**多重循环、方法、递归作业**

1. **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（ BD（A我觉得也是） ）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（ CD ）。（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;  } |

1. **判断题**
2. 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。（ √ ）
3. 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。反了（ × ）
4. 调用方法时，要求实参和形参的个数相同，类型匹配。（ √ ）
5. 判断方法重载的依据是方法名相同，参数不同。参数类型也可以不同，返回值也可以不同，修饰符也可以不同（ × ）
6. 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。更废内存（ × ）
7. 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。（ √ ）
8. **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。

在外层循环前加标识，break后也加相应标识

* 1. 方法重载的定义、作用和判断依据

在一个类中，方法名相同，返回值类型，参数类型，参数个数不同的方法叫重载

作用是同样的名字实现一样的方法，提供更多选择性

* 1. 递归的定义和优缺点

方法自己调用自己就是递归

优点是代码简洁

缺点是疯狂堆栈

* 1. 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?

不需要，void

* 1. 方法的定义中，return是否必须?它有什么作用?

如果方法有返回值，则是必须

Return是把数据返回给调用者，也是函数的结束

* 1. java中，参数的传递使用值传递还是引用传递?

基本数据类型是用值传递，引用类型是引用传递

* 1. 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。

形参就是在定义时，预先使用的，完成接下来参与运算的逻辑

实参是实际调用者传送过来的值，它会替代形参，走完形参的逻辑步骤

* 1. 方法定义时，使用实参还是形参?

形参

* 1. 定义形参时，必须要写变量类型吗?

是的，泛型或者Object、Class也可以

* 1. 实参类型是否必须和形参类型匹配?

是的

* 1. 模仿老师讲“网吧上网的例子”，讲一个自己的事例来说明方法的定义，形参和实参。

形参就像是过山车，你坐不坐他都按着轨道跑

人就是实参，你会替代过山车（有点不恰当，不是完全替代。但代码是完全替代）在预设的路线上跑

1. **编码题**
   1. 打印九九乘法表

文本

描述已自动生成

* 1. 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）

上一篇作业里写了两遍同样的题

* 1. 输入三个班，每班10个学生的成绩，求和并求平均分

文本

描述已自动生成

* 1. 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。

文本

描述已自动生成

1. **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。

文本

描述已自动生成

* 1. 定义方法：打印指定行和列的矩形

文本

描述已自动生成

* 1. 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

文本

描述已自动生成