

循环语句

回顾

选择结构：

if语句

(1) if(条件){...}

(2) if(条件){...}else{...}

(3) 多重if if(条件){...}else if(条件){...}else if(条件){...} else{...}

(4) 嵌套if if(条件){ if(){...}else{...} } else{...}

switch语句：多分支语句，只能做等值判断

```
switch(表达式){  
    case 常量1:  
        ...  
        break;  
    case 常量2:  
        ...  
        break;  
    case 常量3:  
        ...  
        break;  
    default:  
        ...  
        break;  
}
```

注意：表达式的类型：byte short int char String 枚举

常量值不能重复

default 可以不写

break:跳出switch

防止case穿透：写上break;

今天任务

1. 什么是循环
2. 循环的分类
3. while循环
4. do/while 循环
5. for循环
6. 多层循环嵌套
7. 跳转语句的使用

教学目标

1. 理解什么是循环
2. 掌握 while循环
3. 掌握 do/while 循环
4. 掌握 for循环
5. 掌握多层循环嵌套
6. 掌握跳转语句的使用

第一节：循环结构

1.1 什么是循环

循环就是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1-100的和?

```
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. while 循环
2. do/while 循环
3. for 循环

1.4 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. 更新循环变量部分:修改循环变量的值

第二节: while 循环

2.1 格式:

```
while (循环条件) {语句块(循环体)}
```

2.2 执行过程

先判断循环条件表达式的值。若为true,则执行循环体,然后再次判断条件并反复执行,直到条件不成立为止。
特点: 先判断再执行。

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");  
    //4更新循环变量  
    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:求1-100的和

```
//1初始化变量
int i=1;
int sum=0;//保存和
//2循环条件
while(i<=100){
    sum=sum+i;//sum+=i;
    i++;
}
System.out.println("1-100的和是:"+sum);
```

2.3.4 练习四: 需求:求 10 的阶乘

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

2.3.5 练习五: 求 100以内的 偶数的和

```
int z=2;
int sum=0;
while(z<=100){
    sum=sum+z;
    z+=2;
}
System.out.println("1-100的偶数的和是:"+sum);

或
int z=1;
int sum=0;
while(z<=100){
    if(z%2==0){
        sum=sum+z;
    }
    z++;
}
System.out.println("1-100的偶数的和是:"+sum);
```

第三节: do-while循环

3.1 格式

```
do {语句块} while(条件表达式) ;
```

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。

特点: 先执行, 再判断。

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 循环变量变化部分
    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:

1 执行过程不同

while 先执行循环条件, 然后再执行循环体, 一句话: 先判断, 再执行

do/while 先执行循环体 然后再执行循环条件, 一句话: 先执行, 再判断

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

第四节: for循环

4.1 格式

```
for (表达式1 [循环变量初始化]; 表达式2 [循环条件判断]; 表达式3 [更新循环变量]) {
    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的值为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的值为false。

特点: 先判断,再执行。

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");  
        }  
    }  
}
```

4.3.3 使用for实现1-100的和

```
public class Demo11{  
    public static void main(String[] args){  
        int sum=0;  
        for(int i=1;i<=100;i++){  
            sum+=i;  
        }  
        System.out.println("1-100的和是:"+sum);  
    }  
}
```

4.3.4 实现加法表

比如用户输入数字6,

0+6=6

1+5=6

2+4=6

...

6+0=6

```
import java.util.Scanner;
public class Demo12{
    public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        System.out.println("请输入一个数字");
        int num=input.nextInt();

        for(int i=0,j=num;i<=num;i++,j--){
            System.out.println(i+"+"+j+"="+i+j);
        }
    }
}
```

4.4 for循环的特殊形式

1. 省略表达式2

表达式2一般不可省略, 否则为无限循环(死循环)

```
for (int 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和表达式3, 则相当于while(表达式2)语句

```
int i=0;
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 三种循环的比较和注意事项

三种循环的比较:

1 语法不一样

```
while(条件){...}
do{...}while(条件);
for(表达式1;表达式2;表达式3){...}
```

2 执行顺序

while 和 for 都是先判断条件, 然后再执行循环体
do while 先执行循环体, 再判断条件

注意事项

1. 对于同一问题, 三种循环可相互替代。
2. 循环次数确定的情况优先选用for循环, 循环次数不确定的情况, 通常选用while和do-while循环。
3. 要防止无限循环——死循环。

第五节：多重循环：二重循环

多重循环就是循环中嵌套其他循环。

特点：外层循环执行一次，内层循环执行一遍。

5.1 练习一：

使用*号打印矩形。

```
// 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:

打印直角三角形

```
/*
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:

输出等腰三角形

```
请输入等腰三角形的行数: 5
*
***
*****
*****
*****
```

*	1	1	1*2-1
***	2	3	2*2-1
*****	3	5	3*2-1
*****	4	7	4*2-1
*****	5	9	5*2-1
*	1	4	5-1=4
***	2	3	5-2=3
*****	3	2	5-3=2
*****	4	1	5-4=1
*****	5	0	5-5=0

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

5.4 练习四:

输出九九乘法表

```
*
**
***
****
*****

1*1=1
1*2=2 2*2=4
1*3=3 2*3=6 3*3=9
```

```
// 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();
}
```

5.5 输入学员成绩

一个班级有5个人，3门课程，让你输入每个人每门课的成绩？并打印每个人的总分和平均分？

```
import java.util.Scanner;
public class Demo18{
    public static void main(String[] args){
        Scanner input=new Scanner(System.in);

        for(int i=0;i<5;i++){ //外层循环控制的人数
            System.out.println("请输入第"+(i+1)+"个人的姓名");
            String name=input.next();
            double sum=0;//存每个人总分
            double avg=0;//存每个人的平均分
            for(int j=0;j<3;j++){
                System.out.println("输入第"+(j+1)+"门课成绩");
                double score=input.nextDouble();
                sum+=score;
            }
        }
    }
}
```



```
        avg=sum/3;
        System.out.printf(name+"的总分是:"+sum+", 平均分是:%.2f", avg);
        System.out.println();
    }
}
}
```

第六节：跳转语句--流程控制语句

break: 语句用于终止某个语句块的执行

continue: 语句用于跳过某个循环语句块的一次执行,继续下一次执行（结束本次循环，继续下一次循环）

6.1 break

- 使用场合
 - switch结构中：跳出（终止）switch语句
 - 循环结构中：跳出（终止）循环
- 作用：退出switch和循环结构（如果有多重循环，默认跳出离自己最近的循环）。

```
for (int i = 1; i <3 ; i++ ){
    for (int j = 1;j<5 ;j++ ){
        if(j == 2){
            break;// 可以指定 跳出的循环
        }
        System.out.println(i+" "+j);
    }
}
```

使用Label标签实现跳出指定的循环。（了解）

```
out : for (int i = 1; i <3 ; i++ ){//定义一个标签out
    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);
    }
}
```

上机练习1:

打印1到10个数，遇到4的倍数程序自动退出

提示：如果i%4==0,则执行break命令

上机练习2:

要求：小张参加长跑比赛5000米，一共10圈，如果小张坚持不下来，可以中途退出比赛，使用程序描述这个场景？

```
import java.util.Scanner;
```

```
public class Demo21{
    public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        System.out.println("小张准备参加5000米长跑");
        int count=1;
        do{
            System.out.println("开始跑第"+count+"圈");
            System.out.println("跑完了第"+count+"圈");
            count++;
            System.out.println("还能坚持吗?y/n");
            String answer=input.next();
            if(!answer.equals("y")){
                break;
            }

        }while(count<=10);
        System.out.println("比赛结束了....");
    }
}
```

6.2 continue

- 使用场合
 - continue只能用在循环结构中
- 作用：跳过本次循环，执行下一次循环（如果有多重循环，默认继续执行离自己最近的循环）。）

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

使用Label标签改变继续执行的循环

```
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!");
```

上机练习1

打印1到10个数，遇到4的倍数跳过去。

```
/*
打印1到10个数，遇到4的倍数跳过去
*/
public class Demo23{
    public static void main(String[] args){
        for(int i=1;i<=10;i++){
            if(i%4==0){
                continue;
            }
            System.out.println(i);
        }
    }
}
```

上机练习2:

要求：小张参加长跑比赛5000米，一共10圈，如果小张口渴了，喝水后继续跑，使用程序描述这个场景？

```
/*
要求：小张参加长跑比赛5000米，一共10圈，如果口渴了，喝水后继续跑，使用程序这个场景？
*/
import java.util.Scanner;
public class Demo24{
    public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        System.out.println("小张准备参加5000米长跑");
        int count=1;
        do{
            System.out.println("开始跑第"+count+"圈");
            System.out.println("跑完了第"+count+"圈");
            count++;

            System.out.println("口渴了吗吗?y/n");
            String answer=input.next();
            if(answer.equals("n")){ //不渴 继续跑
                continue;
            }
            System.out.println("喝口水");

        }while(count<=10);
        System.out.println("比赛结束了....");
    }
}
```

总结

循环有四部分：1 循环变量初始化 2循环条件 3 循环体 4 循环变量更新

1 三种循环语句

while

先判断条件，再执行循环体

`do while`

先执行循环体，再判断条件，至少执行一次

`for`

先判断条件，再执行循环体

如果循环次数固定优先使用`for`，如果次数不确定用`while`和`do while`

2 二重循环 （难点）

外层循环执行一次，内存循环执行一遍。

3 跳转语句

`break` :跳出 `switch`和循环

`continue`: 结束本次循环，继续下一次循环

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

```
Scanner in=new Scanner(System.in);
int a = in.nextInt();
int b = in.nextInt();
int s=1;
int i;
for(i=2;i<=a&&i<=b;i++)
{
    if(a%i==0&&b%i==0)
    {
        s=i;
    }
}
System.out.println(a+"与"+b+"的最大公约数是"+s);
```

或

```
while(max%min!=0){
    int temp=max%min;
    max=min;
    min=temp;
}
```

min就是最大公约数

最小公倍数=两个数乘积/最大公约数

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。

公鸡x

母鸡y

小鸡 100-x-y

$$5x+y+3(100-x-y)/3=100$$

$$15x+9y+100-x-y=300$$

$$14x+8y=200$$

$$7x+4y=100;$$

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

7. 打印2到10000的所有素数（质数），每行显示8个素数

50 53

8. 商品价格表

(1) 用户从控制台输入需要查询的商品编号，根据编号显示对应商品价格。

(2) 循环查询商品价格 （商品名称 单价）

(3) 输入n退出循环

提示：使用do ... while循环，do while循环中嵌套switch判断商品编号。

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

break: 应用在switch和循环中，作用跳出（终止）语句块

continue: 应用在循环中，作用结束本次循环，继续下一次循环。

return : 用在方法中作用返回结果，结束方法。