

## حلقه ی while و do ... while در جاوا

حلقه در برنامه نویسی برای تکرار یک بلوک خاص از کد استفاده می شود. در این آموزش ، یاد می گیرید که از حلقه ی while و do ... while در برنامه نویسی جاوا استفاده کنید.

تکرار یک حلقه تا برقراری یک شرط خاص ادامه پیدا می کند (شرط نادرست شود).

حلقه ها همان چیزی هستند که کامپیوترها را تبدیل به ماشین های جالب می کنند. فرض کنید که باید ۵۰ بار یک جمله را روی صفحه نمایش چاپ کنید. می توانید این کار را با استفاده از دستور چاپ ۵۰ بار (بدون استفاده از حلقه) انجام دهید. چگونه می خواهید یک میلیون بار یک جمله را چاپ کنید؟ پس باید از حلقه ها استفاده کنید.

این فقط یک مثال بسیار ساده است. در اینجا یاد می گیرید که از حلقه های while و do ... while برای نوشتن برنامه های جالب استفاده کنید.

### حلقه ی while در جاوا

ساختار حلقه ی while به شکل زیر است:

```
while (testExpression) {  
// codes inside body of while loop  
}
```

### حلقه while چگونه کار می کند؟

شرط درون پرانتز یک عبارت از نوع boolean است.

اگر شرط به صورت صحیح (true) ارزیابی شود ،

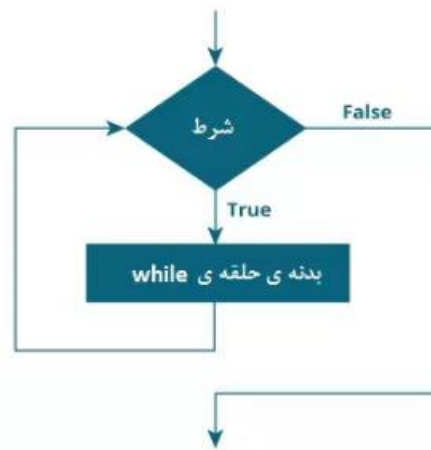
- عبارات داخل بدنه ی حلقه اجرا می شوند.
- سپس ، شرط دوباره ارزیابی می شود.

این روند تا زمانی که شرط به غلط ارزیابی شود ادامه می یابد.

اگر شرط به غلط ارزیابی شود ،

- حلقه خاتمه می یابد.

## فلوچارت حلقه ی while



### مثال ۱ : حلقه ی while

```
1. // Program to print line 10 times
2. class Loop {
3.     public static void main(String[] args) {
4.
5.         int i = 1;
6.
7.         while (i <= 10) {
8.             System.out.println("Line " + i);
9.             ++i;
10.        }
11.    }
12. }
```

خروجی

Line 1  
Line 2  
Line 3  
Line 4  
Line 5  
Line 6  
Line 7

Line 8

Line 9

Line 10

نکته  $i$  : در داخل حلقه بعد از ۱۰ بار تکرار، برابر با ۱۱ خواهد شد. سپس، شرط  $i \leq 10$  غلط ارزیابی می شود و حلقه `while` خاتمه می یابد.

## مثال ۲: حلقه `while`

```
1. // Program to find the sum of natural numbers from 1 to 100.
2. class AssignmentOperator {
3.     public static void main(String[] args) {
4.
5.         int sum = 0, i = 100;
6.
7.         while (i != 0) {
8.             sum += i; // sum = sum + i;
9.             -i;
10.        }
11.
12.        System.out.println("Sum = " + sum);
13.    }
14. }
```

خروجی

Sum = 5050

در اینجا، مقدار متغیر `sum` برابر ۰ و مقدار اولیه  $i$  برابر ۱۰۰ است. در هر تکرار از حلقه، مقدار متغیر به `sum` با  $i$  جمع می شود ( $sum+i$ )، و مقدار  $i$  یک واحد کاهش می یابد تا زمانی که  $i$  برابر با ۰ شود. برای تجسم بهتر،

1st iteration:  $sum = 0+100 = 100, i = 99$

2nd iteration:  $sum = 100+99 = 199, i = 98$

3rd iteration:  $sum = 199+98 = 297, i = 97$

... ..

... ..

99th iteration:  $sum = 5047+2 = 5049, i = 1$

100th iteration:  $sum = 5049+1 = 5050, i = 0$

## حلقه `do ... while` در جاوا

حلقه `do ... while` شبیه به حلقه `while` است با یک تفاوت کلیدی. بدنه ی حلقه `do ... while` برای یک بار اجرا می شود قبل از اینکه شرط بررسی شود.

## ساختار حلقه: do ... while

```
do {  
// codes inside body of do while loop  
} while (testExpression);
```

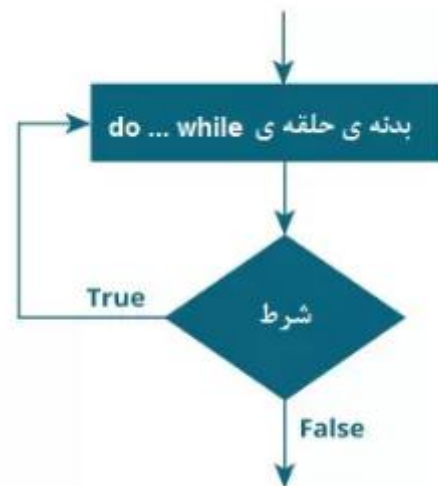
### حلقه do ... while چگونه کار می کند؟

بدنه ی حلقه do ... while یک بار اجرا می شود (قبل از بررسی شرط). در سری های بعد شرط بررسی می شود.

اگر شرط صحیح ارزیابی شود ، کدهای داخل بدنه ی حلقه اجرا می شوند ، و شرط دوباره ارزیابی می شود. این روند تا زمانی که شرط غلط ارزیابی شود ادامه می یابد.

وقتی شرط نادرست شود ، حلقه do ... while خاتمه می یابد.

### فلوچارت حلقه ی do ... while



### مثال ۳: حلقه ی do ... while

برنامه زیر تعداد اعدادی را که کاربر تا زمان ورود ۰ وارد می کند ، محاسبه می کند.

برای گرفتن ورودی از کاربر ، از شی Scanner استفاده کرده ایم.

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

```

2. class Sum {
3. public static void main(String[] args) {
4.
5.     Double number, sum = 0.0;
6.     Scanner input = new Scanner(System.in);
7.     do {
8.         System.out.print("Enter a number: ");
9.         number = input.nextDouble();
10.        sum += number;
11.    } while (number != 0.0);
12.
13.    System.out.println("Sum = " + sum);
14. }
15. }

```

خروجی

```

Enter a number: 2.5
Enter a number: 23.3
Enter a number: -4.2
Enter a number: 3.4
Enter a number: 0
Sum = 25.0

```

## حلقه while بی نهایت

اگر شرط هرگز غلط نشود، بدنه ی حلقه تا بی نهایت انجام می شود. در حال حاضر حلقه while بارها اجرا می شود (حداقل در تئوری). مثلاً

```

1. while (true) {
2.     // body of while loop
3. }

```

مثال دیگر:

```

1. int i = 100;
2. while (i == 100) {
3.     System.out.print("Hey!");
4. }

```