

循环

回顾

今天任务

1. 循环的功能
2. 循环的分类
3. for循环
4. while循环
5. do/while 循环
6. 多层循环嵌套
7. 特殊的流程控制

教学目标

1. 掌握循环的功能
2. 掌握 for循环
3. 掌握 while循环
4. 掌握 do/while 循环
5. 掌握多层循环嵌套
6. 掌握特殊的流程控制

第一节：循环的功能

1.1 循环的作用

在循环条件满足的情况下，反复执行特定代码

1.2 为什么要使用循环

当我们要打印100次helloworld

或者我们想实现1-10的加和

1+2+3+4+5....

```
int sum = 0;
sum = sum + 1;
sum = sum + 2;
sum = sum + 3;
sum = sum + 4;
sum = sum + 5;
```

可以发现有一些是相同的内容。这些相同的内容我们就可以采用循环的方式来实现

1.3 循环的分类

1. for 循环
2. while 循环
3. do/while 循环

4.循环的组成部分

1. 初始化部分:对循环变量赋初值
2. 循环条件部分:判断循环变量是否超出某个界限
3. 循环体部分:要循环执行的具体逻辑.
4. 迭代部分:修改循环变量的值

第二节 : while循环

2.1 格式:

```
while (boolean表达式) {语句块}
```

2.2 执行过程

先判断表达式的值。若`=true`.则执行其后面的语句,然后再次判断条件并反复执行,直到条件不成立为止

2.3.1 练习一：需求: 打印输出5次helloworld

```
// 初始化部分
int count = 0;
// 2循环条件
while(count<5){// 1 2 3
// 3 循环体
    System.out.println("hello world");
    count++;
}
```

2.3.2 练习二：需求：打印输出 1--10

```
int i =1;
while(i<=10){
    System.out.println(i);
    i++;
}
```

2.3.3 练习三：需求：求 1--10 的阶乘

```
int sum = 1;
int j = 1;
while(j<=10){
    sum=sum*j;
    j++;
}
System.out.println("1-10的加和"+sum);
```

2.3.4 练习四：求 100以内的 偶数的和

```
int z = 2;
while(z<=100){
    System.out.println(z);
    z+=2;
}
```

第三节：do-while循环

3.1 格式

```
do {语句块}while(表达式) ;
```

3.2 执行过程

先执行语句,再判表达式的值,若为true,再执行语句,否则结束循环

3.3.1 练习：打印三次helloworld

```
// 1 初始化部分
int i = 0;
do{
    // 2 循环体
    System.out.println("Hello World!");
    // 4 迭代部分
    i++;
}while(i<3);// 3 循环条件
```

3.3.2 用do/while实现打印100以内的奇数

```

int j = 1;
do{
    /*if(j%2==1){
        System.out.println(j);
    }
    j++;*/
    System.out.println(j);
    j+=2;
}while(j<100);

```

3.3.3 100 以内 能够被3整除 但是不能被5整除的数 打印输出

```

int z = 3;
do{
    if(z%3==0 && z%5!=0){
        System.out.println(z);
    }
    z++;
}while(z<100);

```

3.4 while 和 do-while的区别

while 和 do/while 的区别:

while 先执行 循环条件 然后在执行 循环体

do/while 先执行 循环体 然后在执行 循环条件

当第一次 就不满足循环条件的情况下 while循环不能执行循环体, dowhile 可以执行一次

第四节：for循环

4.1 格式

```

for ([循环变量初始值设定]; [循环条件判断]; [改变循环变量的值]) {
    循环体
}

```

4.2 执行过程

首先计算表达式1,接着计算表达式2,若表达式2的值=true,则执行循环体,接着计算表达式3,再判断表达式2的值. 依此重复下去,直到表达式2的值=false。

4.3.1 练习： 需求: 打印输出3次helloworld

```
for(int i = 0;i<3;i++){
    System.out.println("Hello World!");
}
```

4.3.2 练习：打印100以内 能被4整除不到能被7整除的数据，每行打印6个

```
int count = 0;
for(int i = 1; i<=100; i++){
    if(i%4==0 && i%7!=0){
        System.out.print(i+"\t");
        count++; // 6
        if(count%6==0){
            System.out.print("\n");
        }
    }
}
```

4.4 for循环的特殊形式

1. 表达式2一般不可省略, 否则为无限循环

```
for (i=1; ; i++)    sum=sum+i;
// 相当于条件永真、永不为false
```

2. 表达式3亦可省略, 但在循环体中须有语句修改循环变量; 以使 表达式2在某一时刻为false而正常结束循环。

```
for (int sum=0,i=1 ;i<=100; ){
    sum=sum+i;
    i++;
}
```

3. 若同时省略表达式1**, **表达式3, 则相当于while(表达式2)语句

```
for ( ; i<=100; ) {sum+=i; i++;}
```

4. 三个表达式均省略 即for(;;)语句, 此时相当于while(true)语句。

4.5 几种循环的比较

1. 对于同一问题, 三种循环可相互替代
2. for循环功能强于while, do...while. 但若不是明显地给出循环变量初终值(或修改条件), 则可以用while 或do ... while. 以增强程序的结构化和可读性。
3. 要防止无限循环--死循环。

第五节：多循环嵌套

5.1练习:

```

// 外层循环控制行      内层循环 控制列
// *****
// *****
// *****
// *****

for (int j = 0;j<4 ; j++)
{
    for(int i = 0; i< 7 ; i++){
        System.out.print("*");
    }
    System.out.println();
}

```

5.2 练习二:

```

/*
找规律
*           1       1
**          2       2
***
****
*****      5       5

*/
for (int i = 1;i<=5 ;i++ )
{
    // 1 2 3 4 5
    for (int j = 1;j<=i ;j++ )
    {
        System.out.print("*");
    }
    System.out.println();
}

```

5.3 练习三:

```
// 99乘法表
for (int i = 1;i<=9 ;i++ )
{
    for (int j = 1;j<=i ;j++ )
    {
        System.out.print(i+"*"+ j+"="+i*j+"\t");
    }
    System.out.println();
}
```

5.4 练习四:

```
/*
2          2          5
2          2*10的0次方
22         2*10的1次方  + 2
222        2*10的2次方  + 22
2222       2*10的3次方  + 222
22222
*/
// 求 10 的2次方
int num5 = (int)(Math.pow(10,4))*2;
// 固定的 10
// 几次方  上一次的结果
// 要求 把所有的 数据求和
int result = 0;
int sum = 0;
for (int i = 0;i<5 ;i++ )
{
    int num = (int)(Math.pow(10,i))*2+result;
    sum = sum+num;
    System.out.println(num);
    result = num;
}
System.out.println("加和是"+sum);
```

第六节：特殊流程控制

6.1 break 默认跳出离自己最近的循环

```
out : for (int i = 1; i <3 ; i++ )
{
    inner : for (int j = 1;j<3 ;j++ )
    {
        if(j == 2){
            break out;// 可以指定 跳出的循环
        }
        System.out.println(i+" "+j);
    }
}
```

6.2 continue 跳过本次循环执行下一次循环

```
out : for (int i = 1;i<4 ;i++ )
{
    for (int j = 1;j<4 ;j++ )
    {
        if(j==2){
            continue out ;
        }
        System.out.println("i="+i + " j="+j);
    }
}
System.out.println("Hello World!");
```

第七节：总结

默写

一、if-else语句的语法格式

1. if(布尔表达式){

语句或语句块;

}

2. if(布尔表达式){

语句或语句块;

}else{

语句或语句块;

}

3. if(布尔表达式) {

语句或语句块;

} else if(布尔表达式){

语句或语句块;

}

.....

else {

语句或语句块;

}

4. 嵌套的if-else: 一个if-else语句块内包含一个或多个if-else语句块

5. 在if-else里必然能找到一条出路且只能找到一条出路。

二、switch

1. 语句格式:

switch(变量){

case 值1:

表达式1;

表达式2;

break;

case 值2:

表达式3;

break;

case 值3:

表达式4;

break;

default:

表达式5;

break;

}

2. switch语句的用法:

1) 根据变量的值, 来寻找case的值, 如果找到, 执行该case下的语句, 直到碰到break为止! 如果没有break, 则会顺序执行后面的语句。

2) 如果变量的值, 不与任意一条case的值相等, 则会执行default后的语句。default的位置是任意的, 并且是可有可无的。

3) 变量的类型, 可以是: char byte short int 枚举 String(jdk1.7)

4) case的值必须是确定的、固定的值(常量), 不能是取值范围。

if和switch的使用场景

1). 如果对具体的个数的数值进行判断, 用if可以, 用switch也可以, 建议用switch。

因为switch会把所有的备选答案加载进入内存当中, 选择的效率就会更高。

2). 如果要对数据的区间进行判断时, 用if语句。

3). 如果表达式的结果是boolean类型的, 毫无疑问用if语句

作业

- 1.求1至1000之间满足“用3除余2；用5除余3；用7除余2”的数，且一行只打印5个数
- 2.求 $1-3+5-7+\dots-99+101$ 的值
- 3.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
- 4.输入两个正整数m和n，求其最大公约数和最小公倍数
- 5.百元百鸡问题：公鸡5元一只，母鸡3元一只，3只小鸡1元，如果用100元钱，买100只鸡，不余不欠，可以买公鸡，母鸡，小鸡，各多少只。

6.编写一个Java应用程序，用循环结构打印如下的数值列表：

N	$10*N$	$100*N$	$1000*N$
1	10	100	1000
2	20	200	2000
3	30	300	3000
4	40	400	4000
5	50	500	5000

打印2到10000的所有素数，每行显示8个素数

8. 商品价格表

- (1)用户从控制台输入需要查询的商品编号，根据编号现实对应商品价格。
- (2)循环查询商品价格 （商品名称 单价）
- (3)输入n退出循环

9.开发一个标题为“FilpFlop”的游戏程序。它从1计数到100，遇到3的倍数就替换为单词Filp,5的倍数就替换为单词Flop,既为3的倍数又为5的倍数则替换单词FilpFlop.

面试题

1. 什么时候用for循环，什么时候用while循环
2. while循环和do-while循环的区别
3. break、continue、return的区别