,陈华编写了实验五、实验六、实验七、实验八,张若冰编写了实验三、实验四、实验九、实验十。

由于作者的水平和经验有限,疏漏之处在所难免,恳请读者不吝赐教。

目 录

第一部分 基 础 篇

实验一	Java 开发环境.....	2
实验二	Java 语言基础.....	11
实验三	Java 控制语句.....	21
实验四	字符串与数组.....	34
实验五	类与面向对象.....	44
实验六	Java 接口和 Java 包.....	58
实验七	Applet 小程序.....	72
实验八	图形用户界面.....	82
实验九	Java 异常处理.....	96
实验十	输入输出与文件处理.....	101

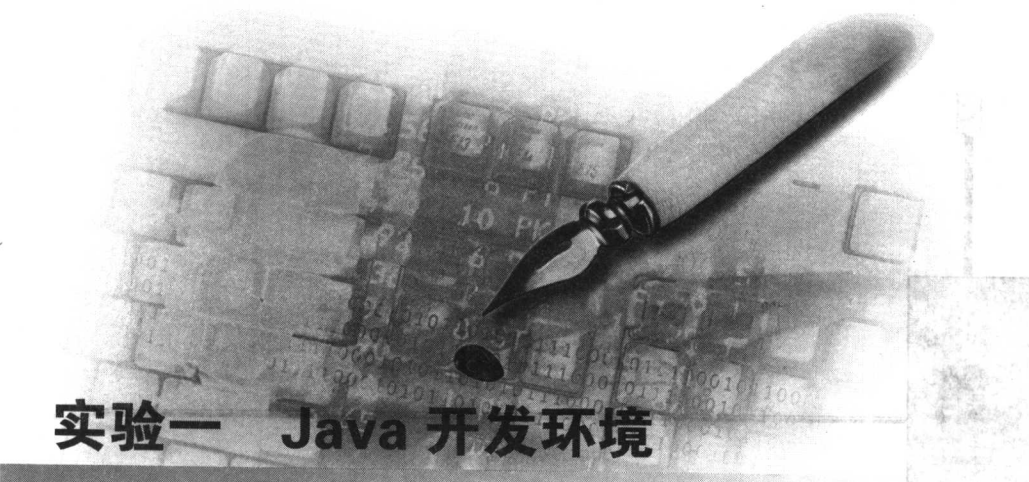
第二部分 提 高 篇

实验十一	Java 时钟.....	116
实验十二	Java 记事本.....	124
附录		154
主要参考文献		157

第一部分

基础篇

本书分为两个部分。第一部分——基础篇，针对的是没有 Java 基础的读者，本部分从最基本的知识点开始，通过图形实例一步步地引领大家对 Java 语言有所认识，为下一步的学习和提高打下基础。其实万丈高楼的建造也是从基础开始的，希望读者能多学、多练、多想。对本部分的学习取得满意的效果。



实验一 Java 开发环境



开篇论

Java 的开发过程需要编写源程序、编译生成字节码和运行三个部分，除了编写源程序可在任意的文本编辑器中完成外，编译和运行都需要在一定的环境下实现。目前可供开发人员选择的 Java 语言开发环境很多，常用的 Java 开发包是 JDK（Java development kit，Java 开发包）。本节实验将学习如何安装 Java 软件开发工具包，以及如何编译和运行不同类型的程序，一旦你掌握了本节实验所讲述的技巧而且选择了合适的开发工具，就开始了 Java 语言的探索之旅。

一、实验目的

- 1) 认识 Java 开发工具包。
- 2) 熟悉开发工具的使用。
- 3) 学习编写简单的 Java Application 和 Java Applet 程序。
- 4) 了解 Java 源代码、字节码文件。
- 5) 掌握 Java 程序的编辑、编译和运行过程。

二、实验内容

- 1) 动手安装和配置 Java 开发工具包。
- 2) 熟练使用开发工具中的命令。
- 3) 编写一个简单的 Java Application 和 Java Applet 程序。

三、实验步骤

古人云，工欲善其事，必先利其器。没有编程开发的环境，一切都无从谈起。因此我们先学会安装和配置 Java 开发工具包。运行 Java 的硬件环境当然是越高档越好，但

是不是高档的机器配置就一定能编写出优秀的程序，“人”才是无限创造力的源泉！所以一些配置稍微差些的机器也能安装运行 Java 的系统，不过稍微慢些。当然，如果是要架设用于实用的服务器，那就另当别论了，这时对硬件的要求就相对高些。一般来说，我们推荐的最低配置至少是 Intel Pentium 处理器，至少 32MB 内存和至少 50MB 的硬盘空间。具备这些软硬件条件，加上我们的学习能力和创造力，相信大家一定能开发出优秀的 Java 程序。

1. 安装包的取得

Java 产品面向的主流平台有 Solaris、Windows、Linux，现在常用的 Java 开发环境是 JDK1.4（本书以 JDK1.4 版本为例），并且一般的应用都是基于 Java2 SDK 的标准，所以只需要下载 J2SE 的版本即可。最新的版本有 JDK1.5，可以到 Sun 公司的网站 (<http://java.sun.com>) 上下载。



基本的 Java SDK 不是一种集成文本编辑器、编译菜单和调试器的开发环境。所有的动作都需要在 shell 窗口中输入命令完成。Java 也有集成开发环境，但它们都比较复杂，对于初学者，使用自己喜欢的文本编辑器学习 Java 仍然是很好的选择。

2. 安装和配置 Java 开发工具包

安装完 Java SDK 后，还需要对执行路径进行设置。对于不同的操作系统，执行路径的设置是不同的。

(1) 设置执行路径

假设 JDK 安装在 c:\jdk1.4 目录下(用户可以自行选择)。

1) 平台为 Windows 9x 时，需修改系统根目录下的 autoexec.bat 文件的 path 和 classpath 变量，例如：

```
set path=%path%; c:\jdk1.4\bin
set classpath=.; c:\jdk1.4\lib
```

重新启动计算机后设置生效。

2) 平台为 Windows 2000/NT/XP 中，打开“控制面板”，如图 1-1 所示。



图 1-1 打开控制面板

双击“系统”图标，打开如图 1-2 所示的“系统”对话框。在“系统属性”中选择“高级”选项卡，打开“环境变量”对话框，在用户变量窗口中找到 path 变量，将 c:\jdk1.4\bin 加到后面。新建环境变量 classpath，置 classpath=.; c:\jdk1.4\lib，保存设置，如图 1-3 和图 1-4 所示。



Java 只有安装和编译指令是和系统有关的，一旦安装好并运行 Java 后，实验中所涉及的内容就与系统无关了。

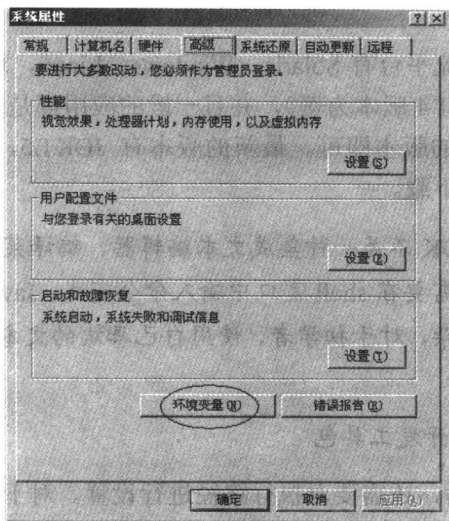


图 1-2 “系统属性”对话框

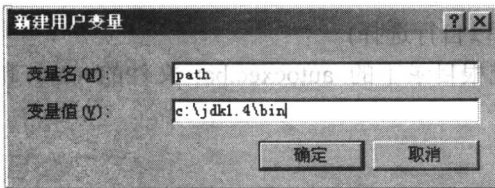


图 1-3 path 变量设置

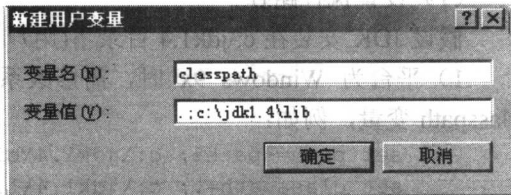


图 1-4 classpath 变量设置

(2) 测试

启动一个 shell 窗口，输入下列命令：

```
C:\>java -version
```

如果出现如图 1-5 所示的提示信息，说明安装成功。也就是说可以编译执行 Java 程序了。

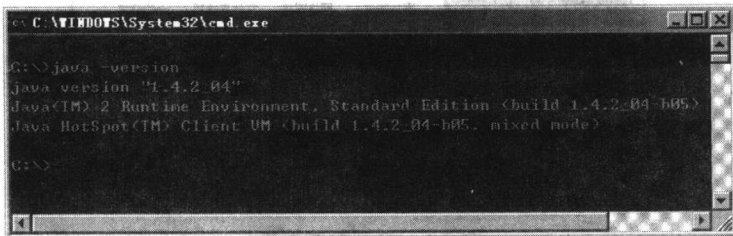


图 1-5 安装成功



Java 提供了一些常用的语言工具，主要包括：

java——解释器

javac——编译器

appletviewer——小应用程序浏览器

javah——头文件生成器

javadoc——API 文档生成器

javap——类文件反汇编器

jdb——Java 语言调试器

这些文件包括在 c:\jdk1.4\bin 目录中，并可以在任何目录中运行，前提是设置了运行程序的相应系统路径。

3. Java 初探

在正式学习 Java 程序前，我们先来欣赏几个用 Java 程序生成的程序截图，如图 1-6 所示。这些程序都可以在 JDK 自带的 demo 包中取得。

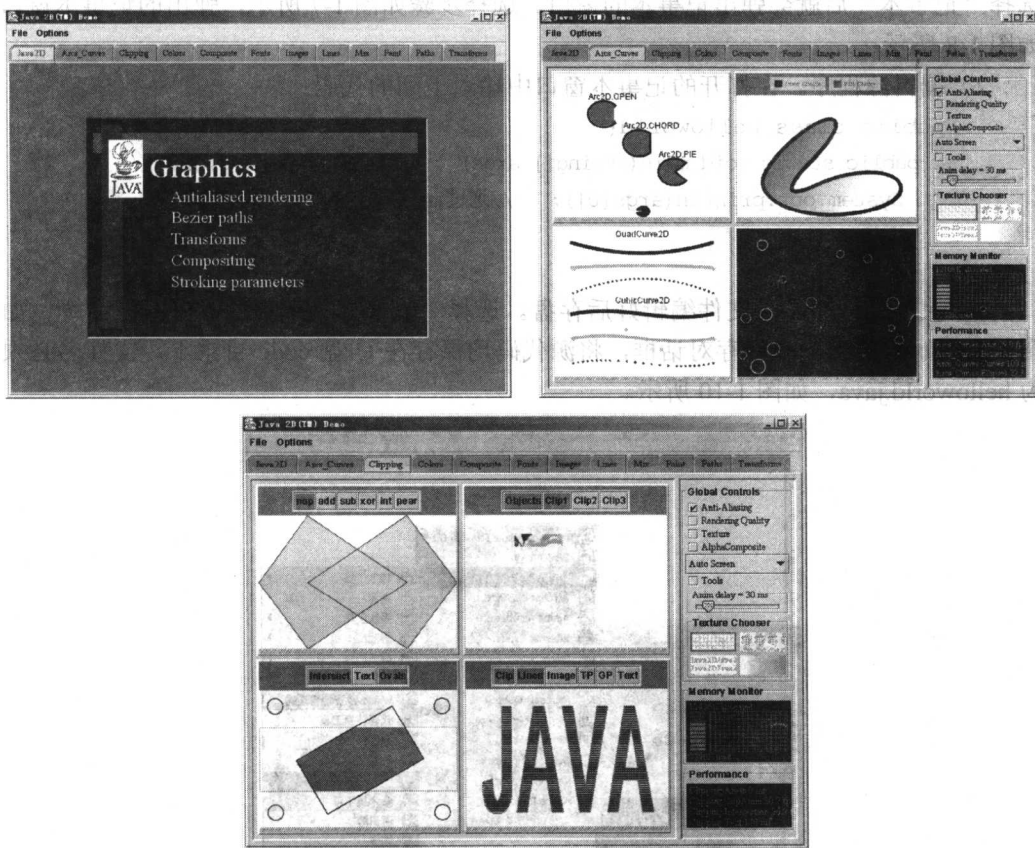


图 1-6 Java 程序截图

欣赏完这些 Java 程序所生成的图形之后，你是否有一些好奇呢？那么，让我们一起进入神奇的 Java 世界吧。

4. 编辑、编译和运行 Java 源程序

Java 程序分为两种类型：一种是 Java Application，它以 main() 方法作为程序入口，

由 Java 解释器加载执行；另一种是 Java Applet，它通过把类名镶嵌到 HTML 文档中，通过浏览器或 Appletviewer 加载执行，它没有作为程序入口 main() 的方法。

初学者可以使用文本编辑器如 UltraEdit，生成一个以 .java 为后缀的文本文件。源文件必须经过编译，生成字节码的类文件后才能运行。Java 的编译器是 javac。

- 1) 编辑 Java 源程序。
- 2) 编译生成字节码文件。
- 3) 解释与运行字节码文件。

下面以经典的 hello world 为例。

例 1-1 Java Application 源程序。

1) 打开记事本。这里我们使用的文本编辑器是记事本，它是 Microsoft 操作系统自带的文本编辑器。通过“开始”按钮，选择“程序”，再选择“附件”，在附件的清单中，选择“记事本”后就会弹出记事本的窗口。选择步骤如图 1-7 所示，弹出的记事本窗口如图 1-8 所示。

2) 编辑源程序。在打开的记事本窗口中输入下列的代码：

```
public class helloworld{
    public static void main(String[] args) { //程序入口main()的方法
        System.out.println(args[0]); //通过main 方法的参数传入字符串
    }
}
```

3) 保存源程序。源文件编辑好后存盘。选择“文件”菜单下的“保存”命令，如图 1-9 所示。弹出一个保存对话框，将源代码均保存在 C 盘 code 目录下，文件名必须为 helloworld.java，如图 1-10 所示。

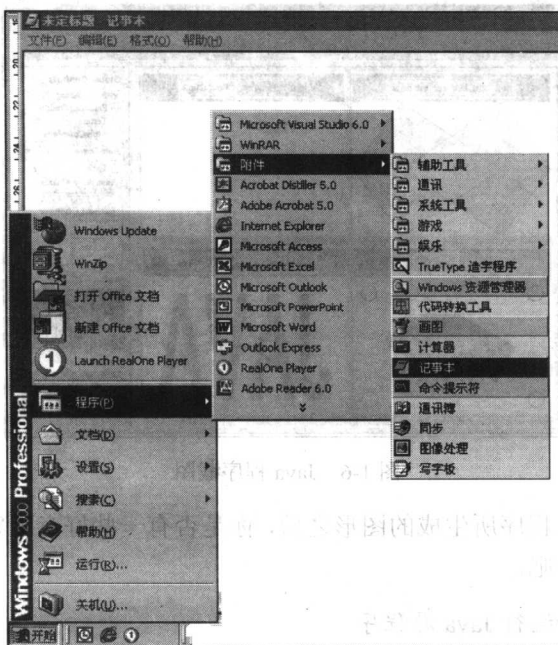


图 1-7 打开记事本

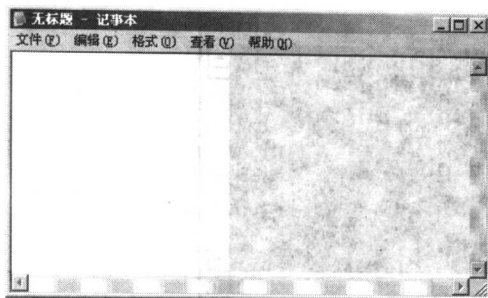


图 1-8 记事本窗口

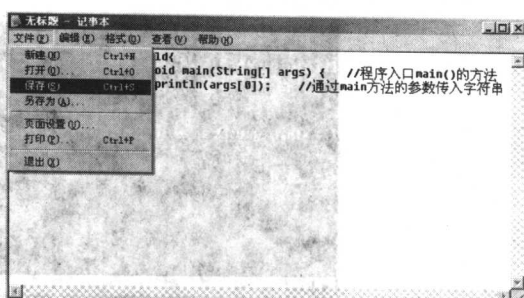


图 1-9 保存源程序

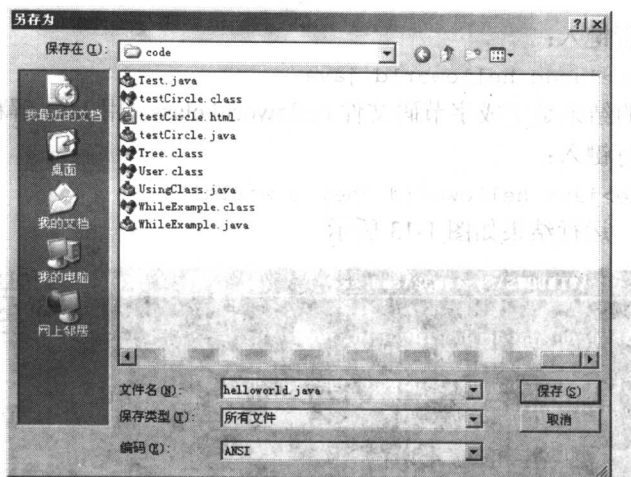


图 1-10 存为 helloworld.java

4) 编译、运行源程序。将源文件存盘为 helloworld.java 后，打开一个 shell 窗口，进行编译。通过“开始”按钮，选择“程序”，再选择“附件”，在附件的清单中，选择“命令提示符”后就会弹出如图 1-11 所示的窗口。

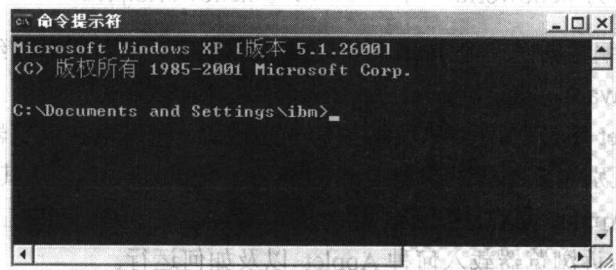


图 1-11 命令提示符窗口

此时当前路径通常不是我们保存源程序的路径，如本书中所有源程序均保存在 C 盘的 code 目录下，图 1-11 所示的当前路径并不是此路径，所以必须改变目录的路径，可以通过 CD 命令来改变当前目录。在命令行键入：

```
cd c:\code
```

具体如图 1-12 所示。

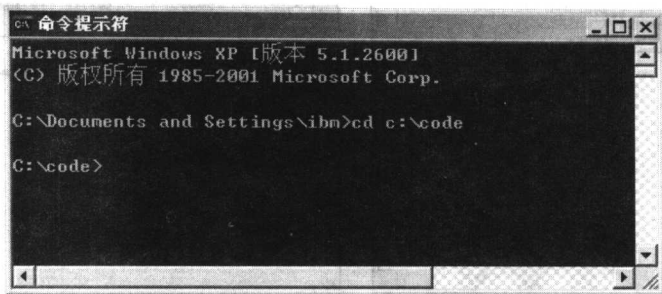


图 1-12 改变路径

此时在命令行键入：

```
C:\code>javac helloworld.java
```

进行编译，编译的结果是生成字节码文件 `helloworld.class`。用 Java 解释器来运行该字节码文件，在命令行键入：

```
C:\code>java helloworld "hello world!"
```

即可运行该程序，运行结果如图 1-13 所示。

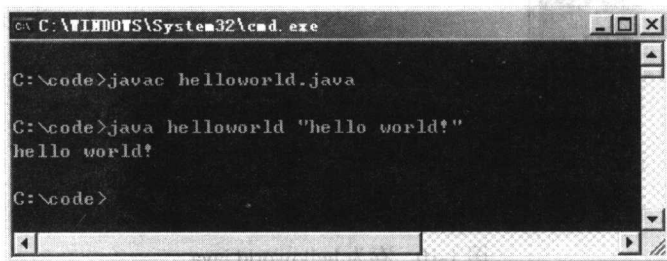


图 1-13 程序 helloworld 的运行结果



源文件存盘时，文件名必须与类名一致，并区分大小写，扩展名为 `.java`。该例的类名为 `helloworld`，所以文件名为 `helloworld.java`。

例 1-1 中我们编写的是一个 Java Application 源程序，下面同样以经典的 hello world 为例，编写一个 Java Applet 源程序。

Java 中，Applet 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 是从其中的 `main()` 方法开始运行的，而 Applet 是在浏览器中运行的，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。

例 1-2 Java Applet 源程序。

1) 编辑源程序。

打开一个文本编辑器，输入下列代码：

```
import java.applet.Applet;
import java.awt.Graphics;

public class helloworldApplet extends Applet{
    public String i;
```

```
public void init(){ //Applet 的初始化方法
    i="hello world!";
}
public void paint(Graphics g){
    g.drawString(i,25,25);
}
}
```

2) 保存源程序。源文件编辑好后存盘。文件名必须是 `helloworldApplet.java`。详细过程可参考例 1-1。

3) 编译源程序。打开一个 shell 窗口, 进行编译。在命令行键入:

```
C:\code>javac helloworldApplet.java
```

编译的结果是生成字节码文件 `helloworldApplet.class`。



源文件存盘时, 文件名必须与类名一致, 并区分大小写, 扩展名为 `.java`。该例中的类名为 `helloworldApplet`, 所以文件名为 `helloworldApplet.java`。

4) 运行源程序。Java Applet 不能直接在命令行下执行, 需要编写 HTML 文件, 把 Applet 嵌入其中。

打开一个文本编辑器输入名为 `helloworldApplet.html` 的文件源代码:

```
<HTML>
<applet code="helloworldApplet.class"WIDTH=200 HEIGHT=100>
</applet>
</HTML>
```



此文件名可随意确定, 该例中文件名为 `helloworldApplet.html`。

有两种方式可以运行 Java Applet:

1) 在浏览器下运行, 如 IE、Netscape 等, 打开 HTML 文件, 即会装载类文件并运行。

2) 用 `appletviewer` 来运行。本例中采用此方式, 在命令行键入:

```
C:\code>appletviewer helloworldApplet.html
```

这时屏幕上弹出一个窗口, 显示:

```
hello world!
```

具体运行如图 1-14 所示, 运行结果如图 1-15 所示。

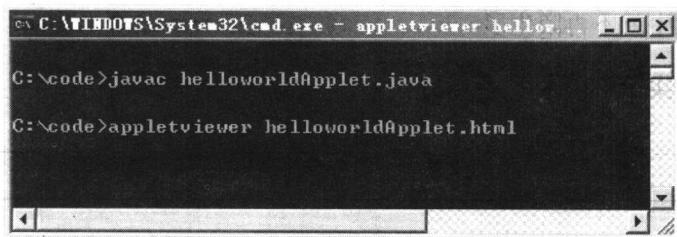


图 1-14 程序 `helloworldApplet` 的编译及运行方式

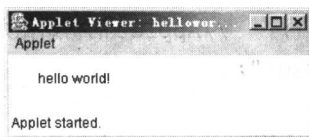


图 1-15 程序 helloworldApplet 的运行结果

四、上机实践题

此环节可以巩固读者在实验环节中所学的内容，同时积累实验环节外进行独立编程的经验。

1) 编写一个 Java Application 程序，编译并运行这个程序，在 shell 窗口上输出“Welcome to java!”。

2) 编写一个 Java Applet 程序，编译并运行这个程序，使之能够在浏览器上显示出“Welcome to java!”。

3) 输入下面源程序并编译运行，写出运行结果（不要求读者完全理解所有输入的语句，但必须严格按下面的内容和格式输入。）。

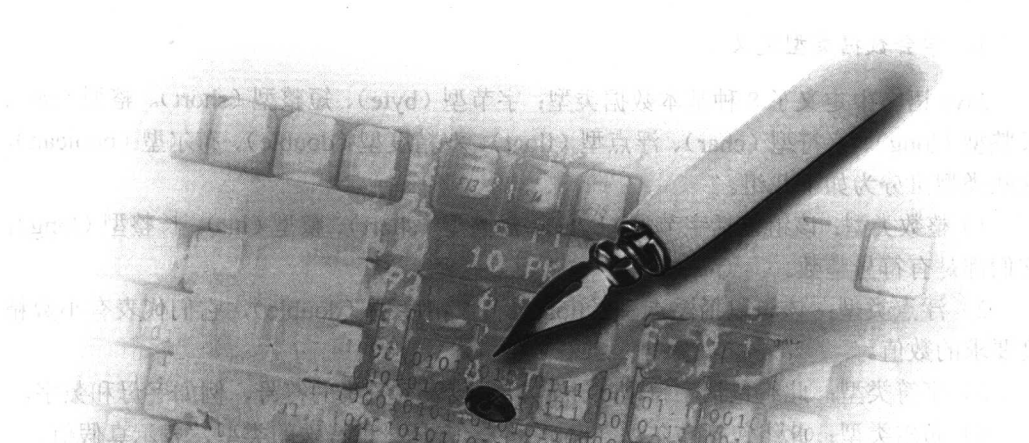
```
public class stars
{
    public static void main(String[] args)
    {
        final String s1="*****";
        final String s2=" *** ";
        final String s3=" * ";
        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s3);
    }
}
```

运行结果为：

五、实验回顾

本节实验主要介绍了 Java 开发工具包的安装，并通过两个简单的小程序掌握 Java 程序的编辑、编译和运行过程。读者可以初步建立对 Java 语言应用程序的整体概念，这样在以后的探索中就不再陌生了。

六、调试分析及实验心得



实验二 Java 语言基础



开篇论

Java 程序设计语言同大多数程序设计语言一样,具有一些基本特性。本节实验主要练习 Java 编程语言的这些基本特性,如 Java 中的简单数据类型,运算符和表达式,这是学习 Java 语言必须经过的基础阶段。掌握了本节实验所讲述的内容后,应该能够对 Java 程序的构成有一个基本的理解,能够灵活运用 Java 语言提供的各种数据类型和运算符。学好 Java 语言必须具备扎实的语言基础,这对后续学习是很有必要的。

一、实验目的

- 1) 掌握 Java 语言的数据类型。
- 2) 掌握 Java 语言的运算符和表达式。
- 3) 掌握 Java 语言运算符的优先级。
- 4) 进一步熟悉 Java 程序的编辑、编译和运行过程。

二、实验内容

- 1) 学会数据类型定义。
- 2) 利用输出语句来实现数据的格式化显示。
- 3) 使用简单的运算符和表达式实现给定的任务。

三、实验步骤

Java 语言源程序是由 Java 语言的基本语句构成的,Java 语言的语句基本上同 C/C++ 语言的语句一样。有些在 C/C++ 语言程序中容易产生错误的语句已被 Java 语言所抛弃。通过下面的实验步骤,读者可以编写简单的 Java 程序。

1. 学会数据类型定义

Java 语言中定义了 8 种基本数据类型：字节型 (byte)、短整型 (short)、整型 (int)、长整型 (long)、字符型 (char)、浮点型 (float)、双精度型 (double)、布尔型 (boolean)，这些类型可分为如下几组。

1) 整数类型：该组包括字节型 (byte)、短整型 (short)、整型 (int)、长整型 (long)，它们都是有符号整数。

2) 浮点类型：该组包括浮点型 (float) 和双精度型 (double)，它们代表有小数精度要求的数值。

3) 字符类型：此组包括字符型 (char)，它代表字符集的符号，例如字母和数字。

4) 布尔类型：此组包括布尔型 (boolean)，它是一种特殊的类型，表示真假值。

上面的数据类型也可成为简单数据类型。Java 中还有复合数据类型包括 class、interface 和数组。



简单数据类型是 Java 中定义的数据类型，也称内构的数据类型。通常是不能由用户修改的。而复合数据类型是用户根据自己的需要定义并实现其运算的类型，它是由简单数据类型组合起来形成的，包括类 (class)、接口 (interface) 和数组 (array)。

常量定义用保留字 final 来实现，格式如下：

```
final typeSpecifier varName=value[,varName[=value]...];
```

如：

```
final int NUM=100;
```

变量定义格式如下：

```
typeSpecifier varName[=value[,varName[=value]...];
```

如：

```
int count;
char c='a';
```

例 2-1 在横线上填出下列变量的定义类型。

```
public class example1{
    public static void main(String args[ ] ){
        _____ x=45;
        _____ y=3.14f;
        _____ z=3.1415;
        _____ truth=true;
        _____ c='6';
        .....
        .....
    }
}
```

2. 利用输出语句来实现数据的格式化显示

Java 语言中的输出语句，可以通过调用 System.out 中的 println 方法(或 print 方法)，