21世纪高等学校计算机科学与技术规划教材

JAVA

程序设计上机指导与习题选解

Java Chengxu Sheji Shangji Zhidao yu Xiti Xuanjie

主编 潘 浩

北京邮电大学出版社
www.buptpress.com

21世纪高等学校计算机科学与技术规划教材

Java程序设计上机指导与习题选解

主　编　潘　浩
副主编　张国英　张世博

北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书作为《Java 程序设计教程》的配套实验教材，基于 Java SE 6 开发平台，进行 Java 程序的设计与开发。本书共有 11 章，包括 Java 编程的基本语法、面向对象编程、数组、Java 图形用户界面编程、多线程、文件与输入/输出流、数据库编程等基础知识。每一章先通过几个典型的案例讲述本章的知识，对于每个案例，都提供了案例的相关知识、源代码、对代码的分析和解释、案例的运行结果分析。然后给出 Java 实验的内容以及每个实验的相关知识和注意事项。最后提供了大量的章后习题供编程者实践。

本书可以作为高等院校 Java 语言上机实验课程的指导教材，也可以作为初步掌握 Java 编程知识的学习者的知识巩固和加强用书。

图书在版编目(CIP)数据

Java 程序设计上机指导与习题选解/潘浩主编. —北京：北京邮电大学出版社，2008

ISBN 978-7-5635-1646-9

Ⅰ. J… Ⅱ. 潘… 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(2008)第 114199 号

书　　名 Java 程序设计上机指导与习题选解

主　　编 潘　浩

责任编辑 沙一飞

出版发行 北京邮电大学出版社

社　　址 北京市海淀区西土城路 10 号(100876)

电话传真 010-62282185(发行部)　010-62283578(传真)

电子信箱 ctrd@buptpress.com

经　　销 各地新华书店

印　　刷 北京忠信诚胶印厂

开　　本 787mm×1 092mm　1/16

印　　张 13.25

字　　数 295 千字

版　　次 2008 年 10 月第 1 版　2008 年 10 月第 1 次印刷

ISBN 978-7-5635-1646-9　　定价：19.00 元

前 言

本书作为《Java 程序设计教程》的配套实验教材，也同样可以作为采用其他 Java 教材的实验教材。Java 语言具有跨平台、面向对象、安全、可靠以及适用于网络等显著特点，是目前最为流行的网络编程语言之一。各高校都开设了 Java 语言课程，作为计算机相关专业的专业课和全校的编程基础课。Java 语言作为一门编程语言，学习它的最终目的是使用 Java 语言进行系统的开发，来解决实际问题。因此它是一门实践性很强的课程，如何培养学生的 Java语言编程能力，使学生能够灵活地使用 Java 语言进行实际编程是当前面临的重要问题。

本书是对 Java 教材的补充，提供了大量新颖的和实用的编程案例帮助学生理解和掌握 Java 语言的每个知识点。书中的每个案例与 Java 的相关知识点结合起来，首先分析案例的解决方案，介绍与本案例相关的知识，提供案例的实现代码，对代码的关键部分进行注释，并且对程序的实现和运行结果进行了详尽的解释。本书将知识点的讲解和习题的编程技巧融于案例之中，使学习者能够从实践中理解和巩固知识。

在本书的每章中，还提供了上机实验部分，阐述了本章的实验目的和实验要求，给出实验的内容，对每个实验题目还给出了编程要求和相关知识的提示。本书还提供了丰富的习题，供学习者进行有效的实践。

本书共有 11 章。第 1 章讲述了 Java 编程环境的安装和配置。第 2 章讲述了 Java 编程的基本语法知识。第 3 章提供了 Java 的分支语句和循环语句的编程实例和相关知识。第 4 章讲述了 Java 面向对象编程基本知识。第 5 章讲述了 Java 的继承、接口和包的相关知识。第 6 章讲述了 Java 数组编程和与数组相关的 Arrays 类和 ArrayList 类的编程实例。第 7 章讲述了 Java 语言中的常用类，包括 String 类、Math 类、Java 基本数据类型包装类和异常处理机制。第 8 章讲述了图像用户界面编程实例和相关知识。第 9 章讲述了输入/输出流和文件读写的相关知识。第 10 章讲述了 Java 多线程的实现原理。第 11 章讲述了使用 JDBC 访问数据库的相关知识。

本书由潘浩、张国英和张世博编写，张国英负责本书第 2 章～第 7 章的编写，潘浩负责对第 1 章、第 8 章和第 9 章的编写，张世博编写了本书的第 10 章和第 11 章。本书最后由潘浩进行统稿。沐春杰和马松岩参与了本书的程序测试工作，提供了本书的部分习题和答案。在这里，首先要感谢易久教授，没有他的帮助，就没有本书的出版。另外，还要感谢香港浸会大学的王大震博士为本书的编写提供的宝贵意见，感谢董琳媛教授和孙滨丽教授给予的诸多支持和帮助。

本书难免存在一些遗漏和不足之处，恳请读者批评指正，给出修改建议。可以通过邮箱“bjpanhao@163.com”直接与作者联系。

作 者

2008 年 3 月

目　录

第 1 章　Java 编程环境

1.1　JDK 的安装与配置

在编写 Java 程序之前，首先需要进行 Java 编辑、编译和运行环境的安装和配置。JDK 是进行 Java 编程的基本工具包，它是 Sun 公司免费提供的 Java 程序开发和运行的工具软件。Java 开发人员可以利用这个工具来开发、编译和运行 Java 程序。

1. JDK 的安装

JDK 安装程序的最新版本可以从网站“http://java.sun.com”上下载，本书使用了 JDK 1.6 版本，文件名为“jdk-1_6_0_10-windows-i586-p.exe”。双击该文件进行 JDK 的安装，安装的默认目录为“C:\Program Files\Java\jdk1.6.0_10\”。本教材中 JDK 的安装目录为“C:\ Java\jdk1.6.0_10\”。JDK 安装完毕后，在安装目录中生成了许多子目录和文件，列举如下：

① bin 子目录：包含 Java 开发工具，包括 Java 编译器、解释器等可执行文件。

② demo 子目录：包含一些实例程序，包括 Java 应用程序和 Java Applet 的源程序。

③ lib 子目录：包含 Java 开发类库，提供了已经实现的功能。

④ jre 子目录：包括 Java 运行环境相关文件，包括 Java 虚拟机、运行类库等。

⑤ include 子目录：包含与 C 语言相关的头文件。

在安装目录下还包含一个压缩文件“src.zip”，它提供了 Java 函数库的源代码，这是学习 Java 类库的很好途径。在源文件中可以查阅 Java 类库中任何类的信息，包括该类拥有的变量和方法。随着对 Java 学习的深入，一些教材和简单的联机信息不能提供的 Java 类库信息，也可以从“src.zip”中获得。

bin 子目录中包含对 Java 程序进行编译和运行的可执行文件，主要包括：

(1) Javac

Java 编译器，用来将 Java 源程序编译成字节码(Bytecode)形式的二进制类文件。Java 编译命令的格式为：

javac [option]文件名.java

其中，option 为可选项，可以实现一些功能。例如，“-d dir”表示编译后产生的类文件存放在 dir 指定的目录下。

(2) Java

Java 解释器，能够解释执行编译后产生的字节码文件。Java 运行命令的格式为：

java [-debug -classpath …]文件名

其中，括号里的为可选项。例如，-debug 表示在调试模式下启动解释器。在调试模式下，Java 提示输入口令，只有输入相应口令后，才能进入调试阶段。

注意：运行命令中文件名不能加后缀“.class”，如要执行文件“Demo.class”，命令为“java Demo”。

(3) Jdb

Java 调试器，是用来调试 Java 应用程序的命令行调试器。命令格式为：

jdb [option][class][arguments]

其中，option 为可选项，可以实现一些功能；class 表示开始调试的类名；arguments 表示执行类中的 main()方法时输入的开关字符串变量，保存在字符串数组 args[]中。

(4) Javap

反编译，将 Java 类文件还原回方法和变量。

(5) javadoc

文档生成器，可以直接从 Java 源代码生成 API 文档。文档生成器通过对 Java 源文件进行分析，生成有说明内容的 HTML 文件。

(6) Appletviewer

Applet 解释器，用来解释运行已经编译成字节码的 Java Applet。Java Applet 不能单独运行，必须嵌入在 HTML 文件中。Appletviewer 解释的文件是包含引用 Java Applet 的 HTML 文件。

2. JDK 的环境配置

JDK 安装完毕后，还需要进行系统环境变量 Path 和 classpath 的配置，以便在 Java 程序编译和运行时，由系统自动定位 Java 类库和调试、运行等工具，不需要手工键入这些工具文件所在的路径。环境变量 Path 中包含 JDK 的 Java 开发工具(bin 子目录)的路径，而 classpath 的值为 Java 类库的路径和程序需要使用的类的路径。

环境变量的配置可以通过编写批处理文件、在控制台直接键入命令、在 Windows 图形界面中直接配置的方法进行。最后一种方法是最容易实现的，即用鼠标右键单击“我的电脑”图标，在弹出的快捷菜单中选择“属性”命令，打开“系统属性”对话框。单击“高级”选项卡，在出现的界面中单击“环境变量”按钮，弹出“环境变量”对话框，如图 1-1 所示。

从图中可以看出，Windows 的系统变量中原来就有 Path 环境变量，只需在 Path 变量值中添加 bin 目录的路径就可以了。在系统变量中选择 Path 变量，用鼠标单击“编辑”按钮，添加路径为“C:\Java\jdk1.6.0_10\bin”。Path 值包含多个路径，路径之间用分号(;)分隔，如图 1-2 所示。

环境变量 classpath 的值为程序加载的类的路径，它的配置需要多加注意，因为以后对程序进行编译时，如果出现一些莫名其妙的编译错误，80%以上都可能是由于环境变量 classpath 设置错误引起的。用鼠标单击“新建”按钮，在弹出的“编辑系统变量”对话框中的“变量名”文本框中输入“classpath”，“变量值”中输入“.;C:\Java\jdk1.6.0_10\lib\dt.jar;C:\Java\jdk1.6.0_10\lib\tools.jar;”，共 3 项值，以分号分隔。需要注意的是最前面的“.”，它代表当前目录。如图1-3所示。

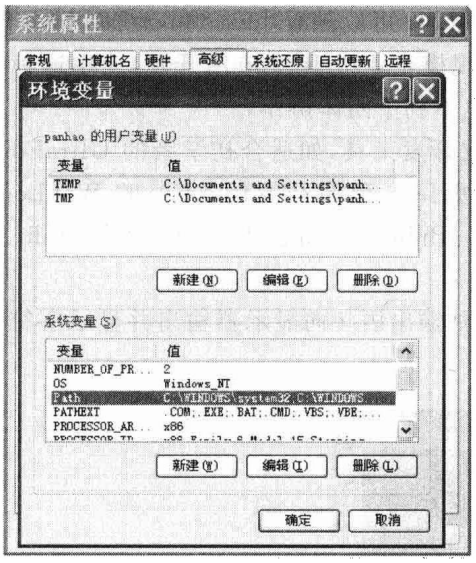

图 1-1　“环境变量”对话框

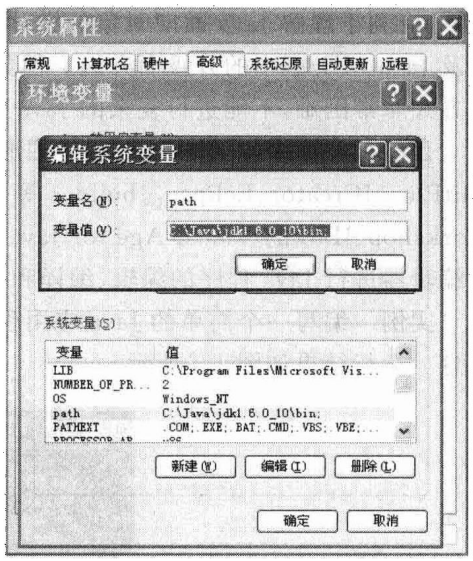

图 1-2　系统变量 Path 的配置

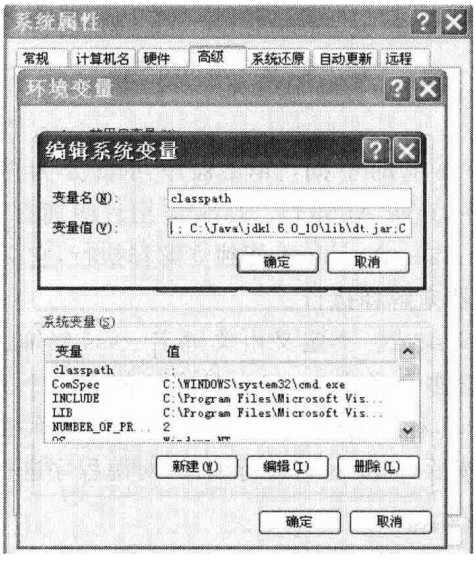

图 1-3　系统变量 classpath 的配置

1.2　Java 程序的编辑、编译和运行

JDK 安装和配置完毕后，Java 程序的编译和运行环境就建立了，接下来可以进行 Java 程序的编辑。Java 源程序可以在任何的文本编辑工具中编写，如 Windows 操作系统中的记事本。编辑完毕后，需要将程序保存为后缀为“.java”的文件，然后在 DOS 命令环境下进行 Java 程序编译和运行。

从初学者角度来看，采用 JDK 开发 Java 程序能够很快理解程序中各部分代码之间的

关系，有利于理解 Java 面向对象的设计思想。并且 JDK 可以随着 Java (Java EE、Jave SE 以及 Java ME)版本的升级而升级。但如果从事大规模企业级 Java 应用开发，使用这种开发方式非常困难，不能进行复杂的 Java 软件开发，也不利于团体协作。

目前存在许多优秀的 Java 程序图形界面的集成开发工具，如适合初学者的 UltraEdit、EditPlus、JCreator、Eclipse，Jbuilder 等，进行企业级 Java 程序开发的 BEA 的 WebLogic Workshop、IBM 的 Visual Age for Java、Borland 的 JBuilder 等等。本书使用 UltraEdit 文本编辑器进行 Java 程序的编辑、编译和运行。

实例 编写一个简单的 Java 应用程序，使用多种输出语句向显示器输出并显示字符串信息。运行结果如图 1-4 所示。

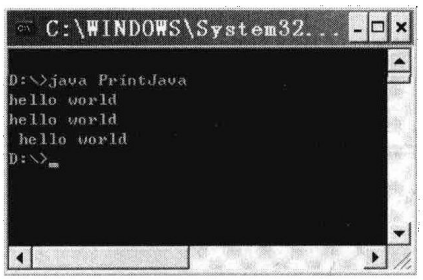

图 1-4 实例输出结果图

相关知识：

根据编程结构和运行环境的不同，Java 程序可以分为两类：Java 应用程序(Java Application)和 Java 小程序(Java Applet)。Java 应用程序拥有 main()方法作为程序运行的入口，能够独立解释执行；Java Applet 不能独立解释执行，它必须嵌入在 HTML 文件中，由内置 Java 解释器的浏览器来解释执行。

Java 应用程序由类定义组成。类定义由关键字 class 修饰，类体部分用大括号({})括起来。一个程序中可以包含多个类定义，但是最多只能有一个类被关键字 public 所修饰。Java 源程序的后缀为“.java”，程序的命名必须与公共类的名称相同。

Java 应用程序编写完毕后，需要对它进行编译，对源程序进行编译的命令为：

javac [option] 文件名.java

其中，option 为可选项。编译后的文件是后缀为“.class”的二进制字节码文件。程序编译后产生的字节码文件与程序中定义的类的数量相同，字节码文件依次命名为程序中定义的类名。

Java 语言是一种解释性语言，与编译型语言不同，编译生成的字节码文件不是可执行程序，必须由 Java 运行程序来解释执行。对字节码文件进行运行的命令为：

java [-debug -classpath…] 文件名

注意：运行命令中的文件名不带后缀。

程序代码：

```
1    public class PrintJava
2    {
```

```
3       public static void main(String args[])
4       {
5           System. out. print("hell");
6           System. out. print("o world");
7           System. out. println();
8           System. out. println("hello world");
9           System. out. printf("%6s %5s","hello","world");
10  }
11  }
```

程序说明：

(1) 程序的功能是利用多种输出方法向显示器输出多个字符串。第 1 行语句定义了一个 PrintJava 类，它被关键字 class 修饰，表明是类的定义。PrintJava 类定义被关键字 public 所修饰，所以源程序的名称为“PrintJava. java”。一个 Java 程序中可以定义多个类，但最多只能有一个公共类。

(2) 第 3 行语句在类中定义了一个 main()方法，它作为程序运行的入口。main()方法头的定义必须为“public static void main (String args[])”。public 表示方法的访问权限是公共的，可供所有类访问；static 表明 main()方法是一个静态方法，它可以通过类名直接调用；void 表明 main()方法没有返回值。main()方法的参数类型是字符串数组类型，该参数又称为命令行参数，用来将用户在运行时输入的参数传递给主方法。

(3) 第 5 行语句中的“System. out”表示标准输出，即显示器。它的类型为 PrintStream，是“java. io”包中的类。调用标准输出流的 print()方法向显示器输出字符串 "hell"，输出完毕后并不换行。

(4) 第 7 行语句调用 println()方法，实现换行的功能。

(5) 第 8 行语句调用 println()方法，向显示器输出字符串"hello world"，输出完毕后换行。

(6) 第 9 行语句调用 printf()方法，向显示器输出格式化的数据。方法中的第 1 个参数表示要输出的数据格式为两个字符串，这两个字符串之间有一个空格。%6s 表示输出的是字符串类型，字符串的长度为 6。第 2、3 个参数表示要显示的字符串的内容。

1.3　制作 JAR 文件包

在进行软件开发中，程序编写、测试完毕后，需要将程序包装起来，制作成为软件安装包，以便于软件的发布，使用户能够方便地安装和部署程序中的类。有多种方法可以实现这一功能，如可以使用 JET 对程序进行处理。这里使用 jar 命令将程序和相关资源文件制作成为一个 JAR 文件，便于程序的发布和运行。

JAR (Java Archive File) 文件称为 Java 存档文件，它是一种与平台无关的文件格式，能够将许多文件组合成一个文件包。JAR 文件可以用于发布和使用类库，还可作为应用程序、扩展的构建单元和组件、Applet 或者插件程序的部署单位。JAR 文件格式以流行的

ZIP 文件格式为基础。与 ZIP 文件不同的是，JAR 文件不仅用于压缩和发布，而且还用于部署和封装库、组件和插件程序，并可以被 JVM 直接使用。并且，在 JAR 中包含了一个 META-INF/MANIFEST. MF 文件，是在生成 JAR 文件的时候自动创建的，用来指示工具如何处理特定的 JAR 文件。例如，要想制作成可执行的 JAR 文件，需要在 META-INF 的 MANIFEST 文件制定 Main-Class。这样可以通过命令行“java -jar jarfile. jar”来执行 JAR 文件，由于 Windows 操作系统默认将后缀为“. jar”的文件使用 javaw -jar 打开，所以在 Windows 操作系统下可以直接双击 JAR 文件运行软件。

JAR 文件的生成和运行通过 jar 命令来实现，jar 命令的语法如下：

jar {ctxu}[vfm0M] [jar-文件] [manifest-文件] [-C 目录] 文件名…

其中，{ctxu}是 jar 命令的子命令，每次 jar 命令只能包含“ctxu”中的一个，它们分别表示：

① c：创建新的 JAR 文件包。

② t：列出 JAR 文件包的内容列表。

③ x：展开 JAR 文件包的指定文件或者所有文件。

④ u：更新已存在的 JAR 文件包。

[vfm0M]中的选项可以任选，也可以不选，它们是 jar 命令的选项参数，分别表示：

① v：生成详细报告并打印到标准输出。

② f：用于指定 JAR 文件名，通常这个参数是必须的。

③m：指定需要包含的 MANIFEST 清单文件。

④0：不压缩 JAR 文件的内容。

⑤M：不产生所有项的清单，此参数会忽略 m 参数 。

[jar-文件]即需要生成、查看、更新或者解开的 JAR 文件包，它是 f 参数的附属参数。

[manifest-文件]即 MANIFEST 清单文件，它是 m 参数的附属参数。

[-C 目录]表示转到指定目录下去执行这个 jar 命令的操作。它相当于先使用 cd 命令转到该目录下再执行不带-C 参数的 jar 命令，它只能在创建和更新 JAR 文件包的时候可用。

文件名…用来指定一个文件/目录列表，这些文件/目录就是要添加到 JAR 文件包中的文件/目录。如果指定了目录，那么 jar 命令打包的时候会自动把该目录中的所有文件和子目录打入包中。

需要注意的是，在运行 jar 命令之前，首先需要确定要打包的文件存放在文件系统的相应层次的文件夹中。这样，创建的 JAR 文件包中包含完整的、与 Java 程序的包结构对应的目录结构。对于 Main-Class 指定的类，也必须是完整的、包含包路径的类名。

下面是 jar 命令的使用方法的例子。

(1) jar cf hello. jar hello

该命令的执行结果是在当前目录生成了“hello. jar”文件，将 hello 目录添加到“hello. jar”文件包中。如果当前目录已经存在“hello. jar”，那么该文件将被覆盖。

(2) jar cvf hello. jar hello

该命令与(1)中的结果相同，但是由于 v 参数的作用，所以在屏幕上显示出了打包过程。

(3) jar 0cf hello.jar hello image

该命令将 hello 目录和 image 目录的文件全部包装到“hello.jar”文件包中，并且不压缩“hello.jar”的内容。

(4) jar tf hello.jar

该命令列出“hello.jar”的内容列表。

(5) jar xf hello.jar

该命令解压缩“hello.jar”的内容，将解压后的文件存放到当前目录。

通过 jar 命令可以方便地将编写的 Java 程序进行打包。除了操作方便外，JAR 文件格式还具有许多其他优点。

① 安全性：通过在 JAR 文件内容中加上数字化签名，使签名识别工具可以有选择地为用户授予软件安全特权，它还可以检测程序代码是否被篡改过。

② 文件压缩功能：可以对 JAR 文件进行压缩以提高存储效率。

③ 传输平台扩展功能：Java 扩展框架(Java Extensions Framework)提供了向 Java 核心平台添加功能的方法，这些扩展是用 JAR 文件打包的。

④ 包密封操作：存储在 JAR 文件中的包可以选择进行密封，以增强版本一致性和安全性。

⑤ 包版本控制：一个 JAR 文件可以包含有关它所包含的文件的数据，如厂商和版本信息。

⑥ 可移植性：JAR 文件的机制是 Java 平台核心 API 的标准部分，这使得 JAR 文件具有跨平台特性。

1.4　上机实验

一、实验目的

1. 掌握 JDK 的安装过程以及环境变量的配置。
2. 掌握 UltraEdit 工具的安装及 Java 编译和运行命令的配置。
3. 掌握 Java 应用程序的编译和运行命令。
4. 理解 Java Applet 的编译和运行命令。

二、实验内容和要求

1. 尝试进行 JDK 的安装，注意设置 JDK 的安装路径，并进行环境变量 classpath 和 Path 的配置。

2. 尝试进行 Ultraedit 的安装，并在 UltraEdit 中“advanced”菜单的“tool configuration”选项中进行 Java 编译、Java 运行和 Applet 运行命令的配置。

3. 下面是一个简单的 Java 程序，功能是使用标准输出“System.out”的 println()方法输出字符串。请为程序命名，并编译运行程序，观察程序的运行结果。

```
public class Hello{
    public static void main (String args[])
    {
        System. out. println("您好,请观察下一条语句的运行情况!");
        System. out. printf("%4d, %5s," 400, "hello");
    }
}
```

4. 编写简单的 Java 程序,实现从键盘接收一个字符串的输入,并将该字符串显示在显示器上。

习题

一、选择填空

1. (1) Java 程序必须以_____为文件扩展名。
   (2) Java SE 开发工具包中的_____命令负责编译 Java 应用程序。
   (3) Java SE 开发工具包中的_____命令负责运行 Java 应用程序。
   (4) 当编译一个 Java 程序时,编译器创建的文件以_____为文件扩展名。
2. 只包含一个带 public 修饰的类"Hello"的 Java 程序的正确命名是(　　)。
   A、Hello　　B、Hello. java
   C、hello. java　　D、一个任意的标识符. java
3. Java 应用程序入口的 main()方法,其声明格式可以是(　　)。
   A、public static int main(String args[])
   B、public static void main(String args[])
   C、public void main(String args[])
   D、public int main(String args[])
4. Java 程序编译后生成的文件是(　　)。
   A、字节码　　B、Espresso　　C、机器码　　D、二进制码
5. Java Applet 源程序文件的扩展名为(　　)。
   A、. java　　B、. exe　　C、. class　　D、. html
6. 一个包含 Java 类 A、B 的源程序"A. java",编译产生的类文件是(　　)。
   A、A. class　　B、A. class, B$. class
   C、A. class, B. class　　D、A$. class, B$. class
7. 编译 Java Application 源程序文件将产生相应的字节码文件,这些字节码文件的扩展名为(　　)。
   A、. java　　B、. html　　C、. exe　　D、. class
8. 下面说法正确的是(　　)。
   A、在命令行运行编译好的字节码文件,只需在命令行直接键入程序名即可运行该程序

B、一个 Java 源程序编译后可能产生几个字节码文件

C、Java 程序的源文件名称与主类的名称相同，后缀可以是“.java”或“.txt”

D、JDK 的编译命令是 java

9. 下列程序有哪种可能的结果？（　　）

```
public class Hello{
  public void main(String [] args)
  {  System.out.println("Hello World");
   }
  }
```

A、程序不能编译　　B、运行时产生异常

C、运行后输出“Hello World”　　D、运行后不输出任何结果

10. 如果一个 Java Applet 源程序文件只定义了一个类，该类的类名为“MyApplet”，则类“MyApplet”必须是_____类的子类，并且存储该源程序文件的文件名是_____。

11. Java 支持 3 种类型的注释，分别是_____、_____、_____。

12. 能够生成 Java 文档注释的命令是_____。

二、简答题

1. 什么是 Java 程序的基本类型？各有何特点？

2. 如何建立和运行 Java 程序？

3. 环境变量 classpath 和 Path 的作用分别是什么？

4. Java 技术体系主要由哪 3 部分组成？各自用在什么场合？

5. 下面“sayHello.java”程序是属于哪种类型的 Java 程序，它的运行结果是_____。

```
    import java.awt.Graphics;
    import java.applet.Applet;
public class sayHello extends Applet{
    public void paint(Graphics g){
        g.drawString("Hello Java!",35,30);
    }
}
```

三、编程题

1. 编写一个输出“I like java ”的程序，用两种方式实现（Application、Applet）。

2. 编写一个程序，使用 printf()方法显示各种类型的数据。

第 2 章　Java 基本数据类型

2.1　Java 基本数据类型应用实例

实例 1　本例实现几种基本数据类型的转换，从而比较自动数据类型转换和强制数据类型转换的区别。本例的运行结果如图 2-1 所示。

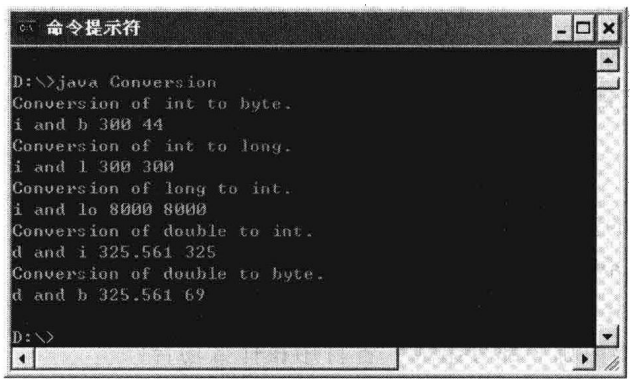

图 2-1　实例 1 输出结果图

相关知识：

Java 编程语言有 8 种基本数据类型。

1. 布尔类型 boolean

boolean 数据类型只有两种值：true 和 false。

在 Java 编程语言中，boolean 类型只允许使用 boolean 值，在整数类型和 boolean 类型之间无转换计算。在 C 语言中允许将数字值转换成逻辑值，这在 Java 编程语言中是不允许的。

2. 字符类型 char

使用 char 类型可表示单个字符，字符是用单引号括起来的一个字符，如'a'。Java 中的字符型数据是 16 位无符号型数据，它表示 Unicode 集，而不仅仅是 ASCII 集。与 C 语言类似，Java 也提供转义字符，以反斜杠(\)开头，将其后的字符转变为另外的含义。常用转义符如表 2-1 所示。

表 2-1　转义符及其含义

转义字符	含义
\n	换行，将光标移至下一行的开始