**选择语句+循环语句作业**

1. **填空题**
2. for循环的语法格式是for (表达式1;表达式2;表达式3) {循环体}，其中在整个循环过程中只执行一次的部分是 表达式1 。
3. 在循环结构中，如果想跳出循环体，结束整个循环结构可以使用 break 语句。
4. \_\_\_continue\_语句用在循环语句体中，用于终止某次循环过程，即跳过循环体中尚未执行的语句，接着进行下一次是否执行循环的判定。即只结束本次循环，而不是终止整个循环的执行。
5. **选择题**

|  |  |  |
| --- | --- | --- |
| 1. **.** | **以下四个选项中和下面代码功能相同的是（ C ）。（选择一项）** | |
|  | **int** i = 1;  **int** sum = 0;  **while** (i <= 100) {  **if** (i % 2 == 0)  sum = sum + i;  i++;  } | |
|  |  |  |
|  | **A** | for (int x =1; x<=100;x++){ sum=sum+x;} |
|  | **B.** | for (int x =0; x<=100;x+=2){ sum=sum+x;} |
|  | **C.** | for (int x =1; x<=100;x+=2){ sum=sum+x;} |
|  | **D.** | 上述全对 |

|  |  |  |
| --- | --- | --- |
|  | **while循环和do-while循环的区别是（ D ）。（选择一项）** | |
|  |  |  |
|  | **A.** | 没有区别，这两个结构在任何情况下效果一样 |
|  | **B.** | while循环比do-while循环执行效率高 |
|  | **C.** | while循环是先循环后判断，所以循环体至少被执行一次 |
|  | **D.** | do-while循环是先循环后判断，所以循环体至少被执行一次 |

|  |  |  |
| --- | --- | --- |
|  | **在Java中有如下代码，则编译运行该类的输出结果是（ D ）。（选择一项）** | |
|  | **public** **static** **void** main(String[ ] args) {  **for**(**int** i=0;i<10;i++){  **if** (i%2!=0)  **return**;  System.*out*.print(i);  }  } | |
|  |  |  |
|  | **A** | 13578 |
|  | **B.** | 02468 |
|  | **C.** | 0123456789 |
|  | **D.** | 0 |

|  |  |  |
| --- | --- | --- |
|  | **下面程序执行的结果是在屏幕上打印（ A ）次Java基础班。（选择一项）** | |
|  | **for**(**int** i=1;i<=10;i++){  **if** (i<5)  **continue**;  System.***out***.println("Java基础班");  } | |
|  |  |  |
|  | **A** | 5 |
|  | **B.** | 6 |
|  | **C.** | 7 |
|  | **D.** | 8 |

|  |  |  |
| --- | --- | --- |
|  | **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 |

|  |  |  |
| --- | --- | --- |
|  | **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 |

1. **判断题（共20个题目，总计10分）**
2. while循环结构的特点是先循环再判断，循环体至少执行一次。（×）
3. for循环的语法格式是for (表达式1;表达式2;表达式3) {循环体}，其中三个表达式都可以省略。( √)
4. break语句可以出现在switch语句和循环语句中。（ √ ）
5. continue语句可以出现在switch语句和循环语句中。（× ）
6. 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。（ √ ）
7. **简答题**
   1. while和do-while语句的异同之处

相同点：

都是循环语句 可以通过条件进行循环

不同点：

语法不同

While(<条件>){

//循环体

}

do{

//循环体

}While(<条件>);

最少循环次数不同

while最少循环0次

Do while 最少循环1次

* 1. break和continue语句的作用

break跳出循环

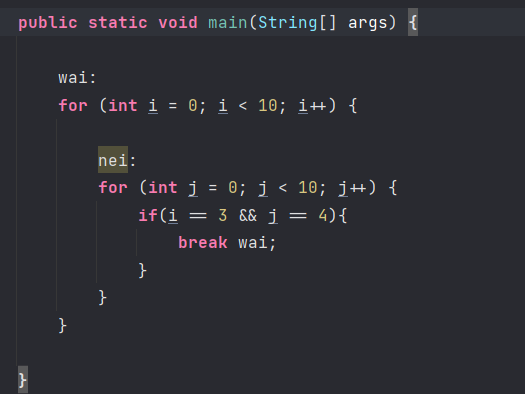
continue跳出本次循环 继续下次循环

* 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C/C++中可以通过关键字goto语句跳转到指定位置

在Java中通过给定义外层循环，然后用 break <外层循环定义关键字>;跳出

如下所示 外层循环定义为wai,内层循环定义为nei,通过break wai; 跳出外层



1. **编码题**
2. 判断一个数是否是素数。

public static void main(String[] args) {

int n = new Scanner(System.in).nextInt();

if(n < 2) {

System.out.println("不是素数");

return;

}

if(n == 2 || n % 2 == 0){

System.out.println("是素数");

return;

}

boolean isSu = true;

for (int i = 2; i < n; i++) {

if(n % i == 0){

isSu = false;

break;

}

}

if(isSu){

System.out.println("是素数");

}else {

System.out.println("不是素数");

}

}

1. 输入三个班，每班10个学生的成绩，求平均分、最高分最低分

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

for (int i = 1; i <=3 ; i++) {

double max=Double.MIN\_NORMAL,min=Double.MAX\_VALUE,sum=0;

for (int j = 1; j <= 10; j++) {

System.out.print("请输入第" + i + "个班中第" + j + "个同学的成绩: ");

double score = scanner.nextDouble();

if(score > max) max = score;

if(score < min) min = score;

sum += score;

}

System.out.println("第" + i + "个班级中\n\t最高分:" + max + "\t最低分:" + min + "\t平均分:" + sum/10);

}

}