



# Увод в програмирането с Java



# Цикли (упражнение)

**Преговор**

# Конструкция за цикъл `while`

**while** изпълнява набор от операции, **докато** дадено условие е вярно:

```
while (условие) {
```

```
    израз1;
```

```
    израз2;
```

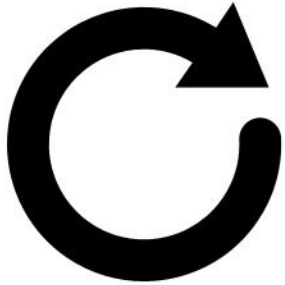
```
    ...
```

```
    изразn;
```

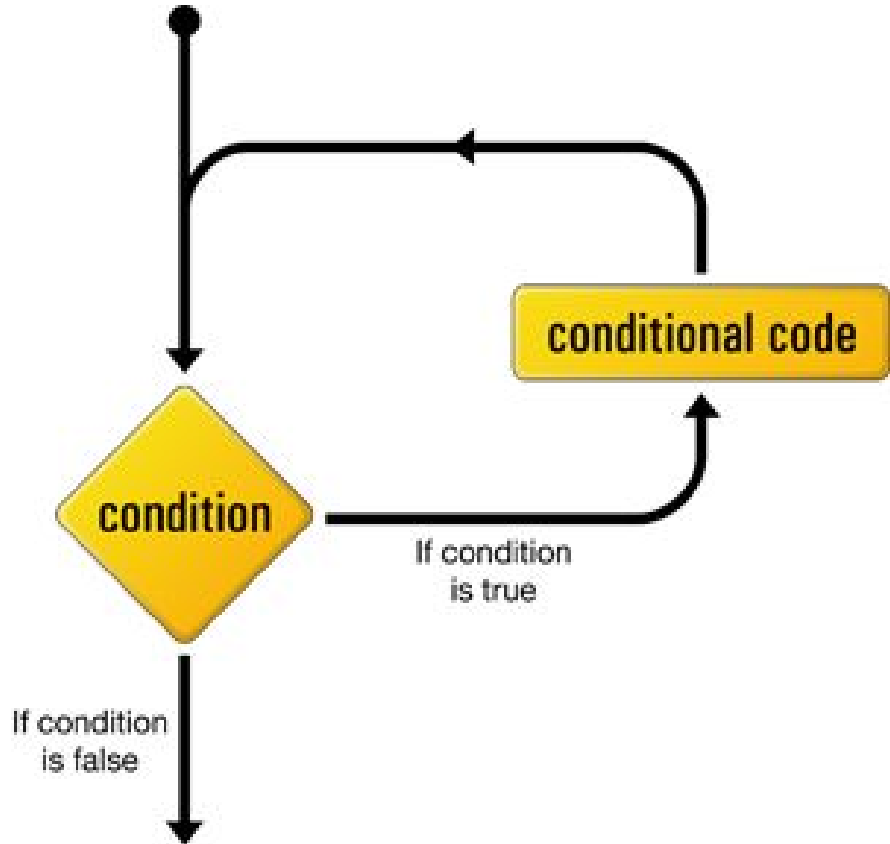
```
    // код, който се изпълнява, ако условието е true
```

```
}
```

# Конструкция за цикъл while



LOOPS REPEAT  
ACTIONS...  
SO YOU DON'T HAVE TO



# 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);*

# Цикъл do-while - пример

```
int number = 1;
```

```
do {
```

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

```
number++;
```

```
}
```

```
while (number <= 10);
```

# Безкраен цикъл

Безкрайният цикъл е цикъл, който никога не завършва. Създаването на безкраен цикъл трябва да се избягва, тъй като той кара програмата ни да “увисне” - да продължава да изпълнява едно и също нещо безкрайно, което от потребителска гледна точка не е желателно да става.

```
while (true) {  
  
System.out.println("Blablabla");  
  
}
```



# Оператор break

Операторът **break** се използва за прекъсване на цикъл. Всичко след него се пренебрегва и програмата излиза от цикъла:

```
while (условие) {
```

```
// код, който ще се изпълни
```

```
break;
```

```
// код, който няма да се изпълни
```

```
}
```

# Оператор break - пример

```
while (true) {  
System.out.println("You'll see me");  
break;  
System.out.println("But you won't see me :(");  
}  
System.out.println("Out of loop");
```

# Оператор break - пример

```
Scanner input = new Scanner(System.in);  
  
int number;  
  
while (true){  
  
    System.out.println("Please, insert a number: ");  
  
    number = input.nextInt();  
  
    if (number == 0)  
  
        break;  
  
}
```

# Оператор `continue`

Операторът **`continue`** се използва за преминаване към следващата итерация. Всичко след него се пренебрегва, но програмата не излиза от цикъла, а се връща към оценяване на условието:

```
while (условие) {
```

```
// код, който ще се изпълни
```

```
continue;
```

```
// код, който няма да се изпълни,
```

```
а вместо него, ще се оцени наново условието
```

```
}
```

# Цикъл for

Използва се, когато знаем точно колко завъртания ще има цикълът.

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

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

# Цикъл for - сума на числата от 1 до n

```
int sum = 0;

for (int i = 0; i <= n; i++) {
    sum += i;
}

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

# Вложени цикли

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

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

# Вложени цикли

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



# Задачи

# Задача

Да се изведат в конзолата всички положителни числа до 2 милиона, които се делят на 11.

# Задача

Да се модифицира предната задача и да се изведат само числата, които се делят едновременно и на 11, и на 3.

# Задача

Напишете прост калкулатор за събиране, който работи по следния начин:

- Очаква въвеждане на първото число в конзолата;
- Очаква въвеждане на второто число в конзолата;
- Извежда в конзолата резултата - сбора от двете числа;
- Ако второто въведено число е 0, прекратява изпълнението, след като изведе резултата.

# Задача

Програмирайте автомат за алкохол. Дадени са 4 напитки (предварително ги определяте в кода). При стартиране на програмата в конзолата се извеждат имената и кодовете на напитките.

Програмата иска въвеждане на възраст и код на напитка. Възможните изходи са:

- Ако е въведена възраст под 18 години - извежда съобщение, че не може да се поръчва алкохол;
- Ако е въведена възраст над 18 години - извежда “Заповядай твоето [име на напитка]!” (например “Заповядай твоето уиски!”)

# Задача

Да се направи въвеждането на възраст и номер на напитка да е в цикъл, като при въвеждане на код 0 (вместо номер на напитка) да се прекратява изпълнението.

# Задача

Да се направи въвеждането на възраст и номер на напитка да е в цикъл, като при въвеждане на код 0 (вместо номер на напитка) да се прекратява изпълнението.

Нека всяка напитка си има цена. При избиране на напитката, на екрана се извежда колко е сметката ви до момента. При натискане на 0, се извежда крайната сметка.

**Домашно**



# Задача

Да се напише програма прост калкулатор. Калкулаторът се състои в следното:

1. Чете се от конзолата действието, което ще се извърши с числата (събиране, изваждане, умножение или деление);
2. Чете се от конзолата първото число;
3. Чете се от конзолата второто число;
4. В конзолата се извежда изразът и резултатът от него

Например, ако са въведени “+”, 2, 3, резултатът е:  $2 + 3 = 5$ .

# Задача

Клиентът ви е строителна компания, която иска от вас да направите каталог на апартаментите в новопостроената им жилищна сграда.

Сградата е на 6 етажа. На всеки етаж има по 10 апартамента.

Номерацията е номер на етаж + число от 1 до 10 (напр. за 2 етаж са 201, 202 и т.н. до 210, за етаж 6 са от 601 до 610 и т.н.).

Апартаментите са ориентирани по следния начин:

- апартаментите с номера от 1 до 5 са ориентирани на запад
- апартаментите с номера от 6 до 10 са ориентирани на изток

Апартаменти 5 и 10 са четиристайни. От останалите апартаменти, тези с четни номера са двустайни, а тези с нечетни - тристайни.

Каталогът трябва да съдържа списък на всички апартаменти в сградата и да има следния формат: ***Floor, Apartment number, Rooms, Orientation***

# Задача

В резултата е включен и заглавния ред, съдържащ имената на колоните (данните).

Пример за етаж 3:

**Floor, Apartment number, Rooms, Orientation**

**3, 301, 3, west**

**3, 302, 2, west**

**3, 303, 3, west**

**3, 304, 2, west**

**3, 305, 4, west**

**3, 306, 2, east**

**3, 307, 3, east**

**3, 308, 2, east**

**3, 309, 3, east**

**3, 310, 4, east**

# Задача

По същия начин отпечатваме и резултатите за всички етажи от 1 до 6 последователно. Резултатът да се отпечата в конзолата.