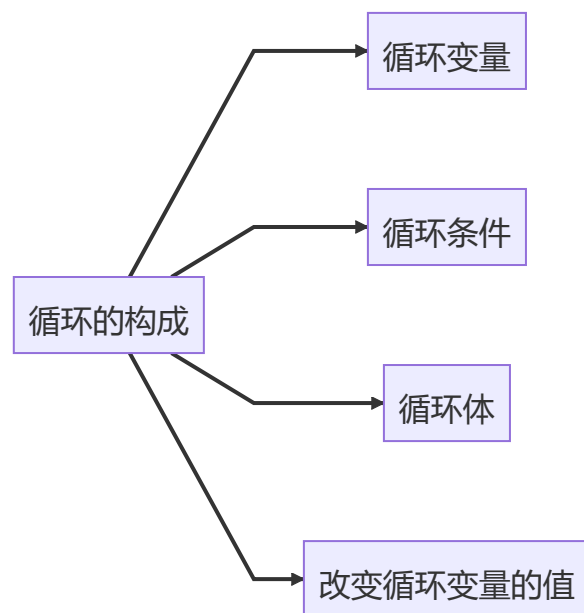


# Java基础(4)<sup>1</sup>

## ✧ 循环结构



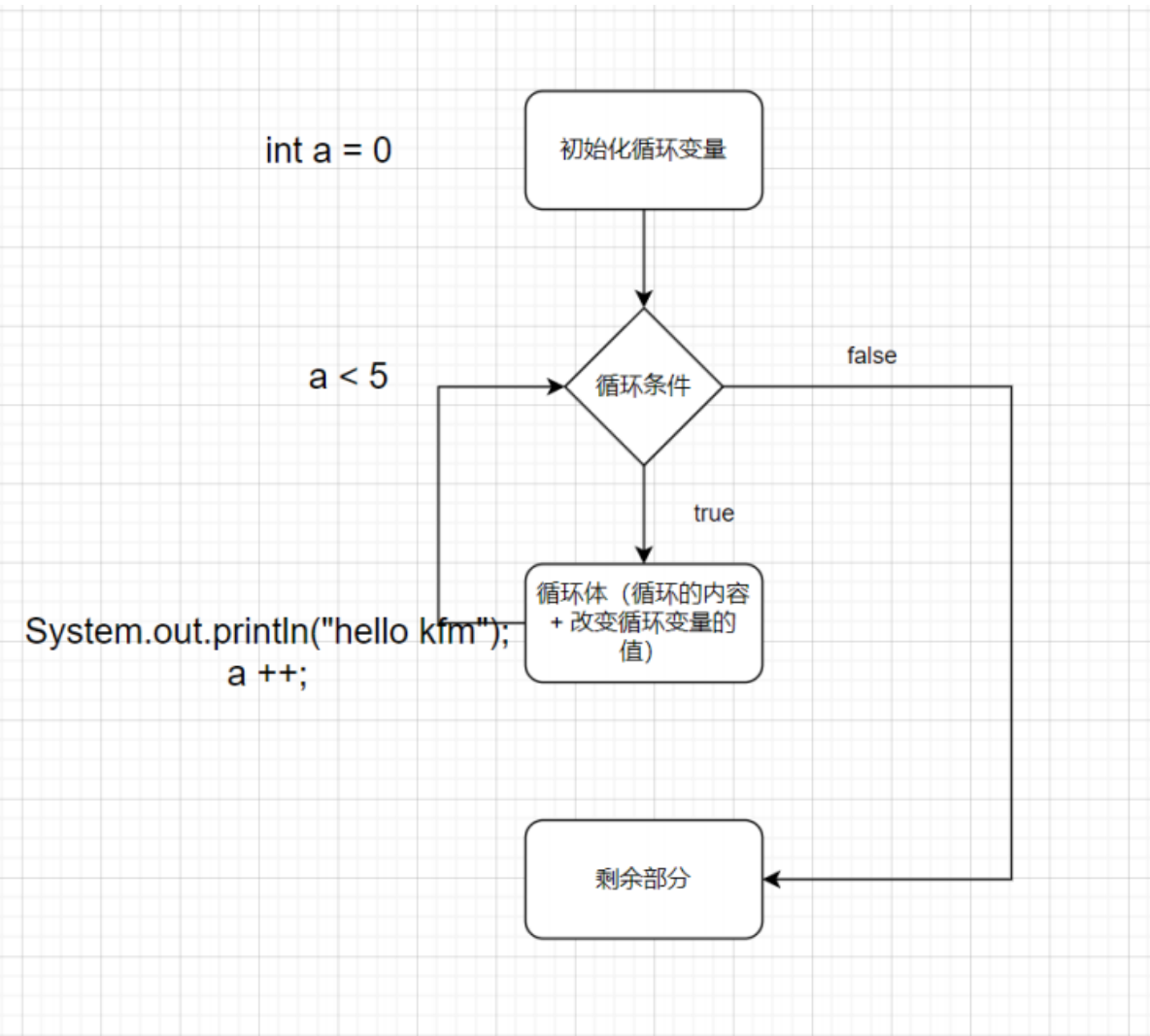
- 循环变量，循环变量用于控制循环次数。
- 循环条件，循环条件用来判断循环是否继续。
- 循环体，循环体是循环条件为true时，要执行的代码段。
- 改变循环变量的值，只有循环变量的值能够改变，循环才有结束的时候，否则是死循环。

### while循环——格式和流程

```
1 // while循环格式:
2 初始化语句;
3 while (循环条件){
4     循环体语句(被重复执行的代码);
5     迭代语句;
6 }
```

**while**循环流程

```
1 // 需求: 打印5行Hello kfm
2 int a = 0;
3 while (a < 5) { // a = 0 a = 1 a = 2 a = 3 a = 4 a = 5
4     System.out.println("hello kfm");
5     a ++;
6 }
```



## for循环——格式和流程

```

1  for (表达式1; 表达式2; 表达式3) {
2      循环体
3  }
4  // 表达式1 : 循环变量的初始化
5  // 表达式2 : 循环条件
6  // 表达式3 : 改变循环变量的值
7  // 循环体 : 要重复循环的部分
8  表达式1 → 表达式2 → 循环体 → 表达式3 → 表达式2 → 循环体 → 表
   达式3 → 表达式2
9  // for循环格式:
10 for (初始化语句; 循环条件; 迭代语句) {
11     循环体语句(重复执行的代码);
12 }
```

### for循环流程

```

1  // 需求: 打印3行Hello World
2  for (int i = 0; i < 3; i ++ ) { // i = 0 i = 1 i = 2 i = 3
3      System.out.println("hello world");
4  }
```

## do-while循环——格式和流程

```

1  // do-while循环格式:
2  初始化语句;
3  do {
4      循环体语句;
5      迭代语句;
6  }while(循环条件);
```

### do-while循环流程

```

1  int i =0;
2  do {
3      System.out.println("hello world");
4      i++;
5  }while(i < 3);
```

- ① for 循环和 while 循环（先判断后执行）；  
do...while（先执行后判断）
- ② for 循环和while循环的执行流程是一模一样的，  
功能上无区别，for 能做的while也能做，反之亦然。

如果已知循环次数建议使用 `for` 循环，如果不清楚要循环多少次建议使用 `while` 循环。

- ③ `for` 循环中控制循环的变量只在循环中使用
- `while` 循环中，控制循环的变量在循环后还可以继续使用

## 死循环

```

1 // for死循环
2 for ( ; ; ){
3     System.out.println("Hello World1");
4 }
5 // while死循环
6 while (true) {
7     System.out.println("Hello World2");
8 }
9 // do-while死循环
10 do {
11     System.out.println("Hello World3");
12 }while (true);

```

死循环有什么应用场景呢？

i

最典型的是可以用死循环来做服务器程序，比如百度的服务器程序就是一直在执行的，你随时都可以通过浏览器去访问百度。如果哪一天百度的服务器停止了运行，也就意味着所有的人都无法使用百度提供的服务了。

## 循环嵌套

所谓循环嵌套，就是一个循环中又包含另一个循环（就是同学们常说的，套娃）。

```

1 for (int i = 0; i < 3; i ++ ) { // i = 0 i = 1 i = 2
2     for (int j = 0; j < 5; j ++ ) { // j = 0 j = 1 j = 2 j = 3
3         System.out.println("我爱你");
4     }
5 }
6 // 外循环次数 * 内循环次数

```

循环嵌套执行流程：外部循环每循环一次，内部循环会全部执行完一轮

## 跳转语句

### 标签

标签可以放在for、while或do语句之前，其语法格式为：

标签：语句；

### break、continue

这里就需要用到跳转语句，需要用到break和continue两个关键字。

- break 作用：跳出并结束当前所在循环的执行
- continue 作用：结束本次循环，进入下一次循环

需要注意的是break和continue不是任何地方都可以使用的。



break：只能用于结束所在循环，或者结束所在switch分支的执行。

continue：只能在循环中进行使用。

## 循环结构总结

### 1 什么是流程控制

答：流程控制是用来控制程序的执行顺序的

### 2 分支结构if和switch，如何选择？

答：if 分支：一般用于对一个范围进行判断；switch 分支：对一个一个值进行匹配

### 3 for 循环和 while 循环、do-while 如何循环

答：知道循环次数用for、不知道循环次数用while；想要先执行，再判断，用do-while

## ✧ 生成随机数

```
1 // 目标：掌握使用Random生成随机数的步骤。
2 // 1、导包。import java.util.Random; (idea会自动完成)
3 import java.util.Random;
4 public class RandomDemo1 {
5     public static void main(String[] args) {
6         // 2、创建一个Random对象，用于生成随机数。
7         Random r = new Random();
8         // 3、调用Random提供的功能：nextInt得到随机数。
```

```
9      for (int i = 1; i ≤ 20; i++) {  
10          // 里面给一个数字n就会生成 (0~n-1) 之间的随机数  
11          int data = r.nextInt(10); // [0 - 9]  
12          System.out.println(data);  
13      }  
14  }  
15 }
```

---

1. 学习Java第四天 ↩