

Java 循环结构

一、什么是循环？

二、无限循环

1、什么是无限循环？

2、无限循环案例

三、while 循环

1、什么是while循环？

2、while 循环基本用法：

3、案例：计算从1到100的和： $(1 + 2 + 3 + 4 + \dots + 100 = ?)$

四、do...while 循环

1、什么是do-while循环？

2、do...while 循环基本用法：

3、实例：计算从1到100的和： $(1 + 2 + 3 + 4 + \dots + 100 = ?)$

五、for 循环

1、什么是for循环？

2、for 循环基本用法：

3、实例：计算从1到100的和： $(1 + 2 + 3 + 4 + \dots + 100 = ?)$

六、增强for循环

1、什么是增强for循环？

2、增强for循环的语法如下

3、增强for循环案例

七、嵌套循环

1、什么是嵌套循环？

2、嵌套循环的语法如下：

3、案例打印九九乘法表

八、do-while循环、for循环、while循环的区别是什么？

九、break与continue

1、什么是break语句？

2、什么是continue语句？

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

十、标签循环

- 1、什么是标签循环？
- 2、标签循环语法
- 3、标签循环案例

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

▼ Java 复制代码

```
1 while (true) {  
2     System.out.println("This is an infinite loop");  
3 }
```

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： $(1 + 2 + 3 + 4 + \dots + 100 = ?)$

- 初始化为0
- 循环条件是 $i \leq 100$
- 把i累加到sum中
- i自身加1

```
Java | 复制代码  
  
1 public static void main(String[] args) {  
2     int sum = 0;  
3     int i = 1;  
4     while (i <= 100) {  
5         sum = sum + i;  
6         i++;  
7     }  
8     System.out.println("1加到100的和是: " + sum);  
9     System.out.println("循环后变量i的值是: " + i);  
10 }
```

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： $(1 + 2 + 3 + 4 + \dots + 100 = ?)$

```
1 public static void main(String[] args) {
2     int sum = 0;
3     int i = 1;
4     do {
5         sum = sum + i;
6         i++;
7     } while (i <= 100);
8     System.out.println("1加到100的和是: " + sum);
9     System.out.println("循环后变量i的值是: " + i);
10 }
```

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 = ?）

```
1 public static void main(String[] args) {
2     int sum = 0;
3     for (int i = 1; i <= 100; i++) {
4         sum = sum + i;
5     }
6     System.out.println("1加到100的和是: " + sum);
7 }
```

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

▼ Java [复制代码](#)

```
1 ▼ for（元素类型 变量名：集合或数组）{  
2     // 循环体  
3 }
```

3、增强for循环案例

▼ Java [复制代码](#)

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

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：

▼ Java [复制代码](#)

```
1 ▼ for（外部循环变量初始化；外部循环条件；外部循环变量更新）{  
2     ▼ for（内部循环变量初始化；内部循环条件；内部循环变量更新）{  
3         // 循环体  
4     }  
5 }
```

3、案例打印九九乘法表

```
1 for (int i = 1; i <= 9; i++) {  
2     for (int j = 1; j <= i; j++) {  
3         System.out.print(i + "*" + j + "=" + i * j + " ");  
4     }  
5     System.out.println();  
6 }
```

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）

```
1 for (int i = 1; i <= 5; i++) {  
2     if (i == 3) {  
3         break;  
4     }  
5     System.out.print(i);//12  
6 }
```

```
1 for (int i = 1; i <= 5; i++) {  
2     if (i == 3) {  
3         continue;  
4     }  
5     System.out.print(i); //1245  
6 }
```

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

```
1 for (外部循环变量初始化; 外部循环条件; 外部循环变量更新) {  
2     for (内部循环变量初始化; 内部循环条件; 内部循环变量更新) {  
3         // 循环体  
4         if (满足条件) {  
5             break 标签;  
6         }  
7     }  
8 }
```

3、标签循环案例

```
1 for (int i = 0; i < 10; i++) {  
2     for (int j = 0; j < 10; j++) {  
3         if (i + j == 5) {  
4             break;  
5         }  
6         System.out.println("i = " + i + ", j = " + j);  
7     }  
8 }
```

