# Java 循环结构

## 

## 一、什么是循环？

循环是一种编程技术，它允许程序在特定条件下重复执行一组语句。循环可以用来处理重复任务，例如遍历数组或字符串，或者执行重复的计算。循环是一种编程技术，它允许程序在特定条件下重复执行一组语句。循环可以用来处理重复任务，例如遍历数组或字符串，或者执行重复的计算。

## 二、无限循环

#### 1、什么是无限循环？

无限循环是指循环的条件始终为真，因此循环将永远执行下去，直到程序被强制终止。

#### 2、无限循环案例

while (true) {  
 System.out.println("This is an infinite loop");  
}

## 三、while 循环

#### 1、什么是while循环？

while循环是一种特殊的循环，它允许程序员在特定条件下重复执行一组语句。while循环通常用于在特定条件下执行重复任务，例如检查用户输入或执行重复的计算。

#### 2、while 循环基本用法：

while (条件表达式) {

// 循环语句

}

#### 3、案例：计算从1到100的和：（1 + 2 + 3 + 4 + … + 100 = ?）

* 初始化为0

* 循环条件是i <= 100

* 把i累加到sum中

* i自身加1

public static void main(String[] args) {  
 int sum = 0;   
 int i = 1;  
 while (i <= 100) {   
 sum = sum + i;   
 i++;   
 }  
 System.out.println("1加到100的和是：" + sum);  
 System.out.println("循环后变量i的值是：" + i);  
 }

## 四、do…while 循环

#### 1、什么是do-while循环？

do-while循环是一种特殊的循环，它允许程序员在特定条件下重复执行一组语句。do-while循环通常用于在特定条件下执行重复任务，例如检查用户输入或执行重复的计算，但是它会先执行一次循环体，然后再检查循环条件。

#### 2、do…while 循环基本用法：

do {

// 执行循环语句

} while (条件表达式);

#### 3、实例：计算从1到100的和：（1 + 2 + 3 + 4 + … + 100 = ?）

public static void main(String[] args) {  
 int sum = 0;  
 int i = 1;  
 do {  
 sum = sum + i;  
 i++;  
 } while (i <= 100);  
 System.out.println("1加到100的和是：" + sum);  
 System.out.println("循环后变量i的值是：" + i);  
 }

## 五、for 循环

#### 1、什么是for循环？

for循环是一种特殊的循环，它允许程序员在特定条件下重复执行一组语句。for循环通常用于遍历数组或字符串，或者执行重复的计算。

#### 2、for 循环基本用法：

for (初始条件; 循环检测条件; 更新) {

// 执行语句

}

#### 3、实例：计算从1到100的和：（1 + 2 + 3 + 4 + … + 100 = ?）

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

## 六、增强for循环

#### 1、什么是增强for循环？

增强for循环（也称为for-each循环）是一种特殊的循环，它可以用来遍历数组或集合中的元素。

#### 2、增强for循环的语法如下

for (元素类型 变量名 : 集合或数组) {   
 // 循环体   
}

#### 3、增强for循环案例

int[] numbers = {1, 2, 3, 4, 5};  
for (int num : numbers) {  
 System.out.println(num);  
}

## 七、嵌套循环

#### 1、什么是嵌套循环？

嵌套循环是一种特殊的循环，它允许程序员在特定条件下重复执行一组语句，其中一个循环体包含另一个循环体。嵌套循环通常用于处理复杂的重复任务，例如遍历多维数组或执行复杂的计算。

#### 2、嵌套循环的语法如下：

for (外部循环变量初始化; 外部循环条件; 外部循环变量更新) {  
 for (内部循环变量初始化; 内部循环条件; 内部循环变量更新) {  
 // 循环体  
 }  
}

#### 3、案例打印九九乘法表

for (int i = 1; i <= 9; i++) {  
 for (int j = 1; j <= i; j++) {  
 System.out.print(i + "\*" + j + "=" + i \* j + " ");  
 }  
 System.out.println();  
}

## 八、do-while循环、for循环、while循环的区别是什么？

初始化变量，for循环当中定义的初始化变量，只有自己才能用；while和do-while循环，初始化变量本来就在外面，所以外面也照样可以使用。

执行次数，for循环和while循环是【先判断后执行】，但是do-while循环是【先执行后判断】。

## 九、break与continue

#### 1、什么是break语句？

break语句是一种控制语句，它可以用来终止循环，跳出循环体，并执行循环之后的语句。break语句可以用于for循环，while循环，do-while循环和switch语句中。

#### 2、什么是continue语句？

continue语句是用于跳过当前循环中剩余的语句，然后继续进行下一次循环的语句。它可以用于while和for循环中，以结束本次循环，开始下一次循环。

#### 3、break 中断（结束循环）, continue 继续（结束本次循环，继续下一次循环）

for (int i = 1; i <= 5; i++) {  
 if (i == 3) {  
 break;   
 }  
 System.out.print(i);//12  
 }

for (int i = 1; i <= 5; i++) {  
 if (i == 3) {  
 continue;   
 }  
 System.out.print(i); //1245  
 }

## 十、标签循环

#### 1、什么是标签循环？

标签循环是一种特殊的循环，它允许程序员在特定条件下跳出多层循环，而不是只能跳出最内层的循环。标签循环使用标签来标记循环，并使用break语句和continue语句来控制循环的执行。

#### 2、标签循环语法

for (外部循环变量初始化; 外部循环条件; 外部循环变量更新) {  
 for (内部循环变量初始化; 内部循环条件; 内部循环变量更新) {  
 // 循环体  
 if (满足条件) {  
 break 标签;  
 }  
 }  
}

#### 3、标签循环案例

for (int i = 0; i < 10; i++) {  
 for (int j = 0; j < 10; j++) {  
 if (i + j == 5) {  
 break;  
 }  
 System.out.println("i = " + i + ", j = " + j);  
 }  
}