**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（ A ）。（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（ C ）。（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){…} |

|  |  |  |
| --- | --- | --- |
| **7.** | **以下选项中添加到代码中横线处会出现错误的是（ AD ）。（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。（ **〇** ）
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。（ **×** ）
4. 调用方法时，要求实参和形参的个数相同，类型匹配。（ **〇** ）
5. 判断方法重载的依据是方法名相同，参数不同。（ **〇** ）
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。( **×** )
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。（ **〇** ）
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. 递归的定义和优缺点
9. **编码题**
   1. 打印九九乘法表

**public class** While{  
 **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+**"="**+i\*j+**"\t"**);  
 }  
 System.***out***.println();  
 }  
 }  
}

执行结果：

1\*1=1

1\*2=2 2\*2=4

1\*3=3 2\*3=6 3\*3=9

1\*4=4 2\*4=8 3\*4=12 4\*4=16

1\*5=5 2\*5=10 3\*5=15 4\*5=20 5\*5=25

1\*6=6 2\*6=12 3\*6=18 4\*6=24 5\*6=30 6\*6=36

1\*7=7 2\*7=14 3\*7=21 4\*7=28 5\*7=35 6\*7=42 7\*7=49

1\*8=8 2\*8=16 3\*8=24 4\*8=32 5\*8=40 6\*8=48 7\*8=56 8\*8=64

1\*9=9 2\*9=18 3\*9=27 4\*9=36 5\*9=45 6\*9=54 7\*9=63 8\*9=72 9\*9=81

* 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

方法1：

**import** java.util.Scanner;  
  
**public class** AverageScore {  
 **public static void** main(String[] args) {  
 **double** sum = 0;  
 Scanner input = **new** Scanner(System.***in***);  
 **for**(**int** i=1;i<=3;i++){  
 System.***out***.println(**"请输入第"**+i+**"个班的成绩"**);  
 **for**(**int** j=1;j<=10;j++){  
 System.***out***.println(**"请输入第"**+j+**"个学生的成绩："**);  
 **double** score = input.nextDouble();  
 sum+=score;  
 }  
 **double** average = sum/10;  
 System.***out***.println(**"第"**+i+**"班的总成绩为："**+sum+**",平均分为："**+average);  
 sum = 0;  
 }  
 }  
}

方法2：

**import** java.util.Scanner;  
**public class** StuScoreCal {  
 **public static void** main(String[] args) {  
 Scanner sc = **new** Scanner(System.***in***);  
 **for**(**int** i = 1;i<=3;i++){  
 **//输入每个班成绩并求和** System.***out***.println(**"请输入第"**+i+**"个班的成绩"**);  
 **double** sum = 0;  
 **for**(**int** j=0;j<10;j++){  
 System.***out***.println(**" 请输入第"**+(j+1)+**"个学生成绩："**);  
 **double** score = sc.nextDouble();  
 System.***out***.println(score);  
 sum += score;  
 }  
 **//求平均分  
 double** avg = sum / 10;  
 **//输出结果** System.***out***.println(**"第"**+i+**"个班学生总分："**+sum+**" 平均分："**+avg); }  
 }  
}

* 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。

**public** **class** TestSum{

**public** **static** **void** main(String[] args) {

System.***out***.println(*sum*(40));

}

**public** **static** **long** sum(**int** n) {

**if** (n <= 0) {

**return** 0;

} **else** **if** (n == 1 || n == 2) {

**return** 1;

} **else** {

**return** *sum*(n - 1) + *sum*(n - 2);

}

}

}

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。

**public class** Number {  
 **public static void** main(String[] args) {  
 **for**(**int** i=1;i<=9;i++){  
 **for**(**int** j=0;j<=9;j++){  
 **for**(**int** k=0;k<=9;k++){  
 **int** hundredth=(**int**)(Math.*pow*(i,3));  
 **int** tenth=(**int**)(Math.*pow*(j,3));  
 **int** single=(**int**)(Math.*pow*(k,3));  
 **int** sum1=hundredth+tenth+single;  
 **int** sum2=i\*100+j\*10+k;  
 **if** (sum1 == sum2){  
 System.***out***.println(sum2);  
 }  
 }  
 }  
 }  
 }  
}

**public** **class** Prog3{

**public** **static** **void** main(String[] args){

**for**(**int** i=100;i<1000;i++){

**if**(*isLotus*(i))

System.***out***.print(i+" ");

}

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

}

//判断水仙花数

**private** **static** **boolean** isLotus(**int** lotus){

**boolean** flag=**false**;

**int** a = lotus % 10;

**int** b = lotus / 10 % 10;

**int** c = lotus / 100 % 10;

**if** (Math.*pow*(a, 3) + Math.*pow*(b, 3) + Math.*pow*(c, 3) == lotus) {

flag = **true**;

}

**return** flag;

}

}

* 1. 定义方法：打印指定行和列的矩形

**public** **class** TestRect {

**public** **static** **void** main(String[] args) {

// 调用printStr方法

*printStr*(5, 4, "\*");

}

**public** **static** **void** printStr(**int** rows, **int** columns, String str) {

**for** (**int** i = 0; i < rows; i++) {

**for** (**int** j = 0; j < columns; j++) {

System.***out***.print(str);

}

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

}

}

}

* 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

**import** java.util.Scanner;  
  
**public class** Number {  
 **public static void** main(String[] args) {  
 Scanner input = **new** Scanner(System.***in***);  
 System.***out***.println(**"请输入1个1到10之间任意的数字："**);  
 **int** number = input.nextInt();  
 **if**(number<1 | number>10){  
 System.***out***.println(**"无效数字"**);  
 **return**;  
 }  
 String str = *testString*(number);  
 System.***out***.println(number+**"的阶乘为："**+str+**"="**+*numberGet*(number));  
 }  
  
 **public static** String testString(**int** n){  
 String result = **""**;  
 **for** (**int** i=1;i<=n;i++){  
 result +=(i+**"\*"**);  
 }  
 **return** result.substring(0, result.length()-1);  
 }  
  
  
  
 **public static int** numberGet(**int** i){  
 **if**(i==1){  
 **return** 1;  
 }**else**{  
 **return** i\**numberGet*(i-1);  
 }  
 }  
}

**public** **class** Test01 {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.***in***);

//输入一个数

System.***out***.println("请输入你要判断的数据：");

**int** n = input.nextInt();

**if**(n>10 || n<1){

System.***out***.println("限制输入的数据在1-10之间");

System.*exit*(0);

}

String str = *testString*(n);

**long** sum = *factRec*(n);//递归

System.***out***.println(n+"!="+str+"="+sum);

**long** sum2 = *factfor*(n);//for循环

System.***out***.println(n+"!="+str+"="+sum2);

}

//实现字符串的输出

**public** **static** String testString(**int** n){

String result = "";

**for** (**int** i=1;i<=n;i++){

result +=(i+"\*");

}

**return** result.substring(0, result.length()-1);

}

//递归实现

**public** **static** **long** factRec(**int** n){

**if**(n==1){

**return** 1;

}**else**{

**return** n\**factRec*(n-1);

}

}

//for循环实现

**public** **static** **long** factfor(**int** n){

**int** sum = 1;

**for** (**int** i = 1; i < n; i++) {

sum += sum \* i;

}

**return** sum;

}

}