

## 第1章 Java入门

### 1. 开发与运行Java程序需要经过哪些主要步骤和过程？

答：（1）编写Java源文件:使用文本编辑器（Edit或记事本），拓展名为. java  
（2）编译Java源文件:使用Java编译器（javac.exe）。得到字节码文件\*. class  
（3）运行Java程序:Java应用程序使用Java解释器（java.exe）执行字节码文件；  
Java小应用程序使用支持Java标准的浏览器来执行。

### 2. 怎样区分应用程序和小应用程序？应用程序的主类或小应用程序的主类必须用public修饰吗？

答：①应用程序必须有main方法，这个方法是程序执行的入口。

小应用程序没有main方法。

②应用程序的主类不一定用public修饰；小应用程序的主类必须用public修饰。

### 3. Java程序是由什么组成的？一个程序中必须要有public类吗？Java源文件的命名规则是怎样的？

答：①Java程序由类组成。

②应用程序可以没有public类；小应用程序一定有一个类是public类（主类）。

③应用程序:如果只有一个类，源文件名与该类的类名相同，拓展名为. java；

有多个类时，如果有public类（最多一个），源文件名与public类的类名相同，拓展名是. java；没有public类，源文件名与任何一个类的类名相同即可，拓展名为. java。

小应用程序:源文件名与主类的类名相同，拓展名是. java。

4. 在运行小程序的HTML文件中可以使用codebase属性指定小程序的字节码所驻留的目录。如果不使用codebase属性，小程序的字节码文件必须和运行它的HTML文件在同一目录中。编写一个小程序并将小程序的字节码存放在某个目录中，比如C:\5000；把运行该小程序的HTML文件（注意其中的codebase属性）：

```
<applet code=你的小程序的字节码 width=200 height=300 codebase=C:\5000>
</applet>
```

存放在另一个目录中。

答：以书中小应用程序为例进行说明：

#### ①编写Java小应用程序源代码

```
import java.applet.*;
import java.awt.*;
public class Boy extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.red);
        g.drawString("我一边喝着咖啡，一边学Java呢", 5, 30);
        g.setColor(Color.blue);
        g.drawString("我学得很认真", 10, 50);
    }
}
```

②将上述代码命名为Boy. java并进行编译得到Boy. class字节码文件；将得到的字节码文件存放在

C:\5000文件夹下；

#### ③编写小应用程序的HTML源文件：

```
<applet code=Boy.class width=200 height=300 codebase=C:\5000> </applet>
```

④将上述编写好的代码命名为Boy. html；并将此文件存放在C:\5000文件夹以外的任意一个文件夹中（如C:\1000）；

⑤运行Boy. html。

## 第2章 标识符、关键字和数据类型

### 1. 什么叫标识符？标识符的规则是什么？

答：①用来标识类名、变量名、方法名、类型名、数组名、文件名的有效字符序列。

②由字母、下划线、美元符号和数字组成，并且第一个字符不能是数字字符，而且关键字不能作为标识符。

### 2. 什么叫关键字？请说出5个关键字。

答：①Java语言中已经被赋予特定意义的一些单词。

②class break public return static extends等。

### 3. Java的基本数据类型都是什么？

答：boolean(逻辑型) char(字符型) float、double(浮点型) byte、short、int、long(整型)。

4. 下列哪些语句是错误的:

```
int x = 8;
byte b = 127;
b = x;
```

答:b=x语句错误;原因是高级别的变量赋值给低级别的变量时,一定要用显式转换即b=(byte)x;。

5. 下列程序的输出结果是什么?

```
public class E
{
    public static void main(String args[])
    {
        long[] a={1, 2, 3, 4};
        long[] b={100, 200, 300, 400, 500};
        b=a;
        System.out.println("数组b的长度:"+b.length);
        System.out.println("b[0]="+b[0]);
    }
}
```

答: 数组b的长度:4

b[0]=1

6. 上机运行下列程序, 注意观察输出结果。

```
public class E
{
    public static void main(String args[])
    {
        for(int i=20302;i<20322;i++)
        {
            System.out.println((char)i);
        }
    }
}
```

答: 低住佐佑侠体佔何佻佻余余佚佛作佻佻佟你佻

7. System.out.println(“你好”); 可输出字符串, 也可以使用System.out.println( )输出变量或表达式的值, 只需使用并置符号“+”将变量、表达式或一个常数值与一个字符串并置即可, 如:

System.out.println(“ ”+x); System.out.println(“:”+123+“大于”+122); 等。上机调试下列程序, 注意观察结果, 特别注意System.out.print( )和System.out.println( )的区别。

```
public class OutputData
{
    public static void main(String args[])
    {
        int x=234, y=432;
        System.out.println(x+"<"+(2*x));
        System.out.print("我输出结果后不回车");
        System.out.println("我输出结果后自动回车到下一行");
        System.out.println("x+y= "+(x+y));
        System.out.println(" "+x+y+"=234432");
    }
}
```

答: 234<468

我输出结果后不回车我输出结果后自动回车到下一行

x+y= 666

234432=234432

8. 编写一个Java应用程序, 输出全部的希腊字母。

答:

```
public class a
{
    public static void main(String args[])
    {System.out.println("这是我的第一个程序");}
}
```

## 第3章 运算符、表达式和语句

1. 下列程序的输出结果是什么？

```
public class E
{
    public static void main(String args[])
    {
        char x='你',y='e',z='吃';
        if(x>'A')
        {
            y='爱';
            z='情';
        }
        else
        {
            y='我';
            z='她';
        }
        System.out.println(" "+x+y+z);
    }
}
```

答：你爱她

2. 下列程序的输出结果是什么？

```
public class E3
{
    public static void main(String args[])
    {
        char c='\0';
        for(int i=1;i<=4;i++)
        {
            switch(i)
            {
                case 1:c='b';
                System.out.print(c);
                case 2:c='e';
                System.out.print(c);
                break;
                case 3:c='p';
                System.out.print(c);
                default:
                System.out.print("!");
            }
        }
    }
}
```

答：beep! !

3. 编写应用程序，求1! +2! +……+10! 。

答：

```
class Fact
{
    public static void main(String args[])
    {
        int fact,sum=0;
        for(int i=1;i<=10;i++)
        {
            fact=1;
            for(int j=1;j<=i;j++)
            fact*=j;
            sum+=fact;
        }
        System.out.println("1到10的阶乘之和是:"+sum);
    }
}
```

```
}
```

4. 编写一个应用程序，求100以内的全部素数。

答：

```
class Primes
{
public static void main(String args[])
{
int w=1;
for(int i=2;i<=100;i++)
{
for(int j=2;j<i;j++)
{
w=i%j;
if(w==0)
break;
}
if(w!=0)
System.out.println(i+"是素数");
}
}
}
```

7. 编写应用程序，分别使用while和for循环计算 $8+88+888+\dots$ 前10项之和。

答：①for循环

```
class TheSum
{
public static void main(String args[])
{
long sum=0,data=8;
for(int i=1;i<=10;i++)
{
sum=sum+data;
data=data*10+8;
}
System.out.println(sum);
}
}
```

②while循环

```
class TheSum
{
public static void main(String args[])
{
long sum=0,data=8,i=0;
while(++i<=10)
{
sum=sum+data;
data=data*10+8;
}
System.out.println(sum);
}
}
```

8. 编写应用程序，输出满足 $1+2+3+\dots+n<8888$ 的最大正整数n。

答：

```
class Maxn
{
public static void main(String args[])
{
int k=1,sum=0;
while((sum+=k)<8888)
k++;
}
```

```

k--;
System.out.println("最大能取到:"+k);
}
}

```

## 第4章 类、对象和接口

1. 类中的实例变量在什么时候会被分配内存空间？

答：当该类创建对象之后，实例变量才会被分配相应的内存空间。

2. 什么叫方法的重载？构造方法可以重载吗？

答：①一个类中可以有多个方法具有相同的名字，但这些方法的参数必须不同，即或者是参数的个数不同，或者是参数的类型不同（方法的重载体现了多态性，即功能多态性）。

②构造方法可以重载。

3. 类中的实例方法可以操作类变量(static变量)吗？类方法(static方法)可以操作实例变量吗？

答：①类中的实例方法可以操作类变量。

②类方法不可以操作实例变量

4. 类中的实例方法可以用类名直接调用吗？

答：类中的实例方法不可以用类名直接调用。

5. 举例说明protected方法和友好方法的区别。

答：当子类与父类不在同一个包中时，子类可以继承父类的protected方法；而友好方法此时则不能被子类继承。

6. 举例说明类变量和实例变量的区别。

答：(1)书写：定义成员变量时，类变量有static修饰；实例变量没有static修饰。

例：class A

```

{
int x; //实例变量
static int y; //类变量
}

```

(2)内存：不创建对象，类的实例变量不会被分配内存空间；类变量被分配相应的内存空间。

不同对象的实例变量被分配不同的内存空间；不同对象的类变量被分配相同的内存空间。

任何一个对象改变类变量，其他对象的相应类变量也发生相应变化。

一个对象的实例变量发生改变，不影响其他对象的相应实例变量。

例：执行语句：A1.x=10; A1.y=20;

这时A2.x的值也是10；而A2.y的值保持原来的初值。

(3)使用：类变量可以被类方法操作；而实例变量不能被类方法操作。

例：class A

```

{
int x;
static int y;
static void func()
{
b=10; //合法
a=20; //非法
}
}

```

类变量可以通过类名访问；实例变量不能通过类名访问。

例：class A

```

{
int x;
static int y;
}
class B
{
public void func();
{
A.x=10; //非法
A.y=20; //合法
}
}

```

7. 子类将继承父类的哪些成员变量和方法？子类在什么情况下隐藏父类的成员变量和方法？在子类中是否允许有一个方法和父类的方法名字相同，而类型不同？

答:①子类和父类在同一个包中时,子类可以继承父类的除private属性的所有方法和成员变量,当子类与父类不在同一个包中时,子类只能继承父类的protected和public属性的成员变量和方法。

②子类中定义的成员变量和父类中的成员变量同名时,子类就隐藏了继承的成员变量;子类中定义一个方法,并且这个方法的名字、返回类型、参数个数和类型与从父类继承的方法完全相同,子类就隐藏从父类继承来的方法(方法的重写)。

③在子类中允许有一个方法和父类的方法名字相同而类型不同。

8. 父类的final方法可以被子类重写吗?

答:父类的final方法不能被子类重写。

9. 什么类中可以有abstract方法?

答:abstract类中可以有abstract方法。

10. 什么叫对象的上转型对象?

答:B类是A类的子类或间接子类,当用子类B创建一个对象b并把这个对象的引用放到A类的对象a中时,称A类对象a是子类对象b的上转型对象。

11. 什么叫接口的回调?

答:可以把实现某一接口的类创建的对象引用赋给该接口声明的接口变量中。

12. 下列程序有什么错误?

```
public class Takecare
{
    int a=90;
    static float b=10.98f;
    public static void main(String args[])
    {
        float c=a+b;
        System.out.println("c="+c);
    }
}
```

答:语句float c=a+b;错误。原因是类方法main操作了实例变量a。

18. 编写一个类,该类有一个方法public int f(int a,int b),该方法返回a和b的最大公约数。然后再编写一个该类的子类,要求子类重写方法f,而且重写的方法将返回a和b的最小公倍数。要求在重写的方法的方法体中首先调用被隐藏的方法返回a和b的最大公约数m,然后将乘积(a\*b)/m返回。要求在应用程序的主类中分别使用父类和子类创建对象,并分别调用方法f计算两个正整数的最大公约数和最小公倍数。

答: class A

```
{
public int f(int a,int b)
{
    if(a<b)
    {
        int temp=0;
        temp=a;
        a=b;
        b=temp;
    }
    int r=a%b;
    while(r!=0)
    {
        a=b;
        b=r;
        r=a%b;
    }
    return b;
}
}
class B extends A
{
    public int f(int a,int b)
    {
        int m;
        m=super.f(a,b);
        return (a*b)/m;
    }
}
```

```

}
}
public class Test
{
public static void main(String args[])
{
A a=new A();
System.out.println("18和102的最大公约数是:"+a.f(18,102));
B b=new B();
System.out.println("18和102的最小公倍数是:"+b.f(18,102));
}
}

```

## 第5章 字符串

1. 使用String类的public String toUpperCase()方法可以将一个字符串中的小写字母变成大写字母，使用public String toLowerCase()方法可以将一个字符串中的大写字母变成小写字母。编写一个程序，使用这两个方法实现大小写的转换。

答: class Test

```

{
public static void main(String args[])
{
String str="I can use Java";
System.out.println("要转换的字符串是:"+str);
String s=str.toUpperCase();
System.out.println("转换成大写字符串是:"+s);
s=str.toLowerCase();
System.out.println("转换成小写字符串是:"+s);
}
}

```

2. 使用String类的public String concat(String str)方法可以把调用该方法的字符串与参数指定的字符串连接，把str指定的串连接到当前串的尾部获得一个新的串。编写一个程序通过连接两个串得到一个新串，并输出这个新串。

答: class Test

```

{
public static void main(String args[])
{
String str1="I can u";
String str2="se Java";
String s=str1.concat(str2);
System.out.println("将字符串"+str1+"与字符串"+str2+"连接后得到的新字符串是:");
System.out.println(s);
}
}

```

3. String类的public char charAt(int index)方法可以得到当前字符串index位置上的一个字符。说出下列程序的输出结果。

```

public class E3
{
public static void main(String args[])
{
String s="中国科学技术大学";
char a=s.charAt(2),b=s.charAt(6);
System.out.print(a);
System.out.println(b);
}
}

```

答: 科大

src - 被复制的数组

srcPos - 从第几个元素开始复制

dest - 要复制到的数组

destPos - 从第几个元素开始粘贴

length - 一共需要复制的元素个数

## 第6章 时间、日期和数字

}

5. 使用BigInteger类计算1! +3! +5! +7! +...的前30项的和。

```
答: import java.math.*;
public class Test
{
    public static void main(String args[])
    {
        BigInteger sum=new BigInteger("0"),
        jieCheng=new BigInteger("1"),
        ONE=new BigInteger("1"),
        i=ONE;
        int k=0;
        while(++k<=30)
        {
            sum=sum.add(jieCheng);
            i=i.add(ONE);
            jieCheng=jieCheng.multiply(i);
            i=i.add(ONE);
            jieCheng=jieCheng.multiply(i);
        }
        System.out.println(sum);
    }
}
```

## 第7章 AWT组件及事件处理

1. Frame类对象的默认布局是什么布局? 和Panel类对象的默认布局相同吗?

答: ①Frame类对象的默认布局是BorderLayout布局;

②不相同, Panel类对象的默认布局是FlowLayout布局。

2. 一个容器对象是否可以使用add方法添加一个Frame窗口?

答: 不可以。

3. Checkbox对象可以触发ItemEvent事件吗?

答: 可以。

11. 编写程序, 观察各种组件设置背景色和前景色的情况。

```
答: import java.awt.*;
import java.awt.event.*;
class WindowColor extends Frame implements ActionListener
{
    Button button; //按钮
    TextField textfield; //文本框
    TextArea textarea; //文本区
    Mypanel panel; //面板
    Checkbox box; //选择框
    Choice choice; //下拉列表
    List list; //滚动列表
    Label label; //标签
    Mycanvas can; //画布
    Button buttonBackColor, buttonForeColor;
    WindowColor()
    {
        button=new Button("我是按钮");
        textfield=new TextField("我是文本框", 10);
        textarea=new TextArea(6, 15);
        textarea.setText("我是文本区");
        textfield.setEditable(false);
        textarea.setEditable(false);
        panel=new Mypanel();
        box=new Checkbox("我是选择框");
```



```

choice=new Choice();
choice.add("我是下拉列表");
list=new List(3, false);
list.add("我是滚动列表");
label=new Label("我是标签");
can=new Mycanvas();
buttonBackColor=new Button("背景色");
buttonForeColor=new Button("前景色");
setLayout(new FlowLayout());
add(button);
add(textfield);
add(textarea);
add(panel);
add(box);
add(choice);
add(list);
add(label);
add(can);
add(buttonBackColor);
add(buttonForeColor);
addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
}
);
buttonBackColor.addActionListener(this);
buttonForeColor.addActionListener(this);
setBounds(100, 100, 300, 300);
setVisible(true);
validate();
}
public void actionPerformed(ActionEvent e)
{
if (e.getSource()==buttonBackColor)
{
button.setBackground(Color. yellow);
textfield.setBackground(Color. yellow);
textarea.setBackground(Color. yellow);
panel.setBackground(Color. yellow);
box.setBackground(Color. yellow);
choice.setBackground(Color. yellow);
list.setBackground(Color. yellow);
label.setBackground(Color. yellow);
can.setBackground(Color. yellow);
}
else if (e.getSource()==buttonForeColor)
{
button.setForeground(Color. blue);
textfield.setForeground(Color. blue);
textarea.setForeground(Color. blue);
panel.setForeground(Color. blue);
box.setForeground(Color. blue);
choice.setForeground(Color. blue);
list.setForeground(Color. blue);
label.setForeground(Color. blue);
can.setForeground(Color. blue);
}
}
}

```

```
}  
}  
}  
class Mycanvas extends Canvas  
{  
    Mycanvas() { }  
    public void paint(Graphics g)  
    {  
        g.drawString("我是画布", 5, 5);  
    }  
}  
class Mypanel extends Panel  
{  
    Button button1;  
    Mypanel()  
    {  
        button1=new Button("我是面板");  
        add(button1);  
    }  
}  
public class Test  
{  
    public static void main(String args[])  
    { new WindowColor();  
    }  
}
```