## Day05

### 5-1 循环概念

* 循环：在java中做重复事情，并且重复的只编写一次
* 循环好处：减少代码的冗余性
* 常用的循环：
  + for循环、while循环、do while循环
* 循环特点：
  + 循环条件、循环操作(循环体)

### 5-2 for循环

* 语法格式：

**for(表达式1;表达式2;表达式3){**

**循环体**

**}**

* **在for循环中，表达式1,完成变量的初始值，如：int i = 1;**
* **在for循环中，表达式2完成循环条件判断，如果条件满足为true则执行循环体，否则执行循环外面的语句，如：i<=5**
* **在for循环中，循环体主要做重复事情，并且重复的事情在循环体重只编写一次**
  + **循环体中可以编写Java中任何语句**
  + **如：System.out.println("HelloWorld!");**
* **在for循环中表达式3完成变量值的更改，如：i++**
* **例如：输出5遍HelloWorld,使用for循环完成**

**for(int i = 1;i <= 5;i++){ //i = 6**

**System.out.println(“HelloWorld!”);**

**}**

**输出结果：**

**HelloWorld!**

**HelloWorld!**

**HelloWorld!**

**HelloWorld!**

**HelloWorld!**

### 5-3 双重循环

* 双重循环也叫做二重循环：在一个完整的循环中又包含另一个完整的循环

**for(;;){ //称为外层循环**

**for(;;){ //称为内层循环**

**}**

**}**

**结论：当外层循环执行一次，则内层循环执行一遍**

### 5-4 for循环注意事项

* 注意：1.在for循环中，三个表达式都可以省略，但是分号必须编写，则出现死循环也叫做无限循环，解决办法：按ctrl + c强行终止程序运行

for(;;){

System.out.println("OK");

}

* 注意：2.在for循环中，省略表达式1，则出现编译错误，解决办法：将表达式1编写在for循环上面

int i = 1;

for(;i <= 5;i++){

System.out.println(i);

}

* 注意：3.在for循环中，省略表达式2，则出现死循环或无限循环，也就是说当省略表达式2时，则条件默认为true

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

System.out.println(i);

}

* 注意：4.在for循环中，当省略表达式3，则出现死循环，解决办法：将表达式3编写在循环体中最后一条语句

for(int i = 1;i <= 5;){

System.out.println(i);

i++;

}

* 注意：5.当for循环中表达式1变量中的值在循环外进行使用，则出现编译错误，解决办法：将表达式1中声明的变量，编写在for循环外面进行声明

int i;

for(i = 1;i <= 5;i++){

System.out.println(i);

}

System.out.println(i); //6

### 5-5 while循环

* 语法格式：

while(循环条件){

循环体

}

* 执行流程：
  + 首先判断条件，当条件为true时，则执行循环体，然后再判断条件………一直到循环条件为false时，则循环结束

### 5-6 do while循环

* 特点：先执行，然后再判断
* 语法格式：

do{

循环体

}while(循环条件)**;**

* 执行流程
  + 先执行循环体，然后判断条件，当条件为true时，则继续执行循环体，然后再判断条件…….一直到循环条件为false时则循环结束

### 5-7 三种循环异同点

* 语法格式

for(表达式1;表达式2;表达式3){

循环体

}

while(循环条件){

循环体

}

do{

循环体

}while(循环条件);

* 执行顺序不同
  + for循环和while循环：先判断当条件为true时，然后再执行
  + do while循环：先执行循环体，然后再判断条件
* 使用情况不同
  + 当循环次数固定时，建议使用for循环
  + 当循环次数不固定时，建议使用while循环、do while循环
    - 先判断，再执行，则使用while循环
    - 先执行，然后再判断，则使用do while循环
* 当循环条件第一次为false时，则：
  + for循环执行0次循环体
  + while循环执行0次循环体
  + do while循环执行1次循环体

得出：

* + for循环：0次-----N次
  + while循环：0次----N次
  + do while循环：1次-----N次

### 5-8 在循环中使用break语句

* break表示中断，当在循环中遇到break则结束当前整个循环，执行循环外面的语句

### 5-9 在循环中使用continue语句

* continue; //表示继续，当遇到continue语句时，则结束当次循环继续执行下一次循环

### 5-10 break与continue区别

* 使用场合不同
  + break:可以在switch case中使用，也可以在循环中使用
  + continue：只能在循环中使用
* 作用不同
  + break:表示中断，当在switch case中或在循环中遇到break结束当前整个switch case或循环，执行外面的语句
  + continue:表示继续，当在循环中遇到continue，则结束当次(本次)循环继续执行下一次循环