

Работа с по-сложни цикли

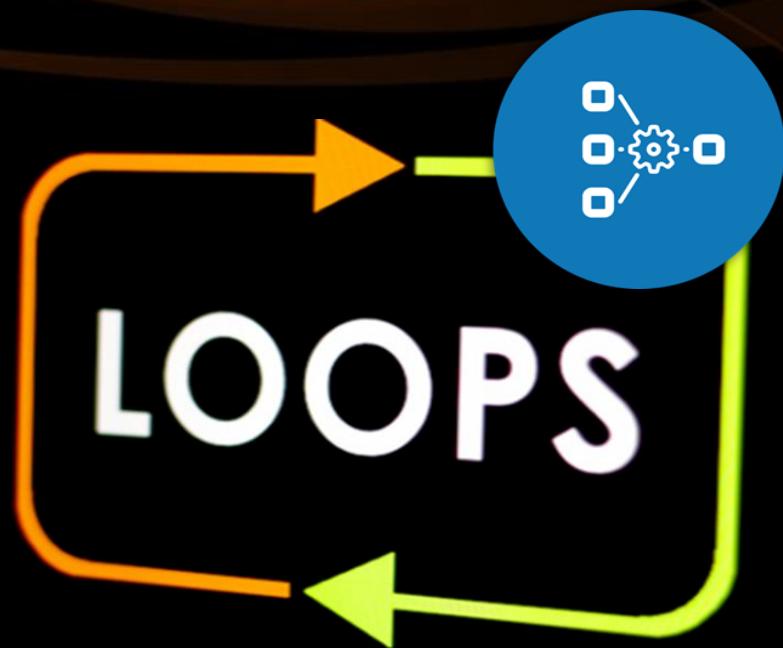
Цикли със стъпка, While, Do...While



Софтуни
трейнърски екип
Софтуерен университет
<http://softuni.bg>



Основи на
програмирането



Имате въпроси?



Sli.do

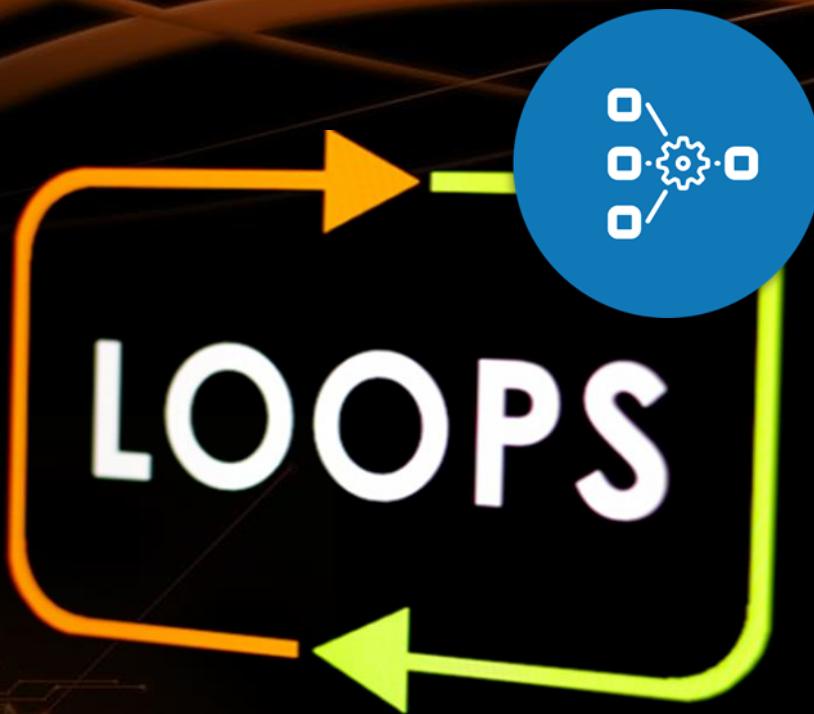
#pb-may

Съдържание

- По-сложни конструкции за цикъл:

- Цикъл със стъпка
- Цикъл с намаляваща стъпка
- Цикъл - **while**
- Цикъл - **do-while**
- Безкраен цикъл
 - Оператор **break**
 - Оператор **continue**





Цикли със стъпка

Работа с по-сложни `for`-цикли

Числата от 1 до N през 3 - условие

- Напишете програма, която:
 - Прочита цяло число **n**
 - Отпечатва числата от **1** до **n** със стъпка **3**
- Примерен вход и изход:

10 → 1, 4, 7, 10



Числата от 1 до N през 3 - решение

```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
for (int i = 1; i <= n; i+=3) {
    System.out.println(i);
```

Задаване
на стъпка

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#0>

- Напишете програма, която:
 - Прочита цяло число **n**
 - Отпечатва числата от **n** до **1** в обратен ред (**стъпка -1**)
- Примерен вход и изход:

100



100, 99, 98, ..., 3, 2, 1

Числата от N до 1 в обратен ред - решение

```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
for (int i = n; i >= 1; i--) {
    System.out.println(i);
}
```

Намаляваща стъпка: -1

Обърнато условие:
 $i \geq 1$

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#1>

Числата от 1 до 2^n с for-цикъл – условие

- Напишете програма, която:
 - Прочита цяло число n
 - Отпечатва числата от 1 до 2^n
- Примерен вход и изход:

8	2	8	
16	128	4	2
2	4	2	4
2	32	8	

10 → **1, 2, 4, 8, 16, 32, ..., 1024**

5 → **1, 2, 4, 8, 16, 32**

Числата от 1 до 2^n с for-цикъл - решение

```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
int num = 1;
for (int i = 0; i <= n; i++) {
    System.out.println(num);
    num = num * 2;
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#2>

Четни степени на 2 - условие

- Напишете програма, която:
 - Прочита цяло число n
 - Отпечатва четните степени на 2 до 2^n : $2^0, 2^2, 2^4, 2^8, \dots, 2^n$
- Примерен вход и изход:

10



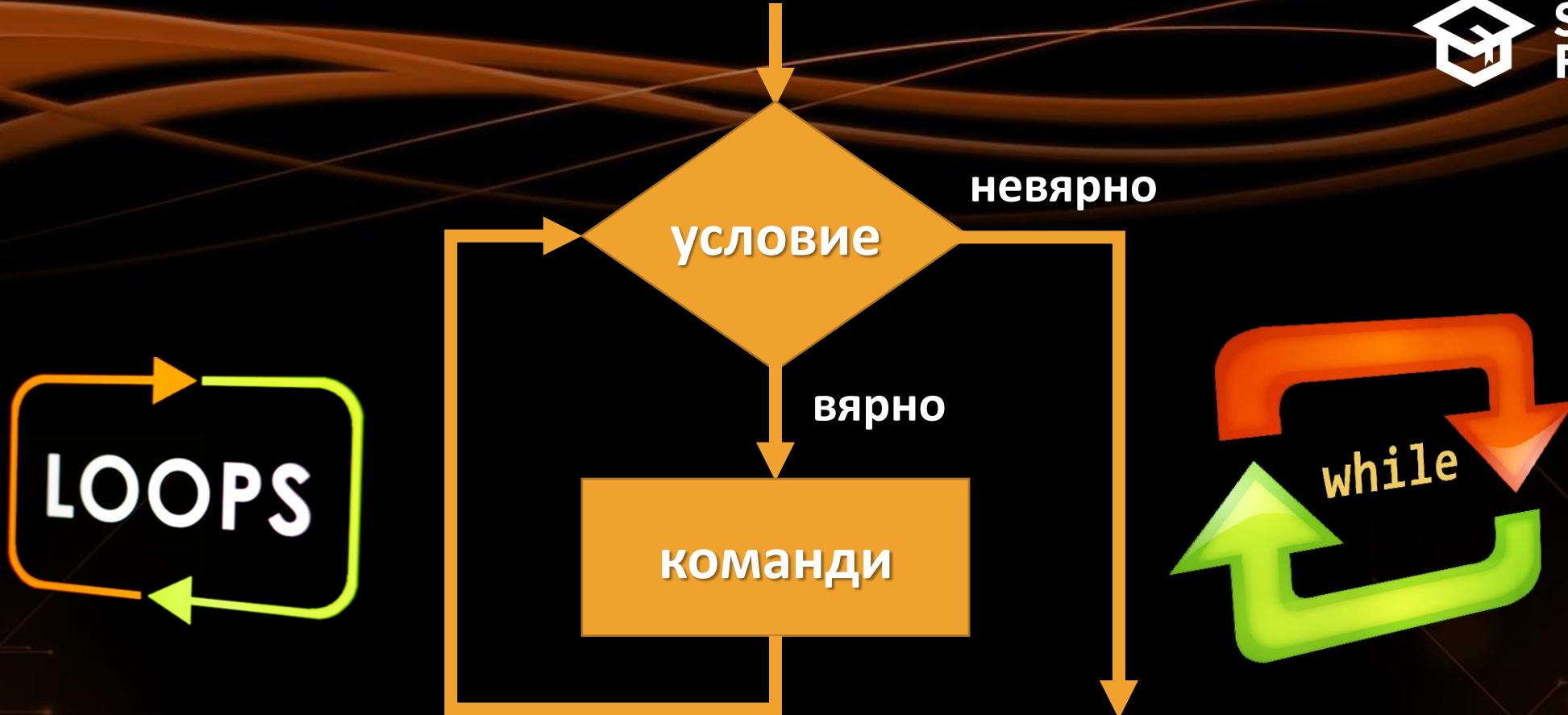
1, 4, 16 , ..., 1024

Четни степени на 2 - решение

```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
int num = 1;
for (int i = 0; i <= n; i+=2) {
    System.out.println(num);
    num = num * 2 * 2;
}
```

Ползваме
стъпка 2

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#3>



While цикъл

Повторение докато е в сила дадено условие

While цикъл

- Тялото на цикъла се изпълнява **докато** е вярно дадено условие

```
Условие  
while (...) {  
    //code  
}
```

Код за
изпълнение
(повторение)

Редица числа $2k+1$ - условие

- Напишете програма, която:
 - Прочита цяло число **n**
 - Отпечатва всички числа $\leq n$ от редицата: **1, 3, 7, 15, 31, ...**
 - Всяко следващо число е равно на предишното * 2 + 1

$1, (1*2)+1 = 3, (3*2)+1 = 7, (7*2)+1 = 15 \dots$

Редица числа $2k+1$ - решение

```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
int k = 1;
while (k <= n) {
    System.out.println(k);
    k = 2 * k + 1;
}
```

Повторение докато е в сила условието $k \leq n$

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#4>

Число в диапазона [1...100] - условие

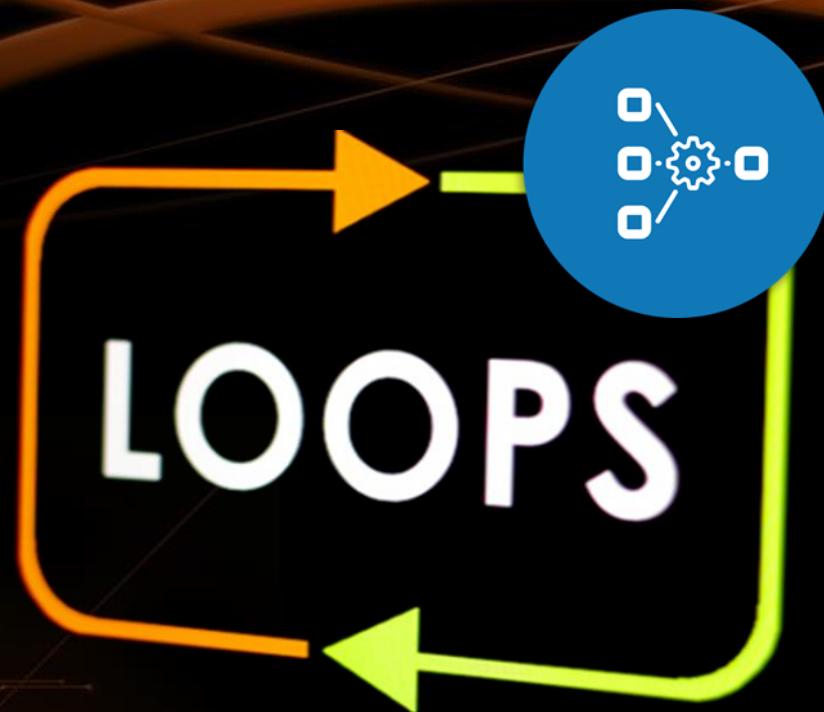
- Напишете програма, която:
 - Прочита цяло число
 - Проверява дали е в диапазона [1...100]
 - При:
 - Намиране на число в диапазона,
прекратява изпълнение
 - Невалидно число прочита ново



Число в диапазона [1...100] - решение

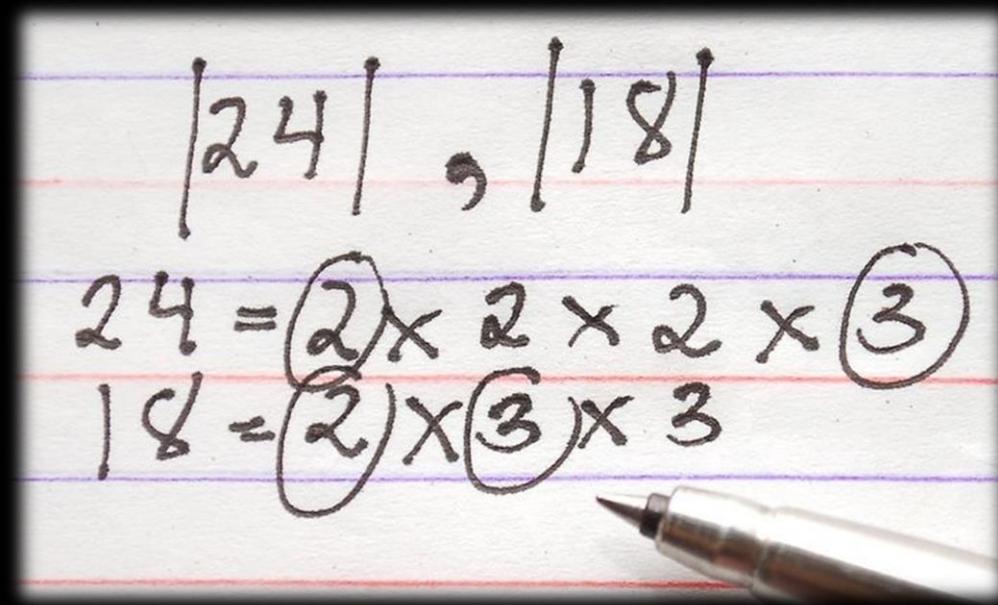
```
Scanner scanner = new Scanner(System.in);
int num = Integer.parseInt(scanner.nextLine());
while (num < 1 || num > 100) {
    System.out.println("Invalid number!");
    num = Integer.parseInt(scanner.nextLine());
}
System.out.println("The number is: " + num);
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#5>



Цикли със стъпка и while цикъл

Решаване на задачи в клас(лаб)

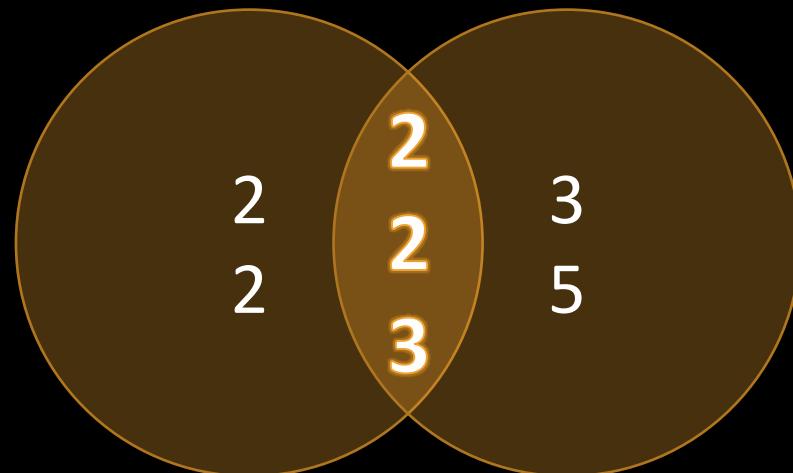


Най-голям общ делител (НОД)

Алгоритъм на Евклид

Най-голям общ делител (НОД)

- Най-голям общ делител (НОД) на две естествени числа **a** и **b** е най-голямото число, което дели едновременно **a** и **b** без остатък
 - $\text{НОД}(24, 16) = 8$
 - $\text{НОД}(67, 18) = 1$
 - $\text{НОД}(12, 24) = 12$
 - $\text{НОД}(15, 9) = 3$
 - $\text{НОД}(10, 10) = 10$
 - $\text{НОД}(100, 88) = 4$



Алгоритъм на Евклид за НОД - условие

Напишете програма, която:

- Прочита 2 цели числа **a** и **b**
- Намира най-големия им общ делител - **НОД(a, b)**

Насоки:

- Докато не се достигне остатък 0:
 - Дели се по-голямото число на по-малкото
 - Взема се остатъка от делението

Псевдо код:

```
while b ≠ 0
    int oldB = b;
    b = a % b;
    a = oldB;
print a;
```

Алгоритъм на Евклид за НОД - решение

```
Scanner scanner = new Scanner(System.in);
int a = Integer.parseInt(scanner.nextLine());
int b = Integer.parseInt(scanner.nextLine());
while (b != 0) {
    int oldB = b;
    b = a % b;
    a = oldB;
}
System.out.println("GCD = " + a);
```

a = 600, b = 136
600 / 136 = 4 (ост. 56)
136 / 56 = 2 (ост. 24)
56 / 24 = 2 (ост. 8)
24 / 8 = 3 (ост. 0)
НОД = 8

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#6>

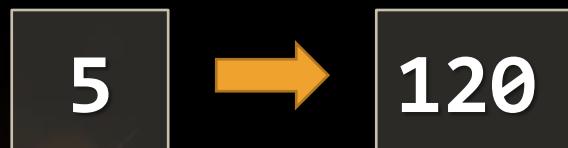


Do...While цикъл

Повторение докато е изпълнено условието

Изчисляване на факториел - условие

- Напишете програма, която:
 - Прочита естествено число **n**
 - Изчислява факториел от n ($n!$)
- Примерен вход и изход:
 - $5! = 1 * 2 * 3 * 4 * 5 = 120$



$n!$

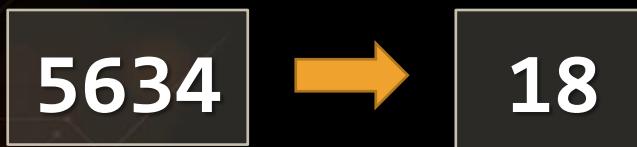
Изчисляване на факториел - решение

```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
int fact = 1;
do {
    fact = fact * n;
    n--;
} while (n > 1);
System.out.println(fact);
```

n!

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#7>

- Напишете програма, която:
 - Прочита цяло положително число **n**
 - Сумира цифрите на **n**
- Примерен вход и изход:
 - $n = 5634: 5 + 6 + 3 + 4 = 18$



Сумиране на цифрите на число - решение

```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
int sum = 0;
do {
    sum = sum + (n % 10);           n % 10 ни дава последната
    n = n / 10;                    цифра на числото n           n / 10 премахва последната цифра на n
} while (n > 0);
System.out.println("Sum of digits: " + sum);
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#8>



Безкрайни цикли и оператор break

Безкраен цикъл

- Безкраен цикъл – повтаряне на блок от код до безкрайност:

Условието е
винаги вярно

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



Условия за прекратяване на цикъл

Условие за прекратяване
на цикъл

```
while(...) {  
    System.out.println("Infinite loop");  
}
```

- Пример:

```
int a = 5;  
while(a < 10) {  
    System.out.println("a = " + a);  
    a++;  
}                                // 10  
System.out.println(a)
```

Безкраен цикъл (2)

- Оператор **break** – прекъсване на цикъла

```
while(true) {  
    System.out.println("Infinite loop");  
    if (...) {  
        break;  
    }  
}
```

Условие за прекъсване на
цикъла

Прости числа - условие

- Напишете програма, която:
 - Прочита цяло число **n**
 - Проверява да ли **n** е просто число
- Насоки:
 - Едно число **n** е **просто**, ако се дели единствено на **1** и **n**
 - Прости числа: **2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, ...**
 - Композитни числа: $10 = 2 * 5$, $21 = 3 * 7$, $143 = 13 * 11$

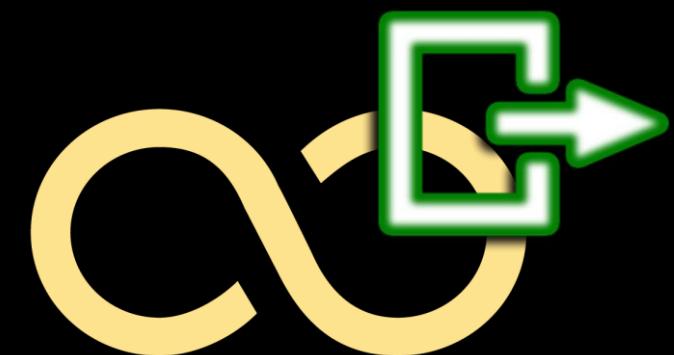
Проверка за просто число - решение

```
int n = Integer.parseInt(scanner.nextLine());  
boolean prime = true;  
for (int i = 2; i <= Math.sqrt(n); i++) {  
    if (n % i == 0) {  
        prime = false;  
        break; // break излиза от текущия цикъл  
    }  
}  
if (prime) { System.out.println("Prime"); }  
else System.out.println("Not prime");
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#9>

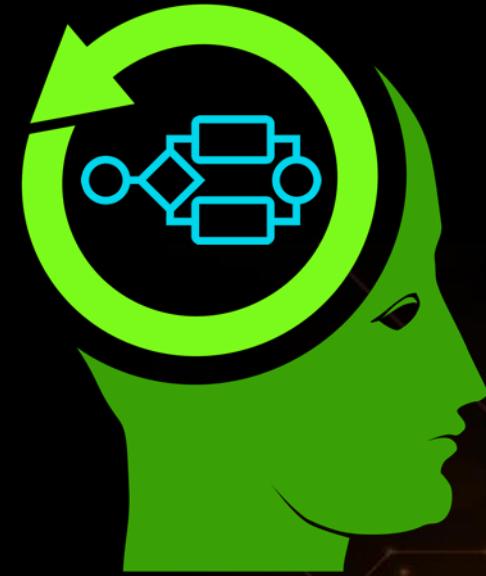
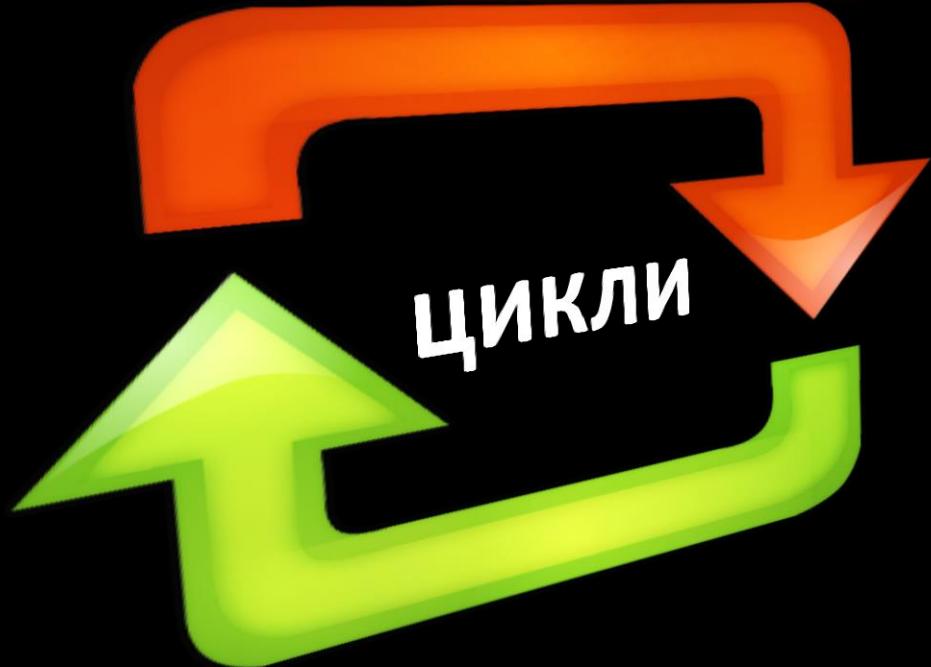
Четно число - условие

- Напишете програма, която:
 - Прочита число **n**
 - Проверява дали **n** е четно
 - При невалидно число се връща към повторно въвеждане



Четно число - решение

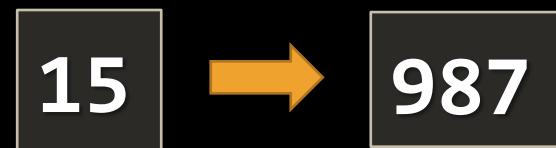
```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
int n = 0;
while (true) {
    System.out.print("Enter even number: ");
    n = Integer.parseInt(scanner.nextLine());
    if (n % 2 == 0) {
        break; // even number -> exit from the loop
    }
    System.out.println("The number is not even.");
}
System.out.println("Even number entered: " + n);
```



Задачи с цикли

Числа на Фибоначи - условие

- Напишете програма, която:
 - Прочита число **n**
 - Пресмята **n-тото** число на Фибоначи
- Числата на Фибоначи са следните: **1, 1, 2, 3, 5, 8, 13, 21, 34, ...**
 - $F_0 = 1$
 - $F_1 = 1$
 - $F_n = F_{n-1} + F_{n-2}$
 - Примерен вход и изход:



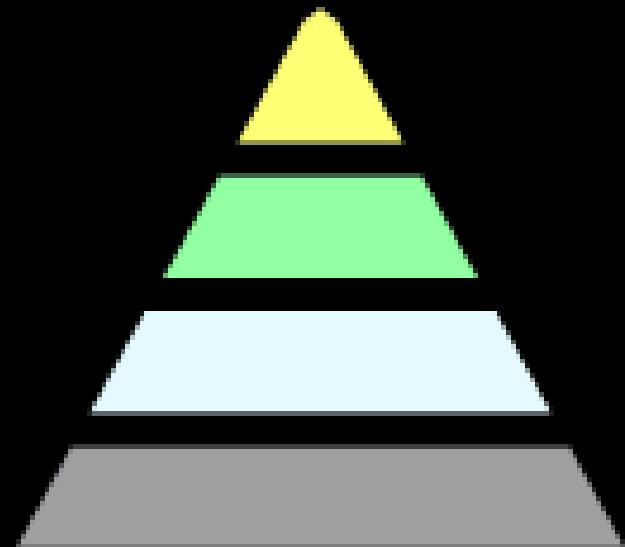
Числа на Фиbonачи - решение

```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
int f0 = 1;
int f1 = 1;
for (int i = 0; i < n-1; i++) {
    int fNext = f0 + f1;
    f0 = f1;
    f1 = fNext;
}
System.out.println(f1);
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#11>

Пирамида от числа - условие

- Напишете програма, която:
 - Прочита цяло число **n**
 - Отпечатва числата от 1 до **n** в пирамида



Пирамида от числа – условие (2)

- Примерен вход и изход:

$n = 7$



1
2 3
4 5 6
7

$n = 10$



1
2 3
4 5 6
7 8 9 10

$n = 12$



1
2 3
4 5 6
7 8 9 10
11 12

$n = 15$



1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

Пирамида от числа – решение

```
int n = Integer.parseInt(scanner.nextLine());
int num = 1;
for (int row = 1; row <= n; row++) {
    for (int col = 1; col <= row; col++) {
        if (col > 1) System.out.print(" ");
        System.out.print(num);
        num++;
        if (num > n) { break; }
    }
    System.out.println();
    if (num > n) { break; }
}
```

1			
2	3		
4	5	6	
7	8	9	10
11	12		

Таблица с числа – условие

- Напишете програма, която:
 - Прочита цяло число n
 - Отпечатва числата от 1 до n в таблица



Таблица с числа – условие (2)

- Примерен вход и изход:

n = 2



1	2
2	1

n = 3



1	2	3
2	3	2
3	2	1

n = 4



1	2	3	4
2	3	4	3
3	4	3	2
4	3	2	1

n = 5

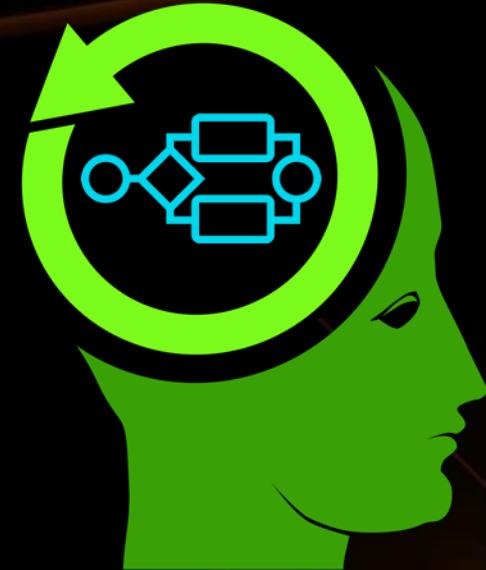
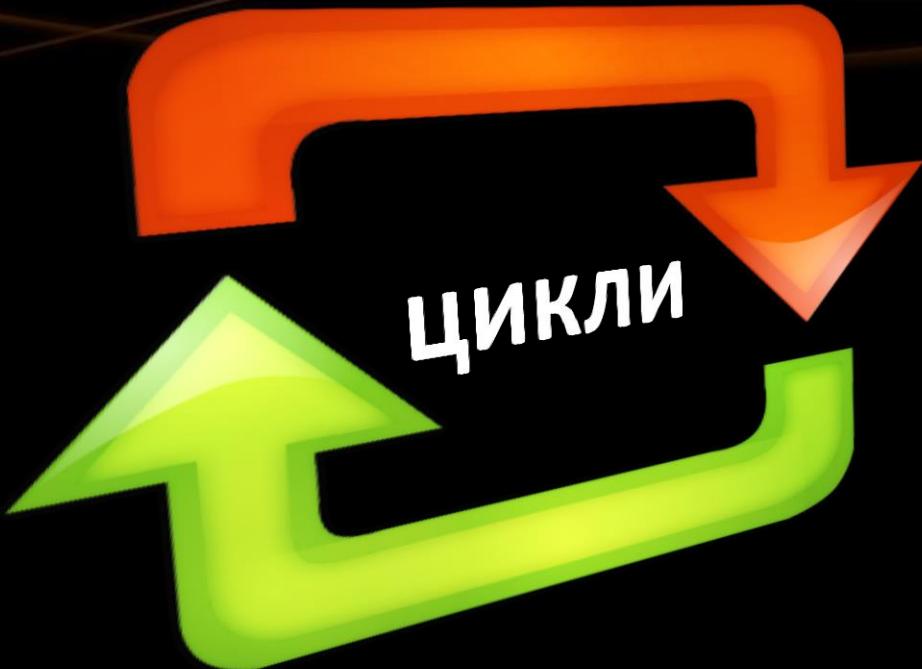


1	2	3	4	5
2	3	4	5	4
3	4	5	4	3
4	5	4	3	2
5	4	3	2	1

Таблица с числа – решение

```
Scanner scanner = new Scanner(System.in);
int n = Integer.parseInt(scanner.nextLine());
for (int row = 0; row < n; row++) {
    for (int col = 0; col < n; col++) {
        int num = row + col + 1;
        if (num > n) { num = 2 * n - num; }
        System.out.print(num + " ");
    }
    System.out.println();
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#13>



По-сложни задачи с цикли

Решаване на задачи в клас(лаб)

Какво научихме днес?

- Можем да ползваме **for**-цикли със стъпка:

```
for (int i = 1; i <= n; i+=3) {  
    System.out.println(i);  
}
```



Какво научихме днес? (2)

- Цикли **while** / **do-while** повтарят блок от код докато е в сила дадено условие:

```
int num = 1;  
while (num <= n) {  
    System.out.println(num++);  
}
```



Въпроси?



SoftUni Diamond Partners



INDEAVR
Serving the high achievers

 **INFRASTICS®**



SoftwareGroup
doing it right

 **xssoftware**

NETPEAK
SEO and PPC for Business

**SUPER
HOSTING**
®.BG

SoftUni Diamond Partners



LIEBHERR



- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"



- Благодарности: настоящият материал може да съдържа части от следните източници
 - Книга "Основи на програмирането със Java" от Светлин Наков и колектив с лиценз CC-BY-SA

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg

