

## 作业题

### 一、选择题

1、以下代码的运行结果是？

```
public class Test {  
    public static void main(String[] args) {  
        int a = 7;  
        int b = 3;  
        System.out.println(a / b);  
    }  
}
```

A. 2.333 **B. 2** C. 3 D. 1

2、执行下列代码后，输出结果为？

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println("5" + 2 + 3);  
    }  
}
```

A. 55 B. 10 **C. 523** D. 5+2+3

3、下列关于关系运算符的说法中，正确的是？

- A. "==" 是赋值运算符
- B. "=" 用于判断两个值是否相等
- C. "!=" 表示不等于**
- D. 关系运算符的运算结果是整数

4、下列哪种情况最适合使用 if 语句格式 3 (if-else if-else) ？

- A. 判断一个数是否为正数
- B. 判断一个数是正数、负数还是零**
- C. 判断开关是否打开
- D. 判断一个整数是否能被 2 整除

5、关于 switch 语句，下列说法错误的是？

- A. switch 后面的表达式可以是 int 类型
- B. case 后面的值必须是常量
- C. 每个 case 语句后必须加 break，否则会报错**
- D. default 语句可以放在 switch 语句的任意位置

### 二、问答题

1、请简述算术运算符的两个注意事项。

1、注意控制运算顺序 2、不同运算符之间的计算优先级。

2、请描述 switch 语句的执行流程及其中 break 语句的作用。

对判断表达式进行判断后，依次对case 后的常量进行匹配，匹配成功就运行该case中的代码，反之则不运行，break的用处是跳出选择结构 不再进行选择结构语句中后续的代码判断和执行。

### 三、代码编程题

1、已知一个四位数 4567，编写程序将其拆分为个位、十位、百位、千位后，输出在控制台。

要求：

```
package com.zuoye;

public class test {
    public static void main(String[] args) {
        int x = 4567;
        System.out.println("个位是: "+x%10);
        System.out.println("十位是: "+x/10%10);
        System.out.println("百位是: "+x/100%10);
        System.out.println("千位是: "+x/1000%10);
    }
}
```

- 分析思路清晰
- 编写完整的Java 代码实现

2、定义一个整数表示星期几（1 表示星期一，2 表示星期二，...，7 表示星期日），使用 if 语句判断该星期几是工作日还是休息日，并在控制台输出结果（周一至周五为工作日，周六和周日为休息日）。

要求：

```
package com.zuoye;

import java.util.Scanner;

public class test {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        while(s.hasNextInt()){
            System.out.println("请问今天是工作日还是休息日? ");
            int x = s.nextInt();
            if(x>=1&&x<=5){
                System.out.println("今天是工作日!!!!");
            }else if(x>5&&x<=7){
                System.out.println("今天是休息日!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
            }else{
                System.out.println("错误的输入!");
            }
        }
    }
}
```

- 分析思路清晰
- 编写完整的 Java 代码实现