

第4章 演算子と分岐文

Java 循环[繰り返し文]

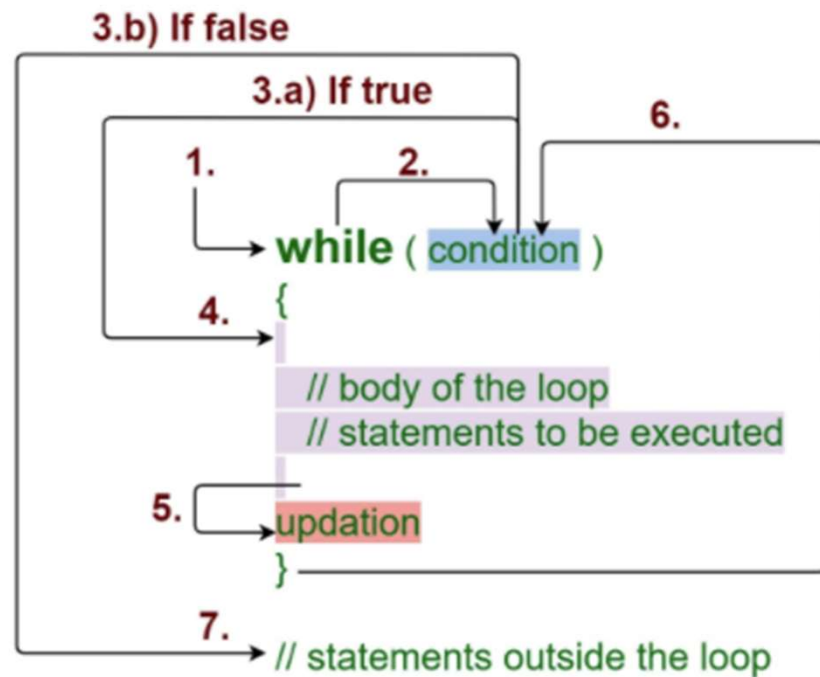
- 只要达到指定条件，循环就可以执行代码块。
- 循环很方便，因为它们可以节省时间，减少错误并且使代码更具可读性。
- Java有这几种循环语句：
 - while
 - do while
 - for
 - for each

Java while循环

- 先看句法:

```
while (condition) {
    // code block to be executed
}
```

- 流程图:



Java while循环

- 尝试代码: While.java
- 思考: 以下代码会怎么样?

```
int i = 0;  
while (true) {  
    System.out.println(i);  
    i++;  
}
```

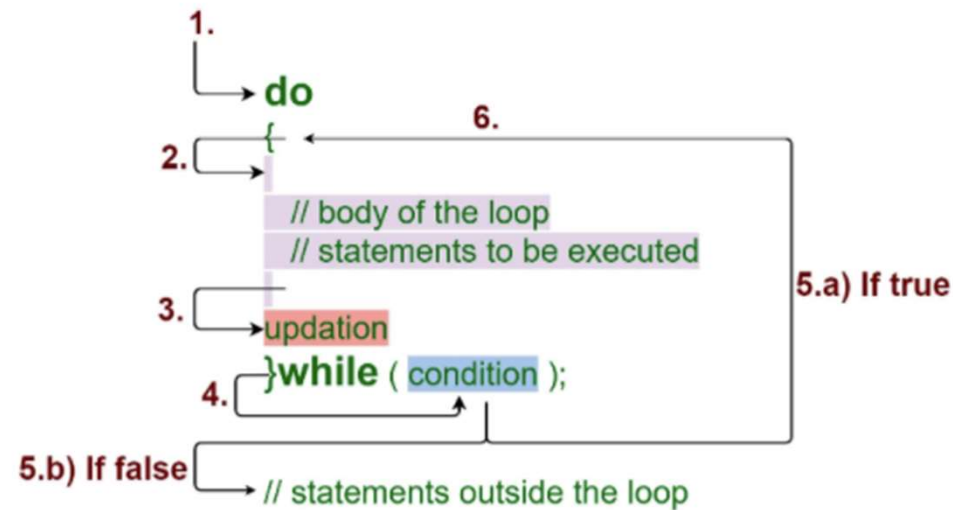
- 不要忘记增加条件中使用的变量, 否则循环将永远不会结束!

do while循环

•先看句法:

```
do {
    // code block to be executed
}
while (condition);
```

•流程图:



do while循环

- 尝试代码: `DoWhile.java`
- 不要忘记增加条件中使用的变量，否则循环将永远不会结束!
- do while循环是while循环的变体。在检查条件是否为真之前，该循环将执行一次代码块，然后只要条件为真，它将重复循环。

```
int i = 0;  
do {  
    System.out.println(i);  
    i++;  
}  
while (i < 5);
```

for 循环[for ループ]

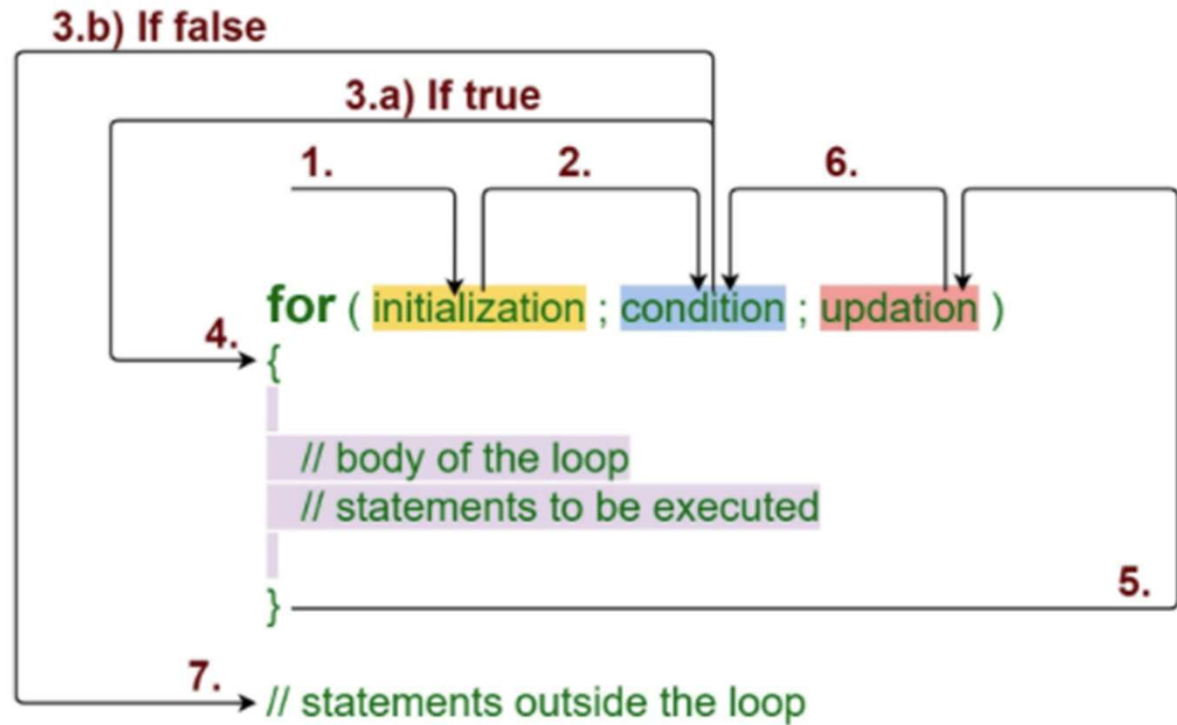
- 先看句法:

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

- statement1在执行代码块{}之前（一次）执行。
- statement2定义了执行代码块的条件。
- statement3在每次执行代码块后都会执行（每次）。

for 循环[for ループ]

•流程图:



•尝试代码: For.java

for-each 循环[for-each ループ]

- 先看句法:

```
for (type variableName : arrayName) {  
    // code block to be executed  
}
```

- 专用于循环遍历数组[配列]中的元素（数组之后会讲，可以回忆一下JS的数组）

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
for (String i : cars) {  
    System.out.println(i);  
}
```

- 尝试代码: Foreach.java

Java break

- 前面出现的break用于跳出switch语句
- break语句还可用于跳出任何循环。
- 当 i 等于4时，此示例将跳出循环：

```
for (int i = 0; i < 10; i++) {  
    if (i == 4) {  
        break;  
    }  
    System.out.println(i);  
}
```

- break只会跳出它所在的那一层循环

Java continue

- 如果出现指定条件，continue语句将中断一次迭代（在循环中），并在循环中继续进行下一次迭代。
- 本示例跳过值4：

```
for (int i = 0; i < 10; i++) {  
    if (i == 4) {  
        continue;  
    }  
    System.out.println(i);  
}
```

练习时间

- 1. 以下代码会输出什么内容？

```
int i = 0;
while (i < 10) {
    System.out.println(i);
    i++;
    if (i == 4) {
        break;
    }
}
```

- 2. 以下代码会输出什么内容？

```
int i = 0;
while (i < 10) {
    if (i == 4) {
        i++;
        continue;
    }
    System.out.println(i);
    i++;
}
```

遍历[走查]数组

- 可以通过for循环遍历数组元素，并使用length(数组长度)属性指定循环应运行的次数：

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (int i = 0; i < cars.length; i++) {
    System.out.println(cars[i]);
}
```

- 还有一个“for-each”循环，专用于循环遍历数组中的元素：

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (String i : cars) {
    System.out.println(i);
}
```


练习时间

- 1.使用while循环遍历一遍上一页的数组cars
- 2.思考：什么时候用while，什么时候用for等循环比较合适？为什么？

多维数组

- 多维数组是包含一个或多个数组的数组。
- 要创建一个二维数组，请将每个数组添加到其各自的花括号内：

```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
```

- 要访问myNumbers数组的元素，请指定两个索引：一个用于内数组，一个用于内数组内的元素。此示例访问myNumbers的第二个数组（1）中的第三个元素（2）：

```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
int x = myNumbers[1][2];
System.out.println(x); // Outputs 7
```

- 尝试代码：MultiArray.java

Q&A

You Have
Questions
We Have
Answers

THANK YOU