**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（ B ）。（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（ A D ）。（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 ）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){…}重载返回值类型不管，因为有的都没有返回值，方法名必须相同，参数不一样（参数类型、个数） |

|  |  |  |
| --- | --- | --- |
| **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;  }    }方法重载：参数个数，参数类型不同 | |
|  |  |  |
|  | **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。（ T ）
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。（ F ）
4. 调用方法时，要求实参和形参的个数相同，类型匹配。（ T ）
5. 判断方法重载的依据是方法名相同，参数不同。（ T ）
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。( F )
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。（ F ）
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。
   2. 方法重载的定义、作用和判断依据
   3. 方法重载时指方法名相同，参数（个数，类型）不同，可以处理不同情况的问题
   4. 递归的定义和优缺点
9. **编码题**
   1. 打印九九乘法表
10. **package** BaseProgramming.homework;  
      
    **public class** Demo {  
     **public static void** multiplicationTabe(**int** num) {  
     **if** (num == 1) {  
     **for** (**int** i = 1; i < 10; i++) {  
     **for** (**int** j = 1; j <= i; j++) {  
     System.***out***.print(i + **"\*"** + j + **"="** + i \* j + **"\t"**);  
     }  
     System.***out***.println();  
     }  
     }  
     **else if**(num == 2){  
     **for** (**int** i = 1; i < 10; i++) {  
     **for** (**int** j = 9; j >= i; j--) {  
     System.***out***.print(i + **"\*"** + j + **"="** + i \* j + **"\t"**);  
     }  
     System.***out***.println();  
     }  
     }  
     }  
      
      
     **public static void** main(String[] args) {  
     Demo.*multiplicationTabe*(2);  
     }  
    }
    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）
    2. 输入三个班，每班10个学生的成绩，求和并求平均分
11. **package** BaseProgramming.homework;  
      
    **public class** Demo {  
     **public static void** multiplicationTabe(**int** num) {  
     **if** (num == 1) {  
     **for** (**int** i = 1; i < 10; i++) {  
     **for** (**int** j = 1; j <= i; j++) {  
     System.***out***.print(i + **"\*"** + j + **"="** + i \* j + **"\t"**);  
     }  
     System.***out***.println();  
     }  
     }  
     **else if**(num == 2){  
     **for** (**int** i = 1; i < 10; i++) {  
     **for** (**int** j = 9; j >= i; j--) {  
     System.***out***.print(i + **"\*"** + j + **"="** + i \* j + **"\t"**);  
     }  
     System.***out***.println();  
     }  
     }  
     }  
      
      
     **public static void** main(String[] args) {  
     Demo.*multiplicationTabe*(2);  
     }  
    }
    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。
12. **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。
13. **package** BaseProgramming.homework;  
      
    **import** java.util.Scanner;  
      
    **public class** Demo {  
      
     **public static void** narcissisticNumber() {  
     **int** number = 0;  
     **while** (number < 1000) {  
     **int** bai = number / 100;  
     **int** shi = (number - bai \* 100) / 10;  
     **int** ge = number - bai \* 100 - shi \* 10;  
     **if** (number == (bai \* bai \* bai + shi \* shi \* shi + ge \* ge \* ge)) {  
     System.***out***.println(number);  
     }  
     number++;  
     }  
     }  
      
     **public static void** main(String[] args) {  
     Demo.*narcissisticNumber*();  
     }  
    }
    1. 定义方法：打印指定行和列的矩形
14. **package** BaseProgramming.homework;  
      
    **import** java.util.Scanner;  
      
    **public class** Demo {  
     **public static void** lineCol(**int** l,**int** c){  
     **for**(**int** i=0;i<l+1;i++){  
     **for**(**int** j=0;j<c+1;j++){  
     System.***out***.print(**"\*"**);  
     }  
     System.***out***.println();  
     }  
     }  
      
     **public static void** main(String[] args) {  
     Demo.*lineCol*(12,34);  
     }  
    }
    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