**选择语句+循环语句作业**

1. **填空题**
2. Java中有两种类型的选择结构的控制语句，分别是if语句和（ switch语句 ）
3. 在Java JDK1.7之前，switch只能支持byte、short、char、int或者其对应的封装类以及Enum类型。在JDK1.7中又加入了类型（ String ）
4. 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<初始化语句> ）。
5. 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语句（ break ）。
6. （ 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。
7. 使用Math.random( )返回带正号的 double值，该值大于等于0.0且小于1.0。使用该函数生成[30,60]之间的随机整数的语句（ (int)(Math.random( )\*30) + 30 ）
8. **选择题**

|  |  |  |
| --- | --- | --- |
| **1.** | **以下代码的执行结果是（ B ）。（选择一项）** | |
|  | **boolean** m = **false**;  **if**(m = **false**){  System.*out*.println("false");  }**else**{  System.*out*.println("true");  } | |
|  |  |  |
|  | **A.** | false |
|  | **B.** | true |
|  | **C.** | 编译错误 |
|  | **D.** | 无结果 |

|  |  |  |
| --- | --- | --- |
| **2.** | **分析如下Java代码，编译运行的输出结果是（ A ）。（选择一项）** | |
|  | **public static void** main(String[ ] args) {  **boolean** a=**true**;  **boolean** b=**false**;  **if** (!(a&&b)) {  System.*out.*print("!(a&&b)");  }**elseif** (!(a||b)) {  System.*out*.println("!(a||b)");  }**else** {  System.*out*.println("ab");  }  } | |
|  |  |  |
|  | **A** | !(a&&b) |
|  | **B.** | !(a||b) |
|  | **C.** | ab |
|  | **D.** | !(a||b)ab |

|  |  |  |  |
| --- | --- | --- | --- |
| **3.** | **下列选项中关于变量x的定义，（ BD ）可使以下switch语句编译通过。（选择二项）** | | |
|  | **switch**(x) {  **case** 100 :  System.*out*.println("One hundred");  **break**;  **case** 200 :  System.*out*.println("Two hundred");  **break**;  **case** 300 :  System.*out*.println( "Three hundred");  **break**;  **default** :  System.*out*.println( "default");  } | | |
|  |  | |  |
|  | **A** | | double x = 100; |
|  | **B.** | | char x = 100; |
|  | **C.** | | String x = "100"; |
|  | **D.** | | int x = 100; |
| **4.** | **阅读下列文件的Java代码，其执行结果是（ D ）。（选择一项）** | | |
|  | **public class** Test {  **public static void** main(String[] args) {  **char** ch = 'c';  **switch** (ch) {  **case** 'a':  System.***out***.print("a"); **break**;  **case** 'b':  System.***out***.print("ab");  **case** 'c':  System.***out***.print("c");  **default**:  System.***out***.print("d");  }  }  } | | |
|  |  | |  |
|  | **A** | | a |
|  | **B.** | | b |
|  | **C.** | | c |
|  | **D.** | | cd |
| **5.** | | **以下Java程序编译运行后的输出结果是（ B ）。（选择一项）** | |
|  | | **public class** Test {  **public static void** main(String[] args) {  **int** i = 0, sum = 0;  **while** (i <= 10) {  sum += i; // 0+1+2+3+4+5+6+7...+9+10;  i++;  **if** (i % 2 == 0)  **continue**;  }  System.*out*.println(sum);  }  } | |
|  | |  |  |
|  | | **A** | 0 |
|  | | **B.** | 55 |
|  | | **C.** | 50 |
|  | | **D.** | 36 |

|  |  |  |
| --- | --- | --- |
| **6.** | **以下四个选项中和下面代码功能相同的是（ B ）。（选择一项）** | |
|  | **int** i = 1;  **int** sum = 0;  **while** (i <= 100) { //循环100次， i的值是1~100之间累加  **if** (i % 2 == 0) //累加偶数  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 |

|  |  |  |
| --- | --- | --- |
| **7.** | **以下do-while循环代码的执行结果是（ A ）。（选择一项）** | |
|  | **int** a=0;  **int** c=0;  **do**{  --c;  a=a-1;  }**while**(a>0);  System.*out*.println(a+""+c); | |
|  |  |  |
|  | **A.** | -1 -1 |
|  | **B.** | 死循环 |
|  | **C.** | -1 -2 |
|  | **D.** | -1 0 |

|  |  |  |
| --- | --- | --- |
| **8.** | **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 |

|  |  |  |
| --- | --- | --- |
| **9.** | **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) //是奇数  break;  System.*out*.print(i);  }  } | |
|  |  |  |
|  | **A** | 13578 |
|  | **B.** | 02468 |
|  | **C.** | 0123456789 |
|  | **D.** | 0 |

|  |  |  |
| --- | --- | --- |
| **10.** | **下面程序执行的结果是在屏幕上打印（ B ）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 |

1. **判断题（共20个题目，总计10分）**
2. if语句的条件表达式的结果都必须是boolean值。( √ )
3. switch选择语句是多分支选择语句，只能处理等值条件判断的情况，表达式可以是int类型、char类型（还可以有....类型），但不能是double,float类型。( √ )
4. while循环结构的特点是先循环再判断，循环体至少执行一次。( × )
5. 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。( √ )
6. break语句可以出现在switch语句和循环语句中。( √ )
7. continue语句可以出现在switch语句和循环语句中。( × )
8. 如果if语句后面不写{},if的控制范围只限于第一句。( √ )
9. **简答题**
   1. 三种控制结构是什么?  
      顺序结构（从上往下一条一条执行，程序的默认执行流程）  
      选择结构（分支结构）：if、switch  
      循环结构：for、while、do-while
   2. 分别叙述单选择、双选择、多选择的区别。  
      单选  
      if(){  
        
      }else if(){  
        
      }else if(){  
        
      }  
        
      多选  
      if(){  
        
      }  
      if(){  
        
      }  
      if(){  
        
      }
   3. if多分支语句和switch语句的异同之处。  
      switch语句只能用作“等值判断”
   4. switch语句的功能是否完全可以使用if -else if -else多选择结构来代替?如果是，为什么还需要switch结构?  
      当然可以代替！

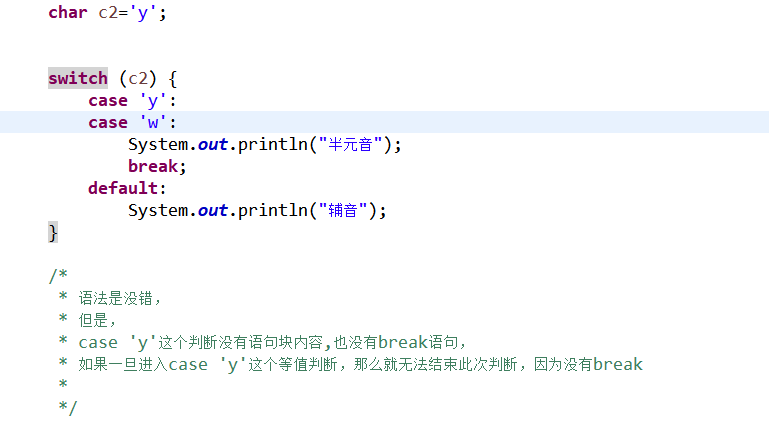
Switch语句的代码可读性强，而且可以避免出现重复的等值判断！

* 1. switch中的表达式的结果需要是什么类型?  
     整型：byte、short、char、int  
     枚举：enum  
     字符串：String
  2. switch语句中，一个case开始执行后，什么时候结束执行这个case的语句?  
     break
  3. switch语句中，default是否必须写?

可以不写，相当于就是if语句的else

* 1. 下面代码有什么错误?

|  |
| --- |
| switch (c2) {  case 'y':  case 'w':  System.out.println("半元音");  break;  default:  System.out.println("辅音");  } |



* 1. 循环结构模拟了现实世界中的什么逻辑?  
     多次重复，直到一个停止的标准

比如：我给车加油，一直加加到满为之

我开车，一直开，开到没油为止

我摘苹果，摘到我框子装满为止

我买各种喜欢的东西，买到穷为止

* 1. 循环结构的四种内部结构是什么?  
     while  
     do-while

for  
 for-each（增强for循环）

* 1. while和do-while语句的异同之处

不同点在于，一个会至少执行一次，然后再判断是否要执行下一次

共同之处就是可以完成循环操作

* 1. for循环相比while循环有什么优势?  
     for循环适用于循环次数确定的  
     while循环适用于循环次数不确定的
  2. 什么情况下，会出现死循环?并写出一个例子来。  
     死循环：无限循环，一直循环下去，循环无法结束。

当循环条件和迭代语句控制的不对时，就会出现死循环

* 1. break和continue语句的作用
  2. 下面代码都有什么问题：

|  |
| --- |
| a的初始值是什么？？ do{  System.out.println(a);  a++;  } while (a<0) |

没有声明变量a

* 1. 下面代码有错误吗?如果有，指出错误：

|  |
| --- |
| for(int i=0;i<=10;i++){  System.out.println(i); //循环体  }  System.out.println("(((((("+i);  变量作用域问题， 这个for循环的外部不能访问到i |

* 1. 语句块能否使用外部的变量?语句块中定义的变量，外部能否使用？  
     语句块中当然能够使用外部的变量！！！（成员变量听过没？？）  
     语句块中定义的变量只能自己语句块中使用，不能外部使用！！！

1. **编码题**
2. 输入一个数，判断是奇数还是偶数

文本

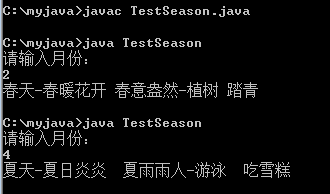
描述已自动生成

1. 根据成绩输出对应的等级，使用if多分支和switch语句分别实现。
   1. A级 [90,100]
   2. B级 [80,90)
   3. C级 [70,80)
   4. D级 [60,70)
   5. E级 [0,60)

文本

描述已自动生成

1. 根据月份，输出对应的季节，并输出至少两个描述该季节的成语和活动。



文本

描述已自动生成

1. 判断一个数是否是素数。

文本

描述已自动生成

1. 从键盘输入一个班5个学生的分数，求和并输出。

电脑萤幕画面

描述已自动生成

1. 从键盘输入某个十进制整数数，转换成对应的二进制整数并输出。

文本

描述已自动生成

1. 用while和for循环输出1-1000之间能被5整除的数，且每行输出3个。

文本

描述已自动生成

1. 编程求：∑1+∑2+……+∑100。

文本

描述已自动生成

1. 打印101-150之间所有的质数。

文本

描述已自动生成

1. 生成0-100随机数，直到生成88为止，停止循环！

文本

描述已自动生成

1. **可选题**
   1. 根据考试成绩输出对应的礼物，90分以上爸爸给买电脑，80分以上爸爸给买手机， 60分以上爸爸请吃一顿大餐，60分以下爸爸给买学习资料。

要求：该题使用多重if完成

文本

描述已自动生成

* 1. 请根据英文单词的第一个字母判断星期几，如果第一个字母是一样的，则继续判断第二个字母。例如如果第一个字母是S,则继续判断第二个字母，如果第二个字母是a,则输出“星期六”。星期的英文单词如下表所示。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 星期一 | 星期二 | 星期三 | 星期四 | 星期五 | 星期六 | 星期日 |
| Monday | Tuesday | Wednesday | Thursday | Fridy | Saturday | Sunday |

文本

描述已自动生成

* 1. 输入一批整数，使用循环求出最大值与最小值，输入0时结束。

文本

描述已自动生成

* 1. 给20块钱买可乐，每瓶可乐3块钱，喝完之后退瓶子可以换回1块钱，问最多可以喝到多少瓶可乐。

文本

描述已自动生成

* 1. 从键盘输入某个十进制小数，转换成对应的二进制小数并输出。（查询十进制小数转换成二进制小数的算法，使用循环来实现。最多保留小数位后7位数字即可）

文本

描述已自动生成

* 1. 编程求：1！+2！+……+15! (阶乘)。

文本

描述已自动生成

* 1. 用while和for循环分别计算100以内奇数和偶数的和，并输出。

文本

描述已自动生成

* 1. 把100~150之间不能被3整除的数输出。

文本

描述已自动生成

* 1. 思考：1！+2！+……+100! 可能会超过基本类型的表数范围，我们应该选用哪个类来处理?(不需要写这个类的代码，回答出名字即可)

BigDecimal