УВОД В ПРОГРАМИРАНЕТО
ТЕМА 6

Цикли в Java

Съдържание:

В настоящата тема ще разгледаме конструкциите за цикли, с които можем да изпълняваме даден фрагмент програмен код многократно. Ще разгледаме как се реализират повторения с условие (while и do-while цикли) и как се работи с for-цикли. Ще дадем примери за различните възможности за дефиниране на цикъл, за начина им на конструиране и за някои от основните им приложения. Накрая ще разгледаме как можем да използваме няколко цикъла един в друг (вложени цикли).

Автор: Румяна Топалска
Изготвил: Георги Кирчев

УВОД В ПРОГРАМИРАНЕТО
ТЕМА 6
Цикли в Java

Що е то цикъл?

В програмирането често се налага многократното изпълнение на дадена последователност от операции. Цикълът (loop) е структурата, която позволява това изпълнение без излишно писане на повтарящ се код. В зависимост от вида на цикъла, програмния код в него се повтаря:
- определени от фиксирано число пъти;
- докато дадено условие е изпълнено.

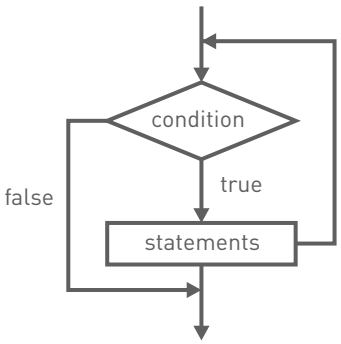
Цикъл, който никога не свършва, се нарича безкраен (infinite).

Цикъл while

Конструкция за цикъл while

Един от най-простите и често използвани цикли е while.

```
while (condition) {
  statements;
}
```



Condition е израз, който връща булев резултат – true или fasle. Той определя докога ще се изпълнява тялото на цикъла и се нарича – loop condition.
Statements са програмният код, изпълняван в цикъла. Те се наричат тяло на цикъла. При while цикъла първо се изпълнява булевия израз, ако резултатът от него е true, се изпълнява и последователността от операции, това се повтаря докато условния израз не върне false. Това е и причината често да бъде наричан цикъл с предусловие (pre-test loop).

Цикли в Java

Използване на while цикли

While циклите изпълняват група операции докато е в сила дадено условие.
Нека разгледаме един съвсем прост пример за използването на while цикъл, при който само се отпечатват на конзолата числата в интервала от 0 до 9 в нарастващ ред:

```
// Initialize the counter
int counter = 0;
// Check the loop condition
while (counter < 10) {
        // Execute statements in loop if the result is true
        System.out.printf("Number : %d%n", counter);
        // Change the counter
        counter++;
}
```

При изпълнение на примерния код получаваме следния резултат:

```
Number : 0
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
Number : 6
Number : 7
Number : 8
Number : 9
```

Примери за цикъл while

Сумиране на числата от 1 до N – пример

В настоящия пример ще разгледаме как с помощта на цикъла while се намира сумата на числата от 1 до N. Числото N се чете от конзолата. Инициализираме променливите num и sum със стойност 1. В num ще пазим текущото число, което ще добавяме към сумата на предходните. При всяко преминаване през цикъла ще го увеличаваме с 1, за да получим следващото число, след което при влизане в цикъла проверяваме дали то отговаря на условието на цикъла, тоест дали е в интервала от 1 до N. Sum е променливата за сумата на числата. При влизане в цикъла добавяме към нея поредното число записано в num. На конзолата принтираме всички num (числа от 1 до N) с разделител "+" и крайния резултат от сумирането след приключване на цикъла.

```
Scanner input = new Scanner(System.in);
System.out.print("n = ");
int n = input.nextInt();
int num = 1;
int sum = 1;
System.out.print("The sum 1");
while (num < n) {
        num++;
        sum += num;
        System.out.printf("+%d", num);
}
System.out.printf(" = %d%n", sum);
```

Изходът на програмата е следният:

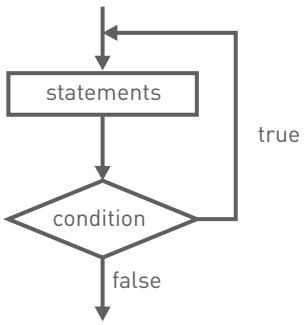
```
n=17
The sum = 1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17 = 153
```

Цикъл do while

Конструкция за цикъл do while

Do-while цикълът е аналогичен на while цикъла, само че при него проверката на булевия израз се прави след изпълнението на операциите в цикъла. Този тип цикли се наричат – цикли с условие в края (post-test loop).
Ето как изглежда един do-while цикъл:

```
do {
  statements;
}
while (expression);
```



Използване на do while цикли

Do-while цикълът се използва, когато искаме да си гарантираме, че поредицата от операции в него ще бъде многократно, но май-малко веднъж.

Примери за цикъл do while

Изчисляване на факториел

В този пример отново ще изчислим факториела на дадено число n, но този път вместо безкраен while цикъл, ще използваме do-while. Логиката е аналогична на тази в предния пример. Умножаваме всяко следващо число с произведението на предходните и го намаляваме с 1, след което проверяваме дали то все още е по-голямо от 0. Накрая отпечатваме получения резултат на конзолата.

```
Scanner input = new Scanner(System.in);
System.out.print("n = ");
int n = input.nextInt();
long factorial = 1;
do {
        // Multiply to become next value of factorial
        factorial *= n;
        // Decrement n to get next number
        n--;
} while (n > 0); // Check the loop condition
System.out.println("n! = " + factorial);
```

Ето го и резултатът от изпълнение на горния пример при n=7:

```
n = 7
n! = 5040
```

Цикъл for

Конструкция за цикъл for

For-циклите са малко по-сложни от while и do-while циклите, но за сметка на това могат да решават по-сложно задачи с по-малко писане на код. Характерната за for-цикъла структура е следната:

```
for (initialization; test; update) {
        statements;
}
```

Тя се състои от инициализационна част за брояча, булево условие, израз за обновяване на брояча и тяло на цикъла. Броячът на for цикъла го отличава от другите цикли. Броят на итерациите на този цикъл най-често се знае още преди да започне изпълнението му.
Безкраен цикъл (infinite loop) чрез оператора for се конструира по следния начин:

```
for (   ;   ;   ) {
        statements;
}
```

Безкраен цикъл означава цикъл, който никога не завършва. Обикновено в нашите програми нямаме нужда от безкраен цикъл, освен, ако някъде в тялото му не използваме break, за да завършим цикъла преждевременно.

Инициализация на цикъл for

For-циклите могат да имат инициализационен блок:

```
for (int num = 0; ...; ...) {
    // Can use num here
}
// Cannot use num here
```

Той се изпълнява веднъж, точно преди влизане в цикъла. Обикновено се използва за деклариране на променливата-брояч. Тази променлива е "видима" и може да се използва само в рамките на цикъла.

Условия на цикъл for

For-циклите могат да имат условие за повторение:

```
for (int num = 0; num < 10; ...) {
    // Can use num here
}
// Cannot use num here
```

То се изпълнява веднъж, преди всяка итерация на цикъла. При резултат true се изпълнява тялото на цикъла, а при false то се пропуска и се преминава към останалата част от програмата. Използва се като loop condition (условие на цикъла).

Примери за цикъл for

Изчисляване на N^M – пример

Ще напишем програма, която пресмята m-тата степен на число n. Ще използваме for-цикъл. Инициализираме променливата-брояч (int i = 0). Определяме условието на цикъла – i < m, така цикълът се изпълнява от 0 до m-1 или точно m пъти. При всяко изпълнение на цикъла n ще се вдига на поредната степен и накрая ще принтираме резултата, за да видим правилно ли работи програмата.

```
Scanner input = new Scanner(System.in);
System.out.print("n=");
int n = input.nextInt();
System.out.print("m=");
int m = input.nextInt();
long result = 1;
for (int i = 0; i < m; i++) {
        result *= n;
}
System.out.println("n^m = " + result);
```

Ето как изглежда изходът от програмата при n=2 и m=10:

```
n=2
m=10
n^m=1024
```

Задачи за домашна работа

Напишете програма, която отпечатва на конзолата числата от 1 до N. Числото N се чете от стандартния вход.