

高职高专

现代信息技术辅导教材

Java 程序设计

——示例与习题解析

朱喜福 马 涛 魏绍谦 编



人民邮电出版社
POSTS & TELECOM PRESS

高职高专现代信息技术辅导教材

Java 程序设计——示例与习题解析

朱喜福 马 涛 魏绍谦 编

人民邮电出版社

图书在版编目 (CIP) 数据

Java 程序设计: 示例与习题解析 / 朱喜福, 马涛, 魏绍谦编.

—北京: 人民邮电出版社, 2004.8

高职高专现代信息技术辅导教材

ISBN 7-115-11904-X

I. J... II. ①朱... ②马... ③魏... 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 (2004) 第 068062 号

内 容 提 要

本书内容详略得当、重点突出, 即可作为 Java 程序设计的辅导教材, 也可供自学使用。

本书分 8 章, 主要介绍 Java 语言基础, 运算符、表达式与控制结构, Java 面向对象程序设计, 异常, 多线程, Java 的输入输出处理, Java 图形用户界面和 Java Applet。

本书可作为高职高专计算机及相关专业的教材, 也可供从事实际应用开发的技术人员参考学习。

高职高专现代信息技术辅导教材

Java 程序设计——示例与习题解析

◆ 编 朱喜福 马 涛 魏绍谦
责任编辑 潘春燕

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67129259
北京汉魂图文设计有限公司制作
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

◆ 印张: 19.5

字数: 474 千字

2004 年 8 月第 1 版

印数: 1-5 000 册

2004 年 8 月北京第 1 次印刷

ISBN 7-115-11904-X/TP · 3736

定价: 25.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

编者的话

本书是人民邮电出版社出版的《Java 程序设计》一书的配套学习辅导书，全书对 Java 语言的基本概念和各个知识点用大量的实例程序进行了深入且通俗易懂的讲解，书中共有例题和习题近 800 个，包含各种不同的题型，如选择题、填空题、简答题和编程题等，用来帮助读者反复强化基本概念、知识点及编程思路和方法。作者根据几年来教学实践中所积累的经验，参考了大量的书籍，从内容的取舍、题型的设计，以及知识结构的安排上下了很大的工夫。例题讲解和练习安排都有很强的目的性，非常适合学生自学，为学习 Java 语言提供了很好的辅助材料，通过知识点和相关例题的学习，再配合书中给出的各种题型进行大量的练习，相信能使学生对 Java 语言及其相关技术有深入的理解，对 Java 编程思维方法的掌握也会有极大的帮助。作者相信，学习任何知识，尤其是计算机编程语言，训练编程的思维方法，只有通过大量的练习和编程实践才是掌握它们的惟一途径。

本书内容丰富，重点突出，便于自学，涉及的内容和习题的编排与《Java 程序设计》一书基本上一致。

书中各章根据知识点给出了丰富的例题讲解，并对每一个知识点给出了大量的练习，方便读者学完每一章后进行巩固练习。为了便于参考和使用，书中的习题都给出了参考答案，可供读者参考。对于编程题，其解答是不惟一的，因此读者也可以通过多种思路和方法来解决它们，但由于篇幅所限，书中只给出一种参考答案。

参与本书编写的人员还有操静涛、翟红英、王永平、李湛、屈敬文、王波、祝铭玉和关忠等。由于编者水平所限，错误在所难免，敬请读者批评指正。

朱喜福
2004 年 6 月

目 录

第1章 Java语言基础	1
1.1 基本概念及知识点	1
1.1.1 两类Java程序	1
1.1.2 Java程序的编辑、编译和运行	2
1.1.3 Java语言的注释和分隔符	3
1.1.4 Java语言的标示符和关键字	3
1.1.5 Java语言的数据类型	4
1.1.6 常量	5
1.1.7 变量	6
1.2 典型例题分析	7
1.3 习题	9
1.4 习题答案	15
第2章 运算符、表达式与控制结构	18
2.1 基本概念及知识点	18
2.1.1 Java语言的运算符与表达式	18
2.1.2 语句	20
2.1.3 Java语言的控制结构	21
2.1.4 数组	25
2.1.5 方法	27
2.1.6 方法的重载	28
2.1.7 应用程序命令行传参数	28
2.2 典型例题分析	29
2.3 习题	38
2.4 习题答案	50
第3章 Java面向对象程序设计	65
3.1 基本概念及知识点	65
3.1.1 面向对象程序设计的基本概念	65
3.1.2 类	66
3.1.3 类的继承	67
3.1.4 Java语言的修饰符	69
3.1.5 接口	70
3.1.6 内部类和匿名类	70

3.1.7 Java 语言的包 (package)	71
3.1.8 Java 的 API 与类库	71
3.1.9 Math 类	71
3.1.10 字符串(String)类	72
3.1.11 StringBuffer 类	76
3.1.12 StringTokenizer 类	78
3.1.13 Vector 类	78
3.2 典型例题分析	80
3.3 习题	98
3.4 习题答案	116
第4章 异常	126
4.1 基本概念及知识点	126
4.1.1 Java 异常	126
4.1.2 Java 异常处理	127
4.1.3 自定义异常	129
4.2 典型例题分析	129
4.3 习题	134
4.4 习题答案	138
第5章 多线程	140
5.1 基本概念及知识点	140
5.1.1 多线程	140
5.1.2 线程的状态和生命周期	141
5.1.3 多线程实现的两种方法	141
5.1.4 多线程的基本控制	142
5.1.5 多线程 Applet 小程序	143
5.1.6 线程的互斥锁和线程的同步	144
5.2 典型例题分析	145
5.3 习题	156
5.4 习题答案	162
第6章 Java 的输入输出处理	171
6.1 基本概念及知识点	171
6.1.1 概述	171
6.1.2 文件输入流 (FileInputStream) 和文件输出流 (FileOutputStream)	172
6.1.3 File 类	173
6.1.4 文件的随机访问 (RandomAccessFile 类)	175

6.1.5 过滤流	177
6.1.6 数据流 (DataInputStream 类和 DataOutputStream 类)	178
6.1.7 BufferedInputStream 类和 BufferedOutputStream 类	179
6.1.8 回压输入流 (PushbackInputStream)	181
6.1.9 字节数组流 (ByteArrayInputStream 类和 ByteArrayOutputStream 类)	181
6.1.10 SequenceInputStream 类	182
6.1.11 对象串行化和对象流	183
6.1.12 管道流 (PipedOutputStream 类和 PipedInputStream 类)	184
6.2 典型例题分析	185
6.3 习题	207
6.4 习题答案	218
第 7 章 Java 图形用户界面	228
7.1 基本概念及知识点	228
7.1.1 文本、字体和颜色	228
7.1.2 Java 图形	230
7.1.3 图像显示	232
7.1.4 标签	233
7.1.5 按钮	234
7.1.6 单行文本域	235
7.1.7 多行文本域	237
7.1.8 容器和组件	238
7.1.9 面板和画布	239
7.1.10 布局	240
7.1.11 复选框和单选框	243
7.1.12 列表	244
7.1.13 鼠标事件	246
7.1.14 键盘事件	247
7.1.15 窗口事件	248
7.2 典型例题分析	250
7.3 习题	275
7.4 习题答案	276
第 8 章 Java Applet	290
8.1 基本概念及知识点	290
8.1.1 Applet 的生命周期及基本方法	290
8.1.2 向 Applet 传递参数	291
8.1.3 判定 Applet 的来源	291
8.1.4 下载图像	291

8.1.5 下载声音	291
8.1.6 动画	291
8.2 典型例题分析	292
8.3 习题	299
8.4 习题答案	301

第 1 章 Java 语言基础

1.1 基本概念及知识点

提要

- Java 程序的两种基本形式
- Java 程序的基本组成
- Java 语言的数据类型

1.1.1 两类 Java 程序

1. Java 应用程序

(1) Java 应用程序 (Java Application) 是一种能在支持 Java 的平台上独立运行的程序, 它是通过 Java 虚拟机 (Java Virtual Machine, JVM) 解释执行的。Java 应用程序的主类中一定包含 main() 方法, 它是应用程序的主进程, 即程序执行的入口。

(2) Java 应用程序的程序框架如下:

```
public class 类名
{
    public static void main(String args[])
    {
        .....//程序代码
    }
    //其他属性和方法定义
}
```

2. Java 小程序

(1) Java 小程序 (Java Applet) 是一种通过 <applet></applet> 标记符将字节码文件内嵌在 HTML 文档中, 通过支持 Java 浏览器才能运行的程序。小程序没有自己的主进程。

(2) 小程序的程序框架如下:

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

```

public class Applet1 extends Applet
{
    //变量或属性定义;
    public void init()//初始化方法
    {
        //...;
    }
    public void paint(Graphics g)//小程序显示结果的方法
    {
        //...;
    }
    //其他方法
}

```

(3) 嵌入小程序的 HTML 文档

小程序不能独立运行，它是通过支持 Java 的浏览器运行。因此，必须将小程序编译后形成的字节码（.class）文件利用标记符<applet></applet>嵌在 HTML 文档中。

```

<html >
<head><title>...</title></head>
<body>
<applet code = 类名.class width = 300 height = 150 >
</applet >
/body>
</html >

```

1.1.2 Java 程序的编辑、编译和运行

1. 编写Java源程序

Java 源程序的名称要与 Java 中的主类（公共类）名称一致。一个 Java 源程序可以定义多个类，但只能有一个公共类。保存文件的扩展名为.java。

2. Java程序的编译

在 JDK 安装目录的 bin 子目录下提供了很多编译、调试和运行 Java 程序的命令。

在命令行可通过 javac 将 Java 源程序编译为 Java 的字节码文件，编译后的文件名与源文件名相同，文件的扩展名为.class。一个 Java 源程序如果定义多个类，编译后会产生多个.class 文件。

3. 运行Java程序

应用程序的运行：通过 JDK 安装目录下的 bin 子目录中的 Java 命令读入字节代码，在本地计算机解释执行。

小程序的运行：小程序不能独立运行，要编写网页文件，通过标记符<applet></applet>把要运行的 Java 字节码文件嵌入网页文件中，通过浏览器运行。在命令行下，通过 bin 目录下的命令 appletviewer 运行网页即可。

1.1.3 Java 语言的注释和分隔符

1. Java语言的注释

多行注释: /* 注释部分 */ 或 /** 注释部分 */。

行注释: 以// 开始的本行的后面部分为注释内容, 用于单行的注释。

2. 几个重要的分隔符

大括号 { }——用来定义类体、方法体、复合语句和数组的初始化。

分号 ; ——是语句结束标志。

逗号 , ——区分方法的各个参数, 区分变量说明的各个变量。

冒号 : ——用于语句标号。

1.1.4 Java 语言的标识符和关键字

1. Java语言的标识符

标识符用来给自定义的变量、方法和类及类对象命名。

Java 语言的标识符必须由字母、下划线 (“_”) 或美元符号 (“\$”) 开始, 其余字符可以是上述 3 种符号或数字 (0~9)。由于 Java 语言使用 Unicode 字符集 (16 位), 所以字母包括下面几种:

- ‘A’~‘Z’之间的 26 个大写字符;
- ‘a’~‘z’之间的 26 个小写字符;
- Unicode 字符集中序号大于 0xC0 的所有符号, 如中文字符、日文字符、阿拉伯字符等都可用作 Java 语言的标识符。

标识符不能以数字开头, 不能用关键字给变量或对象命名。

2. Java语言的关键字

关键字又称保留字, 是 Java 中具有特定含义的标识符。用户只能按系统规定的方式使用。关键字一律用小写字母表示。Java 的关键字有 50 个, 按用途划分为如下几组。

- 用于数据类型:

boolean, byte, char, double, false, float, int, long, new, null, short, true, void, instanceof

- 用于语句:

break, case, catch, continue, default, do, else, for, if, return, switch, try, while, finally, throw, this, super

- 用于修饰:

abstract, final, native, private, protected, public, static, synchronized, transient, volatile

- 用于方法、类、接口、包和异常:

class, extends, implements, interface, package, import, throws

- 保留字:

const, goto

3. Java语言的命名规则

- 类名第一个字母大写, 如 System、Applet 等。
- 方法名第一个字母小写, 如 main()、print()、println()等。

- Java 语言中，类名和方法名都比较长，这样便于阅读程序。如果类名或方法名由多个单词构成，单词之间的第一个字母大写。
- Java 语言的关键字小写。
- 自定义的变量名或一个类的对象名一般小写。

1.1.5 Java 语言的数据类型

1. Java语言的数据类型包含简单数据类型和复合数据类型

图 1-1 列举了 Java 语言的数据类型。

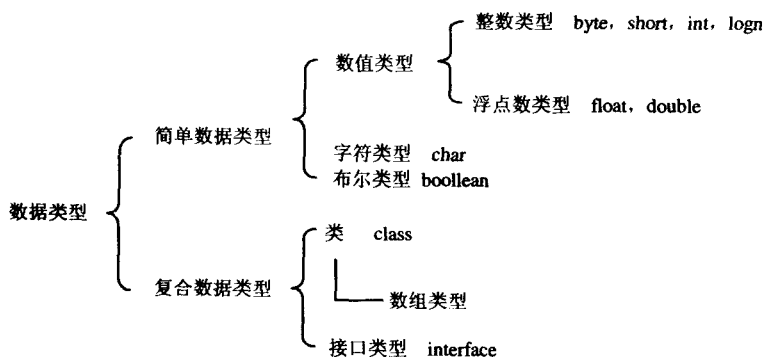


图 1-1 Java 语言的数据类型

2. Java语言的简单数据类型

Java 语言的简单数据类型也称为基本数据类型或原始数据类型。表 1-1 给出了各种基本数据类型及其取值范围。

表 1-1

Java 语言的基本数据类型

类型名称	关键字	占用位数	取值范围
字节型	byte	8	$-2^7 \sim 2^7 - 1$ ($-128 \sim 127$)
短整型	short	16	$-2^{15} \sim 2^{15} - 1$ ($-32768 \sim 32767$)
整型	int	32	$-2^{31} \sim 2^{31} - 1$
长整型	long	64	$-2^{63} \sim 2^{63} - 1$
单精度浮点型	float	32	大约 $-10^{38} \sim -10^{-45}$ 和 $10^{-45} \sim 10^{38}$
双精度浮点型	double	64	大约 $-10^{308} \sim -10^{-324}$ 和 $10^{-324} \sim 10^{308}$
字符型	char	16	0~65535 或 '\u0000'~'\uFFFF'
布尔型	boolean	8	true 和 false

说明

- (1) 字节型 (byte) 占用 8 位：取值范围 $-128 \sim 127$ ，赋值超出此范围会产生编译错误。
- (2) Java 语言中的字符型 (char) 占 16 位，编码采用 Unicode 编码。
- (3) Java 语言中的布尔型 (boolean) 只有两个取值：true 和 false，且不能和其他任何类型转换。

(4) 经常使用的字符串类型 (String) 是复合数据类型 (类类型), 但可以像简单数据类型一样赋值和相加。如果不给字符串变量赋值, 则其取值为 null。

1.1.6 常量

常量是在程序运行期间值不改变的量, 常量不能被程序修改。

在 Java 语言中, 定义常量通过 final 关键字实现。Java 的常量包括整型、实型、字符型及布尔型常量, 还有字符串常量。

1. 整型常量

整型常量默认为 int 类型, 用 4 个字节的存储单元存放。

要表示一个数为长整型, 需在这个数后面加上一个字母 L (或 l)。

八进制整数只能包含 0~7 这 8 个数字及正、负号, 而且必须以数字 0 为前导。

十六进制整数只能包含数字 0~9、字母 A~F 或 a~f、正、负号, 而且必须以 0x 为前导。

2. 实型常量

实型数据分单精度 (float) 和双精度 (double) 两种类型。单精度的实数占 4 个字节内存, 双精度实数占 8 个字节内存。

实型常量的小数形式, 由一个整数加上小数点, 然后加上小数部分组成。

在 Java 中, 无类型后缀的实型常量默认为双精度类型 (double), 也可加后缀 D 或 d; 指定单精度浮点类型常量, 必须在常量后面显式地加上后缀 F (或 f)。

实型常量也可表示为指数形式, 如双精度数: 2.1E8, 5.3e-9D; 单精度数: 9e-2f 等。其中的 8、-9、-2 分别表示指数。

3. 字符常量

字符常量是无符号的常量, 占 2 个字节内存, 其范围是 0~65535。

有以下 4 种表达方式。

- 用单引号括起来的字符。例如, 'a', '+', '-'。
- 用单引号括起来的转义序列, 以 "\" 开始表示转义, 例如, 制表符 '\t'、换行符 '\n'、反斜杠 '\\' 等。
- 用单引号括起来的八进制转义序列, 形式为 '\ddd', 此处的 ddd 表示 3 位八进制数。例如, '\141' 是字母 a。
- 用单引号括起来的 Unicode 转义字符形式为 '\U××××', 此处的 ×××× 为 4 位十六进制数。例如, '\U0061' 是字母 a。

表 1-2 列出了常用的转义字符。

表 1-2 常用的转义字符

转义字符	含 义
\n	换行, 将光标移至下一行的开始处
\t	水平制表 (Tab 键), 将光标移至下一个制表符位置
\b	光标退一格, 相当于 Backspace 键
\r	回车, 将光标移至当前行的开始, 不移到下一行

续表

转义字符	含 义
\\	反斜杠，输出一个反斜杠
'\"'	单引号，输出一个单引号
\"	双引号，输出一个双引号

4. 布尔常量

布尔常量仅有两个值：`true` 和 `false`，分别代表布尔逻辑中的“真”和“假”。

在 Java 语言中，布尔常量不能转换成任何数据类型，`true` 常量不等于 1，而 `false` 常量也不等于 0。

5. 字符串常量

字符串 (`String`) 不是简单数据类型，是复合数据类型 (类类型)。

一个字符串常量是用双引号括起来的 0 个或多个字符组成的序列。例如：

```
"" //空串
```

```
"Hello World\n" //一行字符，最后一个是回车换行符
```

两个字符串可以用连接符“+”连接，此时允许分行写。例如：

```
" This is a string"+
```

```
" That is another string" 等价于" This is a string That is an other string "
```

1.1.7 变量

变量是 Java 程序中的基本存储单元，它具有名称、类型、值和作用域等特性。在使用任何变量之前必须先定义。如果是基本数据类型的变量，在声明变量类型时，就为其分配了内存单元，因为它们的存储长度是固定的。如果是复合数据类型的变量，声明完其数据类型之后，还要用关键字 `new` 为其分配内存单元。

1. 变量定义

每个变量必须定义为某一种数据类型，而且只能声明为惟一的数据类型，不允许重复定义。变量定义是用标识符为变量起名、确定其数据类型，还可以为它赋初值（称作变量初始化）。变量必须是先定义后使用。

变量的定义格式为：

```
数据类型 标识符 1[ = 初值], [标识符 2……];
```

标识符用于为变量命名；赋初值使变量一开始有确定的值，为可选项；可用同一类型定义多个变量，各变量之间用逗号作为分隔符。

2. 变量赋值与类型转换

在进行赋值之前，首先检查赋值运算符左右两端数据的类型是否一致。当出现类型不匹配的赋值操作时，表示数的范围较小的数据类型可自动向表示数的范围较大的数据类型转换；否则，必须强迫类型转换，不然将导致编译错误。

所有整型数据类型都能向 `float` 或 `double` 自动转换，反之必须强制转换。

字符类型可向 `int`、`long`、`float` 或 `double` 自动转换，反之必须强制转换。

字符类型转换为字节类型必须强制转换。

布尔型不能与任何其他数据类型转换。

例如: `byte a=(byte)268;`

`int b=(int)998877665544332L;`

`float c=998877665544332L。`

1.2 典型例题分析

例 1.1 本例是一个应用程序, 定义了 `byte`、`int` 和 `char` 等数据类型并显示。

程序中 `System.out.print()` 输出结果后光标停留在本行; `System.out.println()` 输出结果后光标在下一行, 可用 `println()` 方法输出空行。

程序如下:

```
class A
{
    public static void main(String[] args)
    {
        //定义 byte 类型
        byte a1=126,a2=(byte)256,a3=(byte)'0';
        System.out.println("a1="+a1+"\ta2="+a2+"\ta3="+a3);
        //定义 int 类型
        int b1=12345,b2=(int)987654321000L,b3='A',b4=0xff;
        System.out.print("b1="+b1+"\tb2="+b2);
        System.out.print("\tb3="+b3+"\tb4="+b4);
        System.out.println();
        //定义字符型
        char c1='a',c2=98,c3='\u0042',c4='\n';
        System.out.println("c1="+c1+"\tc2="+c2+c4+"c3="+c3);
    }
}
```

程序的输出结果如图 1-2 所示。

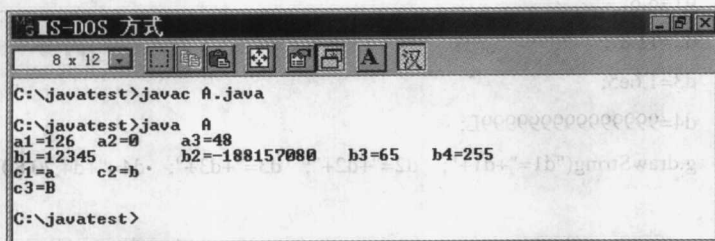


图 1-2 例 1.1 程序输出的结果

例 1-2: 本例是一个小程序, 定义了 `String`、`boolean`、`float` 和 `double` 类型的数据并显示。

小程序窗口中显示结果的方法是 `drawString` (“字符串”, x 坐标, y 坐标); 该方法的第一个参数必须是字符串类型, 第 2、3 个参数是小程序窗口的 x 坐标和 y 坐标。小程序窗口坐标系的原点在窗口的左上角, 水平向右是 x 的正方向, 垂直向下是 y 的正方向。

程序中定义了: `String s="java 程序设计";`, 因为 `s` 是 `String` 类的一个实例, 所以语句: `b3=s instanceof String;`返回的结果为 `true`。

程序如下:

```
import java.awt.*;
import java.applet.*;
public class Applet1 extends Applet
{
    public void paint(Graphics g)
    {
        //定义一个字符串
        String s="java 程序设计";
        g.drawString(s,20,20);
        //定义布尔型变量
        boolean b1,b2,b3;
        b1=true;
        b2=5>6;
        b3=s instanceof String;
        g.drawString("b1="+b1+"; b2="+b2+"; b3="+b3,20,40);
        //定义单精度小数
        float c1,c2,c3,c4;
        c1=5.8f;
        c2=(float)7.8;
        c3=1.6e-6f;
        c4='a';
        g.drawString("c1="+c1+"; c2="+c2+"; c3="+c3+"; c4="+c4,20,60);
        //定义双精度小数
        double d1,d2,d3,d4;
        d1=8.8;
        d2=11.8f;
        d3=1.6e5;
        d4=9999999999999999L;
        g.drawString("d1="+d1+"; d2="+d2+"; d3="+d3+"; d4="+d4,20,80);
    }
}
```

编译和运行小程序如图 1-3 所示。

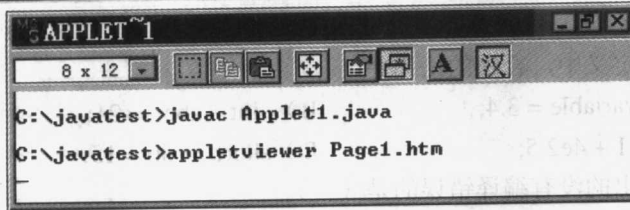


图 1-3 编译和运行小程序

程序的输出结果如图 1-4 所示。

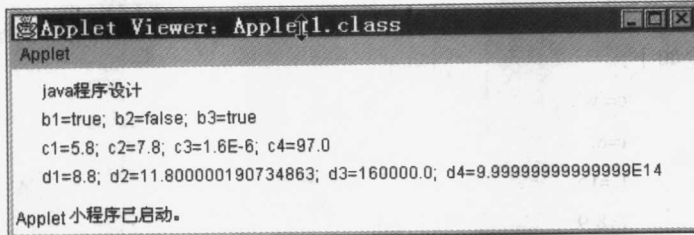


图 1-4 例 1.2 程序输出的结果

1.3 习 题

一、选择题（下面的选择题中选项至少有一个是正确的，请将正确的选项填写到题后的括号中）

- 下面说法正确的是（ ）。
 - Java 程序的源文件名称与主类（公共类）的名称相同，后缀可以是.java 或.txt 等。
 - JDK 的编译命令是 java。
 - 一个 Java 源程序编译后可能产生几个字节码文件。
 - 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。
- 下面的说法正确的是（ ）。
 - Java 语言是面向对象的、解释执行的网络编程语言。
 - Java 语言具有可移植性，是与平台无关的编程语言。
 - Java 语言可对内存垃圾自动收集。
 - Java 语言编写的程序虽然是“一次编译，到处运行”，但必须要 Java 的运行环境。
- 下面（ ）不是 Java 的关键字。

A) integer	B) double	C) float	D) char
------------	-----------	----------	---------
- 以下语句中没有编译错误或警告提示信息的是（ ）。

A) byte b=256;	B) double d=89L;	C) char c="a";	D) short s=8.6f;
----------------	------------------	----------------	------------------
- 以下的选项中能正确表示 Java 语言中的一个整型常量的是（ ）。

A) 35.d	B) -20	C) 1,234	D) "123"
---------	--------	----------	----------
- 下面的数据类型（ ）是 float 型。