

Java第八天练习题

编写一个方法 `isEvenAndDivisibleByThree(int number)`，该方法接收一个整数参数 `number`，判断该数是否既是偶数又能被3整除。如果是，则返回 `true`，否则返回 `false`。

```
1 public static void main(String[] args) {
2     Scanner input = new Scanner(System.in);
3     System.out.print("请输入一个数: ");
4     int num = input.nextInt();
5     // 使用方法
6     boolean status = isEvenAndDivisibleByThree(num);
7     if (status == true){
8         System.out.print(num + " 是偶数又能被3整除");
9     }else {
10        System.out.println(num + " 不满足是偶数又能被3整除的条件");
11    }
12 }
13 // 编写一个方法 isEvenAndDivisibleByThree(int number),
14 // 该方法接收一个整数参数 number, 判断该数是否既是偶数又能被3整除。如果是, 则返回 true, 否则返回 false。
15 public static boolean isEvenAndDivisibleByThree(int number) {
16     boolean status = false;
17     if (number % 2 == 0 && number % 3 == 0) {
18         status = true;
19     }
20     return status;
21 }
```

编写一个方法 `isPositiveAndGreaterThanTen(double number)`，该方法接收一个双精度浮点数参数 `number`，判断该数是否是正数且大于 10。如果满足条件，则返回 `true`，否则返回 `false`。

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

```

2      Scanner input =new Scanner(System.in);
3      System.out.print("请输入一个数: ");
4      double num = input.nextDouble();
5      // 使用方法
6      boolean status = isPositiveAndGreaterThanTen(num);
7      if (status == true){
8          System.out.print(num + " 是正数且大于 10");
9      }else {
10         System.out.println(num + " 不满足是正数且大于 10");
11     }
12 }
13 // 编写一个方法 isPositiveAndGreaterThanTen(double number),
14 // 该方法接收一个双精度浮点数参数 number, 判断该数是否是正数且大于 10。
    如果满足条件, 则返回 true, 否则返回 false。
15 public static boolean isPositiveAndGreaterThanTen(double
number){
16     boolean status = false;
17     if (number > 0 && number > 10){
18         status = true;
19     }
20     return status;
21 }

```

编写一个方法 **isInvalidRange(int number)**, 该方法接收一个整数参数 **number**, 判断该数是否在有效范围内。有效范围定义为大于等于 0 且小于等于 100。如果在有效范围内, 则返回 **true**, 否则返回 **false**。

```

1  public static void main(String[] args) {
2      Scanner input =new Scanner(System.in);
3      System.out.print("请输入一个数: ");
4      int num = input.nextInt();
5      // 使用方法
6      boolean status = isInvalidRange(num);
7      if (status == true){
8          System.out.print(num + " 在有效范围内");
9      }else {
10         System.out.println(num + " 不在有效范围内(0 ≤ num ≤
100)");
11     }
12 }
13 // 编写一个方法 isInvalidRange(int number),

```

```

14 // 该方法接收一个整数参数number，判断该数是否在有效范围内。有效范围定义
    为大于等于 0 且小于等于 100。
15 // 如果在有效范围内，则返回 true，否则返回 false。
16 public static boolean isValidRange(int number){
17     boolean status = false;
18     if (number ≥ 0 && number ≤ 100){
19         status = true;
20     }
21     return status;
22 }

```

编写一个方法 **isBothConditionsTrue(boolean condition1, boolean condition2)**，该方法接收两个布尔类型的参数 **condition1** 和 **condition2**，判断这两个条件是否都为 **true**。如果两个条件都为 **true**，则返回 **true**，否则返回 **false**。

```

1 public static void main(String[] args) {
2     boolean num = isBothConditionsTrue(true,true);
3     System.out.println(num);
4 }
5 // 编写一个方法 isBothConditionsTrue(boolean condition1,boolean
    condition2),
6 // 该方法接收两个布尔类型的参数 condition1 和 condition2，判断这两个
    条件是否都为 true。
7 // 如果两个条件都为 true，则返回 true，否则返回false。
8 public static boolean isBothConditionsTrue(boolean
    condition1,boolean condition2){
9     boolean status = false;
10    if (condition1 == true && condition2 == true){
11        status = true;
12    }
13    return status;
14 }

```

编写一个方法 **isPalindromeNumber(int number)**，该方法接收一个四位整数参数 **number**，判断该数是否是回文数。回文数是指正序和倒序读都相同的整数。如果是回文数，则返回 **true**，否则返回 **false**。

```

1  public static void main(String[] args) {
2      Scanner input = new Scanner(System.in);
3      System.out.print("请输入一个四位数: ");
4      int num = input.nextInt();
5      // 使用方法
6      boolean status = isPalindromeNumber(num);
7      if (status == true){
8          System.out.print(num + " 是回文数");
9      }else {
10         System.out.println(num + " 不回文数");
11     }
12 }
13 // 编写一个方法 isPalindromeNumber(int number),
14 // 该方法接收一个四位整数参数 number, 判断该数是否是回文数。
15 // 回文数是指正序和倒序读都相同的整数。如果是回文数, 则返回 true, 否则
    返回false。
16 public static boolean isPalindromeNumber(int number){
17     int a = number / 1000;
18     int b = number / 100 % 10;
19     int c = number / 10 % 10;
20     int d = number % 10;
21     boolean status = false;
22     if (a == d && b == c){
23         status = true;
24     }
25     return status;
26 }

```

编写一个Java程序，实现一个简单的计算器。用户输入两个数字和一个操作符(如加法、减法、乘法或除法)，程序根据操作符进行相应的计算并输出结果。

```

1  public static void main(String[] args) {
2      Scanner input = new Scanner(System.in);
3      System.out.print("请输入第一个数: ");
4      double num1 = input.nextDouble();
5      System.out.print("请输入运算符: ");
6      String op = input.next();
7      System.out.print("请输入第二个数: ");
8      double num2 = input.nextDouble();
9      // 字符串需要使用equals()来判断是否相等
10     if ("+".equals(op) || "-".equals(op) || "*".equals(op) ||
        "/" .equals(op)){

```

```

11         double res = isCounter(num1, num2, op);
12         System.out.println("计算结果为: " + res);
13     }else {
14         System.out.println("输入的运算符不合法");
15     }
16 }
17
18 // 编写一个Java程序, 实现一个简单的计算器。
19 // 用户输入两个数字和一个操作符(如加法、减法、乘法或除法), 程序根据操作符
    进行相应的计算并输出结果。
20 public static double isCounter(double num1, double num2,
String op) {
21     double res = 0;
22     switch (op) {
23         case "+":
24             res = num1 + num2;
25             break;
26         case "-":
27             res = num1 - num2;
28             break;
29         case "*":
30             res = num1 * num2;
31             break;
32         case "/":
33             res = num1 / num2;
34             break;
35         case "%":
36             res = num1 % num2;
37             break;
38         default:
39             System.out.println("运算符不合法的运算符");
40     }
41     return res;
42 }

```

编写一个Java程序, 使用循环计算 π 的近似值。可以使用Leibniz级数来近似 π 的值: $\pi=4* (1-1/3+1/5-1/7+1/9-1/11+\dots)$,通过增加级数项的数量来改进近似结果。

```

1 public static void main(String[] args) {
2     // 编写一个Java程序, 使用循环计算 $\pi$ 的近似值。
3     // 可以使用Leibniz级数来近似 $\pi$ 的值:  $\pi=4* (1-1/3+1/5-1/7+1/9-$ 
     $1/11+\dots)$ ,通过增加级数项的数量来改进近似结果。
4     double res = isPI(100000000);

```

```

5      System.out.println("PI的近似值 = " + res);
6  }
7  public static double isPI(int num){
8      double res= 0;
9      for (int i = 1; i < num; i++) {
10         if (i % 2 == 0){
11             res += - 1.0/(2 * i -1);
12         }else {
13             res += 1.0/(2 * i -1);
14         }
15     }
16     return 4 * res;
17 }

```

编写一个Jva程序，使用循环实现一个简单的登陆验证系统。系统预先设置一个用户名和密码，用户最多有3次输入机会。如果输入的用户名和密码与预设值匹配，则输出“登陆成功”，否则提示剩余尝试次数。

```

1  public static void main(String[] args) {
2      // 编写一个Jva程序，使用循环实现一个简单的登陆验证系统。
3      // 系统预先设置一个用户名和密码，用户最多有3次输入机会。
4      // 如果输入的用户名和密码与预设值匹配，则输出“登陆成功”，否则提示剩
      余尝试次数。
5      Scanner input = new Scanner(System.in);
6      String user = "admin";
7      int password = 123456;
8      int count = 3;
9
10     for (int i = 3; i > 0; i--) {
11         System.out.print("请输入用户名: ");
12         String us = input.next();
13         System.out.print("请输入密码: ");
14         int pwd = input.nextInt();
15         if (user.equals(us) == false || password != pwd) {
16             count--;
17             if (count != 0){
18                 System.out.println("用户名错误, " + "还有" +
count + "次输入机会");
19             }else {
20                 System.out.println("3次机会已用完");
21             }

```

```
22         } else {
23             System.out.println("登陆成功");
24             break;
25         }
26     }
27 }
```

编写一个Jva程序，使用循环输出所有的完美数。完美数是等于其所有因子之和的正整数，如 $6=1+2+3$ 。

```
1  public static void main(String[] args) {
2      // 编写一个Jva程序，使用循环输出所有的完美数。
3      // 完美数是等于其所有因子之和的正整数，如 $6=1+2+3$ 。
4      perfectNumber(10000);
5  }
6
7  public static void perfectNumber(int number) {
8      for (int i = 1; i < number; i++) {
9          int sum = 0;
10         for (int j = 1; j ≤ i / 2; j++) {
11             if (i % j == 0) {
12                 sum += j;
13             }
14         }
15         if (sum == i) {
16             System.out.println(sum);
17         }
18     }
19 }
```