

## 《Java 程序设计基础》 复习思考题

### 1. 位运算符

1) 编写程序，任意输入 2 个整数，计算出它们的和、差、积和商。

[参考代码段]

```
:
    int a,b;
    Scanner x=new Scanner(System.in);
    a=x.nextInt();
    b=x.nextInt();
    System.out.println(a+" "+b+"="+ (a+b));
    System.out.println(a+" - "+b+"="+ (a-b));
    System.out.println(a+" * "+b+"="+ (a*b));
    System.out.println(a+" / "+b+"="+ (a/b));
:
```

2) 编写程序利用左移运算，输出该整数的二进制表示。

[参考代码段]

```
:
int a=15;
int b=128;
System.out.print((a&b)==128?1:0);
a=a<<1;
System.out.print((a&b)==128?1:0);
a=a<<1;
:
```

### 2. 分支

1) 编写程序，输入某年、某月，判断该月的天数。

[参考代码段]

```
:
switch (month) {
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        numDays = 31;
        break;
    case 4:
```

```

        case 6:
        case 9:
        case 11:
            numDays = 30;
            break;
        case 2:
            if ((year %4 == 0)&&!(year%100 == 0))||(year%400 == 0))
                numDays = 29;
            else
                numDays = 28;
            break;
    } //switch结束
    System.out.println("number of Days = " + numDays);
:

```

2) 编写程序，判断 1980-2015 年中哪些年份是闰年。

[参考代码段]

```

:
    for(int i = 1980;i<=2015;i++)
    {
        if(((i%4==0)&&!(i%100==0))||(i%400==0))
            System.out.println(i+"是闰年");
    }
:

```

3. 编写程序，任意输入 10 个数，排序后输出。

[参考代码段]

```

:
//假设10个整数已经放入数组arrInt中
    for(i=0;i<arrInt.length-1;i++)
        for(j=0;j<arrInt.length-1-i;j++){
            if(arrInt[j]>arrInt[j+1]){
                temp=arrInt[j];
                arrInt[j]=arrInt[j+1];
                arrInt[j+1]=temp;
            }
        }
:

```

4. 输入 16 个整数，读入一个 4×4 阵列中，并输出该阵列两条对角线上的元素的累加和。

5. 构建 Calculator 类，包含 add(), minus(), multiply(), divide() 四个方法，计算两个整数的加、减、乘、除。

```

class Calculator{
    Calculator(){}
    float add(float a, float b){
        return a+b;
    }
    float minus(float a, float b){
        return a-b;
    }
    float multiply (float a, float b){
        return a*b;
    }
    float divide (float a, float b){
        return a/b;
    }
}

```

#### 6. 静态方法

- 1) 编写程序，模拟掷两个骰子的数字。
- 2) 模拟掷一个骰子的数字。

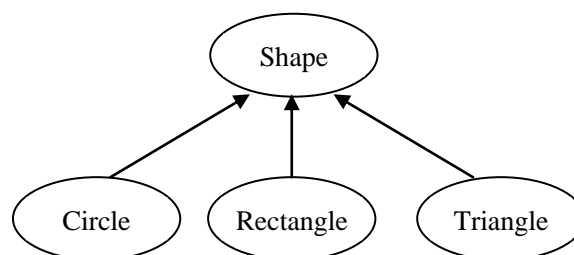
[参考代码段]

```

:
    for (int i=0; i<10; i++){
        System.out.print((int) (Math.random()*1000)%6+1);
        System.out.print(" "); //留出一个空格
    }
:

```

7. 计算 Shape 的面积需要 Circle, Rectangle, Triangle 的 area() 方法来具体实现，它们的继承/实现关系如下：



- (1) 把 Shape 看着抽象类，编写 Rectangle 类。
- (2) 把 Shape 看着接口，编写 Triangle 类。

8. 在 java.util.Vector 基础上封装一个类 VectorStack 实现堆栈，VectorStack 有压入方法 push() 和弹出方法 pop()，见下面程序：

```

import java.util.Vector;
class VectorStack{
    static final int CAPACITY=5;

```

```

Vector v;
VectorStack(){
    v = new Vector();
}
void push(Object obj){
    v.addElement(obj);
    System.out.print(" PUSH: "+obj);
}
Object pop(){
    Object obj = v.lastElement();
    v.removeElementAt(v.size()-1);
    System.out.println(" Pop: "+obj);
    return obj;
}
}

```

改进该 pop()方法，使得在弹出数据时，如果栈为空，抛出一个异常。并编写测试程序进行验证。

[参考答案]

1) 自定义异常

```

public class MyException extends Exception{
    public MyException(String msg) {
        super(msg);
    }
}

```

2) 改进后的类

```

import java.util.Vector;
public class VectorStack{
    static final int CAPACITY=5;
    Vector v;
    VectorStack(){
        v = new Vector();
    }
    void push(Object obj){
        v.addElement(obj);
        System.out.print(" PUSH: "+obj);
    }
    Object pop() throws MyException{
        if(v.size()==0) throw new MyException("栈无元素");

        Object obj = v.lastElement();
        v.removeElementAt(v.size()-1);
        System.out.println(" Pop: "+obj);
        return obj;
    }
}

```

```

    }
}

```

### 3) 测试程序

```

public class Test {
    public static void main(String[] args) {
        VectorStack vs = new VectorStack();
        try {
            vs.pop();
        } catch (MyException e) {
            System.out.println("发现异常: "+e.getMessage());
        }
    }
}

```

### 9. 在图形界面的窗口中，读入姓名，并初始化一个 Human 对象。

[参考代码段]

```

class InstanceOfHuman extends Frame implements ActionListener{
    Label namelabel = new Label("姓名");
    TextField nametextfield = new TextField(10);
    Button okbutton = new Button("确定");
    Label out = new Label("          ");
    public InstanceOfHuman() {
        setLayout(new FlowLayout());
        add(namelabel);
        add(nametextfield);
        add(okbutton);
        add(out);
        okbutton.addActionListener(this);
        setSize(400,100);
        show();
    }
    public void actionPerformed(ActionEvent a){
        Human p = new Human(nametextfield.getText());
        out.setText("I am " + p.name);
    }
}

```

### 10. 编写一个 Applet 程序，输出自己的姓名。写出该程序的源代码和包含<APPLET> 标签的网页。

[参考代码段]

```

public class HelloWorld extends Applet //继承Applet类
{
    public void paint(Graphics g )

```

```

{
    g.drawString("Hello World!", 5, 25);
}
}

```

11. 编写程序，显示当前目录下的文件和目录。

[提示] 需要用到File 的几个重要的方法：

public boolean exists() 若文件或目录存在，则返回true；否则返回false；

public boolean isFile() 若对象代表有效文件，则返回true；

public boolean isDirectory() 若对象代表有效目录，则返回true；

public String getAbsolutePath() 获取当前绝对路径；

public String[] list() 列出目录中的所有文件(包括有效文件和路径)，将目录中的所有文件名保存在字符串数组中返回；

[参考答案]

```

public class ShowAll {
    public static void main(String[] args) {
        String path="d:/";
        File file=new File(path);
        File[] tempList = file.listFiles();
        System.out.println("该目录下对象个数: "+tempList.length);
        for (int i = 0; i < tempList.length; i++) {
            if (tempList[i].isFile()) {
                System.out.println("文    件: "+tempList[i]);
            }
            if (tempList[i].isDirectory()) {
                System.out.println("文件夹: "+tempList[i]);
            }
        }
    }
}

```