

基础题目

第一题：语法练习

- 语法点：方法定义，方法重载
- 按步骤编写代码，效果如图所示：

```
iVar:10, iVar2:10
翻倍:r=20,p=20
iVar:10, iVar2:10
*****
iVar:20, iVar2:20
```

- 编写步骤：
 1. 定义类 Test1,类中定义 main方法,定义int类型a为10, b为10.
 2. 定义printNum方法,参数为(int iVar, int iVar2),返回值无,输出iVar和iVar2的值
 3. 定义doubling方法,参数为(int r, int p),返回值无,方法内r翻倍,p翻倍,并调用printNum方法,输出r和p的值
 4. 定义doubling方法,参数为(int r),返回值int, 方法内r翻倍,返回r.
 5. main方法中,调用printNum方法,传入a,b
 6. main方法中,调用doubling方法,传入a,b
 7. main方法中,调用printNum方法,传入a,b
 8. main方法中,调用doubling方法,传入a,用a接收返回值
 9. main方法中,调用doubling方法,传入b,用b接收返回值
 10. main方法中,调用printNum方法,传入a,b
- 参考答案：

```
public class Test1 {
    public static void main(String[] args) {
        // 定义int类型a为10, b为10.
        int a = 10;
        int b = 10;
        printNum(a, b);
        doubling(a, b);
        printNum(a, b);

        System.out.println("*****");

        a = doubling(a);
        b = doubling(b);
    }
}
```

```

        printNum(a, b);
    }

    // 定义printNum方法,参数为(int iVar, int iVar2),返回值无,输出iVar和iVar2的值
    public static void printNum(int iVar, int iVar2) {
        System.out.println("iVar:" + iVar + ", iVar2:" + iVar2);
    }

    // 定义doubling方法,参数为(int r, int p),返回值无,方法内r翻倍,p翻倍,并调用
    printNum方法,输出r和p的值
    public static void doubling(int r, int p) {
        r *= 2;
        p *= 2;
        System.out.println("翻倍:r=" + r + ",p=" + p);
    }

    // 定义doubling方法,参数为(int r),返回值int, 方法内r翻倍,返回r.
    public static int doubling(int r) {
        r *= 2;
        return r;
    }
}

```

第二题：语法练习

- 语法点：方法定义，if
- 按步骤编写代码，效果如图所示：

```

num: -208
true
num: 213
false

```

- 编写步骤：
 1. 定义类 Test2,类中定义 main方法
 2. 定义doCheck方法,参数为(int iVar),返回值boolean类型
 3. doCheck方法内,定义变量boolean flag.
 4. doCheck方法内,判断num是否为偶数.
 5. 如果是偶数,使用for循环,初始化值i为0,i<=20进入循环,步进表达式i++
 6. 循环内,num-=i;
 7. flag赋值为true.
 8. 否则是奇数,使用for循环,初始化值i为0,i<=20进入循环,步进表达式i++
 9. 循环内,num+=i;
 10. flag赋值为false.

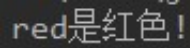
11. 输出num的值
12. 返回flag
13. 调用doCheck方法,传入2,保存返回值,并输出
14. 调用doCheck方法,传入3,保存返回值,并输出

■ 参考答案:

```
public class Test2 {  
    public static void main(String[] args) {  
        boolean b = doCheck(2);  
        System.out.println(b);  
  
        boolean b1 = doCheck(3);  
        System.out.println(b1);  
    }  
  
    // 定义doCheck方法,参数为(int iVar),返回值boolean类型  
    public static boolean doCheck(int num){  
        // 3.doCheck方法内,定义变量boolean flag.  
        boolean flag ;  
  
        // 4.doCheck方法内,判断num是否为偶数.  
        if (num % 2== 0 ) {  
            // 如果是偶数,使用for循环,初始化值i为0,i<=20进入循环,步进表达式i++  
            for (int i = 0; i <= 20; i++) {  
                num-=i;  
            }  
            flag = true;  
        }else {  
            // 否则是奇数,使用for循环,初始化值i为0,i<=20进入循环,步进表达式i++  
            for (int i = 0; i <= 20; i++) {  
                num+=i;  
            }  
            flag = false;  
        }  
        // 输出num的值  
        System.out.println("num:"+ num);  
        return flag;  
    }  
}
```

第四题：需求实现

- 定义showColor方法，根据英文单词，输出对应的颜色。
- 代码实现，效果如图所示：

A terminal window showing the output of the program: "red是红色!". The text is white on a dark background.

- 参考答案：

```
public class Test4 {  
    public static void main(String[] args) {  
        showColor("red");  
    }  
    // showColor方法中,使用switch语句,判断颜色.  
    public static void showColor(String color) {  
        switch (color) {  
            case "red":  
  
                System.out.println(color + "是红色!");  
                break;  
            case "blue":  
  
                System.out.println(color + "是蓝色!");  
                break;  
            case "green":  
  
                System.out.println(color + "是绿色!");  
                break;  
            default:  
                System.out.println(color+" 颜色未知!");  
        }  
    }  
}
```

第五题：需求实现

- 定义getValue方法，获取三个数中的最值，能够通过字符串，指定获取最大值或者最小值。
- 代码实现，效果如图所示：

A terminal window showing the output of the program: "最小值为:5". The text is white on a dark background.

- 开发提示：
 - getValue中，带有String类型参数，可以指定『大』或者『小』
 - 分别定义最大值方法，最小值方法，以供getValue调用。
- 参考答案：

```
public class Test5 {
    public static void main(String[] args) {

        getExtValue("小" , 5, 6, 7);

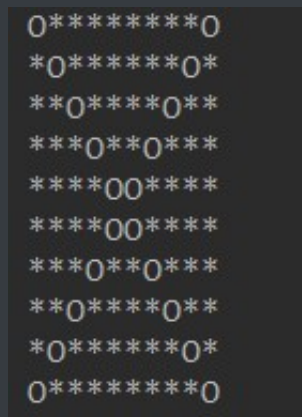
    }
    // 定义getExtValue方法,参数为(String str, int n, int n2, int n3),返回值无
    public static void getExtValue(String ext , int n1,int n2 , int n3) {
        switch (ext) {
            // // 当str为大时,调用getMax方法,获取n,n2,n3中的最大值输出
            case "大":
                int max = getMax(n1,n2,n3);
                System.out.println("最大值为:" + max);
                break;
            // 当str为小时,调用getMin方法,获取n,n2,n3中的最小值输出
            case "小":
                int min = getMin(n1,n2,n3);
                System.out.println("最小值为:" + min);
                break;
            default:
                System.out.println("指令有误!");
        }
    }

    private static int getMin(int i, int j, int k) {
        int min = i < j ? (i < k ? i : k) : (j < k ? j : k);
        return min ;
    }

    private static int getMax(int i, int j, int k) {
        int max = i > j ? (i > k ? i : k) : (j > k ? j : k);
        return max ;
    }
}
```

第六题：需求实现

- 定义printX方法，打印任意行的图形。
- 代码实现，效果如图所示：



```
O*****O
*O*****O*
**O*****O**
***O***O***
****O**O****
*****OO*****
*****OO*****
***O***O***
**O*****O**
*O*****O*
O*****O
```

- 开发提示：
 - 参考之前的练习，将代码抽取为一个方法。
- 参考答案：

```
public class Test6 {
    public static void main(String[] args) {

        printX(10);

    }

    /*
```

2.定义printX方法,参数为(int m) , 返回值无

3.printX方法中,使用for循环,初始化变量x = 1,如果x<=m进入循环,步进表达式x++

4.for循环内部,再嵌套定义一套for循环,初始化变量y = 1,如果y<=m进入循环,步进表达式y++

5.在内循环内部,判断x==y 或者 x+y==1+m ,则打印"O",否则打印"*"

```
*/
```

```
public static void printX(int m) {

    for (int x = 1; x <= m; x++) { //循环7行
        for (int y = 1; y <= m; y++) { //循环7列
            if (x == y || x + y == m + 1) { //对角线打印O
                System.out.print("O");
            } else {
                System.out.print("*"); //其他位置打印.
            }
        }
    }
}
```

```
        System.out.println();//换行
    }
}
}
```

第七题：需求实现

- 定义round方法，接收一位小数，实现四舍五入运算，并返回结果。
- 代码实现，效果如图所示：

```
10.1->10
10.4->10
10.5->11
10.9->11
```

- 开发提示：
 - round方法中，参数+0.5后，转换为int类型，并返回。
- 参考答案：

```
public class Test7 {
    public static void main(String[] args) {

        System.out.println( 10.1 + "->" + round(10.1));
        System.out.println( 10.4 + "->" + round(10.4));
        System.out.println( 10.5 + "->" + round(10.5));
        System.out.println( 10.9 + "->" + round(10.9));
    }
    // 定义round方法,参数为(double d) , 返回值int
    public static int round(double d) {
        // round方法中,d+0.5后,转换为int类型,并返回.
        int n = (int) (d + 0.5);
        return n;
    }
}
```