多重循环、方法、递归作业

一、 选择题

1. 分析下面的 Java 多重循环代码片段，编译运行后的输出结果是（ D ）。（选择一项）

```
for (int i = 0; i < 6; i++) {
    int k = ++i;
    while (k < 5) {
        System.out.print(i);
        break;
    }
}
```

A. 024
B. 02
C. 123
D. 13

2. 给定如下 Java 代码，编译运行的结果是（ A）。（选择一项）

```
public class Test {
    public static void main(String[] args) {
        int sum=0;
        for(int i=1;i<10;i++){
            do{
                i++;
                if(i%2!=0)
                    sum+=i;
            }while(i<6);
        }
        System.out.println(sum);
    }
}
```

A 8
B. 15
C. 24
D. 什么也不输出

3. Java 中 main 方法的返回值是（ D ）。（选择一项）

A String

B. int

C. char

D. void

4. 在 **Java** 的程序类中如下方法定义正确的是（ AD）。（选择两项）

A
```
public int ufTest(int num){
        int sum=num+100;
        return sum;
}
```

B.
```
public String ufTest(int num){
        int sum=num+100;
        return sum;
}
```

C.
```
public void ufTest(int num){
        int sum=num+100;
        return sum;
}
```

D.
```
public float ufTest(int num){
        int sum=num+100;
        return sum;
}
```

5. 以下关于方法调用的代码的执行结果是（ B ）。（选择一项）

```
public class Test {
    public static void main(String args[]) {
        int i = 99;
        mb_operate(i);
        System.out.print(i + 100);
    }
    static void mb_operate(int i) {
        i += 100;
    }
}
```

A. 99

B. 199

C. 299

D. 99100

6. 下面（ D ）方法是 **public void example(){...}**的重载方法。（选择二项）

A private String example(){...}

B. public int example(String str){...}

C. public void example2(){...}

D. public int example(int m,float f){...}

7. **以下选项中添加到代码中横线处会出现错误的是（ BD ）。（选择二项）**

```
public class Test {
    public float aMethod(float a, float b) {
        return 0;
    }
    ____________________________________
}
```

A
```
public float aMethod(float a, float b, float c) {
        return 0;
    }
```

B.
```
public float aMethod(float c, float d) {
        return 0;
    }
```

C.
```
public int aMethod(int a, int b) {
        return 0;
    }
```

D.
```
private int aMethod(float a, float b) {
        return 0;
    }
```

二、 判断题

1. 多重循环是指一个循环体内又包含另一个完整的循环结构。外层循环变量变化一次，内层循环变量要从头到尾变化一遍。（ T）
2. 定义方法时的参数称为实在参数，调用方法时的参数称为形式参数。（ F ）
3. 调用方法时，要求实参和形参的个数相同，类型匹配。（ T ）
4. 判断方法重载的依据是方法名相同，参数不同。（ T ）
5. 程序调用自身的编程技巧称为递归。递归的特点是编程简单并且节省内存。(F)
6. 任何可用递归解决的问题也能使用循环解决，递归既花时间又耗内存，在要求高性能的情况下尽量避免使用递归。（ T ）

三、 简答题

1. 在多重循环中，如何在内层循环中使用 break 跳出外层循环。
2. 方法重载的定义、作用和判断依据
3. 递归的定义和优缺点
4. 方法的定义中，是否必须要有返回值类型?如果确实不需要返回值，使用哪个关键字声明?
5. 方法的定义中，return 是否必须?它有什么作用?
6. java 中，参数的传递使用值传递还是引用传递?
7. 方法定义中，形式参数和实际参数有什么区别?用自己的话描述。
8. 方法定义时，使用实参还是形参?
9. 定义形参时，必须要写变量类型吗?

10. 实参类型是否必须和形参类型匹配?

四、 编码题

1. 打印九九乘法表
2. 判断 1-100 之间有多少个素数并输出所有素数。（将判断一个数是否是素数的功能提取成方法，在循环中直接调用即可）
3. 输入三个班，每班 10 个学生的成绩，求和并求平均分
4. 编写递归算法程序：一列数的规则如下: 1、1、2、3、5、8、13、21、34…… 求数列的第 40 位数是多少。

五、 可选题

1. 打印出所有的"水仙花数"，所谓"水仙花数"是指一个三位数，其各位数字立方和等于该数本身。例如：153 是一个"水仙花数"，因为 153=1 的三次方＋5 的三次方＋3 的三次方。
2. 定义方法：打印指定行和列的矩形
3. 编写 Java 程序，实现接收用户输入的正整数，输出该数的阶乘。要求：限制输入的数据在 1-10 之间，无效数据进行提示，结束程序。要求使用递归和循环分别实现。输出结果如：4！=1*2*3*4=24

1.　在多重循环中，如何在内层循环中使用 break 跳出外层循环。
在内层循环中标记break跳出操作
2. 方法重载的定义、作用和判断依据
方法名一样，参数不一样。
3. 递归的定义和优缺点
定义：直接或间接调用自己本身的函数；优点：结构清晰、可读性强；缺点：运行效率低，耗费计算时间与存储空间。
4. 方法的定义中，是否必须要有返回值类型?如果确实不需要返回值，使用哪个关键字声明?
不用，void
5. 方法的定义中，return 是否必须?它有什么作用?
不必须，返回值，如果无返回值不需要return
6. java 中，参数的传递使用值传递还是引用传递?
值传递
7. 方法定义中，形式参数和实际参数有什么区别?用自己的话描述。
形参是定义方法里括号里的，实际参数是你带入的数值
8. 方法定义时，使用实参还是形参?
形参
9. 定义形参时，必须要写变量类型吗?
必须写
10. 实参类型是否必须和形参类型匹配?
必须匹配

1

```java
public class MultiplicationTable {
    新 *
    public static void main(String[] args) {
        for (int i = 1; i < 10; i++) {
            for (int j = 1; j <= i ; j++) {
                System.out.print(j + "*" + i + "=" + j * i + "" + "\t");
            }
            System.out.println();
        }
    }
}
```

2.

```java
public class DeterminingPrimeNumbers {
    1个用法  新 *
    public static boolean determinePrimeNumber(int n){
        if (n == 1){
            return false;
        }
        for (int i = 2; i < n; i++) {
            if (n % i == 0){
                return false;
            }
        }
        return true;
    }
    新 *
    public static void main(String[] args) {
        int count = 0;
        for (int i = 1; i <= 100 ; i++) {
            if (determinePrimeNumber(i)){
                count++;
            }
        }
        System.out.println(count);
    }
}
```

3

```java
public class SumAndMeanOfClass {
    新 *
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int [][] students = new int[3][10];
        for (int i = 0; i < students.length; i++) {
            for (int j = 0; j < students[i].length; j++) {
                int grade = input.nextInt();
                students[i][j] = grade;
            }
        }
        int count = 0;
        int sum = 0;
        for (int[] student : students) {
            for (int grade : student) {
                count++;
                System.out.println(grade);
                sum += grade;
            }
        }
        System.out.println("sum: " + sum);
        System.out.println("mean: " + sum / count);

    }
}
```

4.

```java
public class Recursive {
    3 个用法  新 *
    public static int recursivePracticing(int n){
        if (n == 0 || n == 1){
            return n;
        }
        return recursivePracticing( n: n - 2) + recursivePracticing( n: n - 1);
    }

    新 *
    public static void main(String[] args) {
        System.out.println(recursivePracticing( n: 3));
    }
}
```