



Увод в програмирането с Java



Цикли (II част)

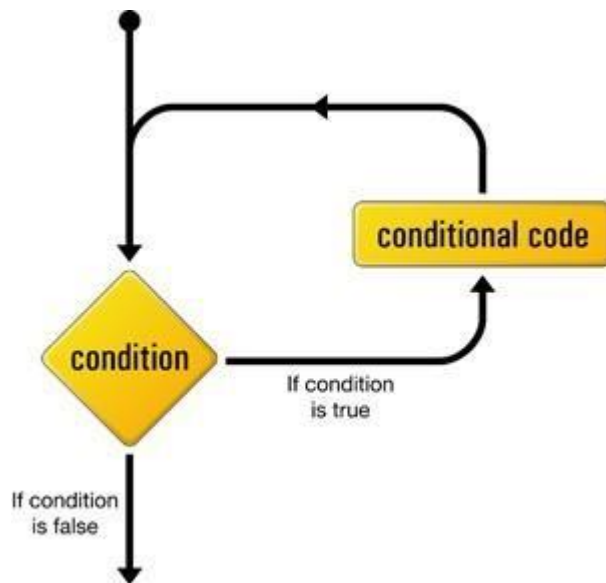
Съдържание

- Преговор
 - while
 - do-while
 - Цикъл for
 - Вложени цикли
- Домашно

ПРЕГОВОР

Цикли

Позволяват изпълнението на код много пъти, докато някакво условие е вярно.

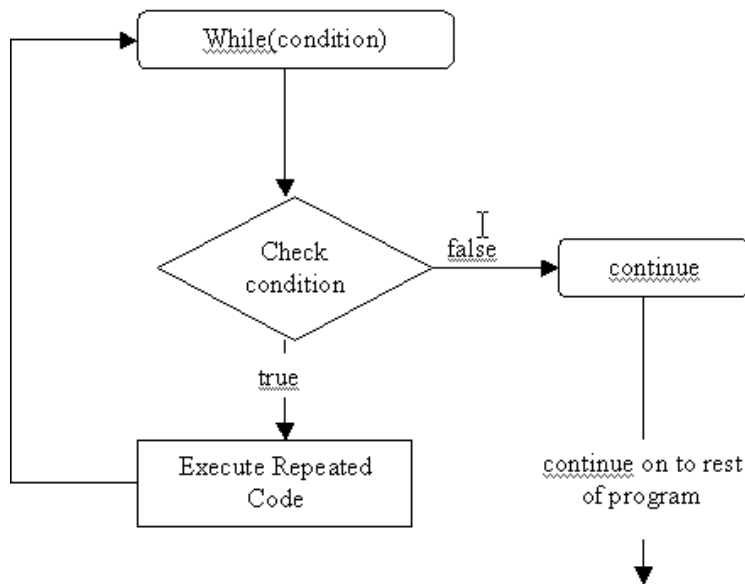


Цикъл while

Изпълнява се докато условието в скобите е вярно. Трябва така да се измисли цикъла, че условието да стане грешно по някое време, защото иначе програмата ще е безкрайна.

```
while (условие) {  
    //код, който се повтаря  
}
```

Flow Diagram of a while loop



Цикъл do-while

Конструкцията do-while е аналогична на while, като разликата е, че условието се оценява след изпълнението на операциите в цикъла (гарантираме най-малко едно изпълнение):

```
do {  
  
statement1;  
  
statement2;  
  
...  
}  
while (Boolean expression);
```

Задача

Суиурайте числата от 1 до 100

Цикъл For

Цикъл For

Използва се когато знаем колко точно завъртания ще има цикъла.

```
for (int i = 0; i < 10; i++) {  
    System.out.println(i);  
}  
for (int i = 10; i > 0; i--) {  
    System.out.println(i);  
}
```

i е брояча, който се променя с всяко завъртане на цикъла. Трябва да му зададем първоначална стойност, какво е условието при което цикъла ще спре и също така - как брояча ще се променя. Променливите, които се инициализират в рамките на цикъла не съществуват извън него.

Пример

Сумирайте числата от 1 до n:

```
int n = 20;
```

```
int sum = 0;
```

```
for (int i = 1; i <= n; i++) {
```

```
    sum += i;
```

```
}
```

```
System.out.print(sum);
```

Задача

Въвеждаме число n . Пресметнете $n!$ (чете се ен факториел).

$$1! = 1$$

$$2! = 1 * 2$$

$$3! = 1 * 2 * 3$$

....

$$n! = 1 * 2 * 3 * * (n-1) * n$$

Решение

```
int mult = 1;
```

```
int n = 100;
```

```
for (int i = 1; i <= n; i++) {
```

```
    mult = mult * i;
```

```
}
```

//в този случай брояча ни е вграден в цикъла и се казва i

Задача

Да се напише програма, която въвежда от клавиатурата целите числа m и n , ($m < n$). Програмата да извежда всички числа в интервала $[m, n]$, които са кратни на 5.

Вложени цикли

Вложени цикли

Вложените цикли представляват конструкция от няколко цикъла един в друг. Най-вътрешния цикъл се изпълнява най-много пъти.

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


Пример

```
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 5; j++) {  
        System.out.print (j);  
    }  
    System.out.println();  
}
```

//результат

//01234

//01234

//01234

Пример

По дадено число n да отпечатаме на конзолата триъгълник с n на брой реда, изглеждащ по следния начин:

1

1 2

1 2 3

...

1 2 3 ... n

Решение

```
Scanner input = new Scanner(System.in);
```

```
int n = input.nextInt();
```

```
for (int row = 1; row <= n; row++) {
```

```
    for (int col = 1; col <= row; col++) {
```

```
        System.out.print(col + " ");
```

```
    }
```

```
    System.out.println();
```

```
}
```

Задача

Напишете програма, която изкарва на конзолата всички щастливи четирицифрени числа. Щастливи числа са, когато сборът на първите две цифри на дадено четирицифрено число (ABCD) е равен на сбора на последните две ($A+B == C+D$).

Решение

```
for (int a = 1; a <= 9; a++) {  
    for (int b = 0; b <= 9; b++) {  
        for (int c = 0; c <= 9; c++) {  
            for (int d = 0; d <= 9; d++) {  
                if ((a + b) == (c + d)) {  
                    System.out.println("" + a + b + c + d);  
                }  
            }  
        }  
    }  
}
```

Задачи

Задача

1. Напишете програма, която приема дадено число n от конзолата и отпечатва всички числа от 0 до n , които се делят на 5 или на 3 без остатък.

Задача

2. Напишете програма, която отпечатва на конзолата всички четни числа от 0 до 1000.

Задача

3. Напишете програма, която приема дадено число n и отпечатва триъгълник, като в примера по-долу:

$n = 3$:

//резултат

*

**

Задача

4. Модифицирайте задача 3, като след като въведете дадено число, резултатът който трябва да се показва да бъде триъгълник изглеждащ по следния начин:

$n = 3$

//резултат

*

**

**

*

Домашно

Задача

1. Напишете програма която отпечатва триъгълник със страна n .

Пример:

$n = 3$

```
*  
  
***  
  
*****
```

$n = 5$

```
  *  
 ***  
*****  
  
*****  
*****
```

Задача

2. Напишете програма, която чете от конзолата положително цяло число N ($N < 20$) и отпечатва матрица с числа като на фигурата по-долу:

$N = 3$

1	2	3
2	3	4
3	4	5

$N = 4$

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

Задача

3. Напишете програма, която чете от конзолата положително цяло число n и отпечатва ромб като на фигурата по-долу:

$n = 1$

^
v

$n = 2$

^
/ ^ \
\\ v /
v

$n = 3$

^
/ ^ \
// ^ \\
\\ v //
\\ v /
v

Задача

Гледайте това видео:

<https://www.youtube.com/watch?v=eNpqcfvH7jw>

- Ако искате пропуснете частта за foreach, за момента тя не ни интересува
- Урока е за C#, но на Java е същото

Задача

Прочетете 6 глава от книгата: „**Цикли**”

<http://www.introprogramming.info/intro-java-book/read-online/glava6-cikli/>