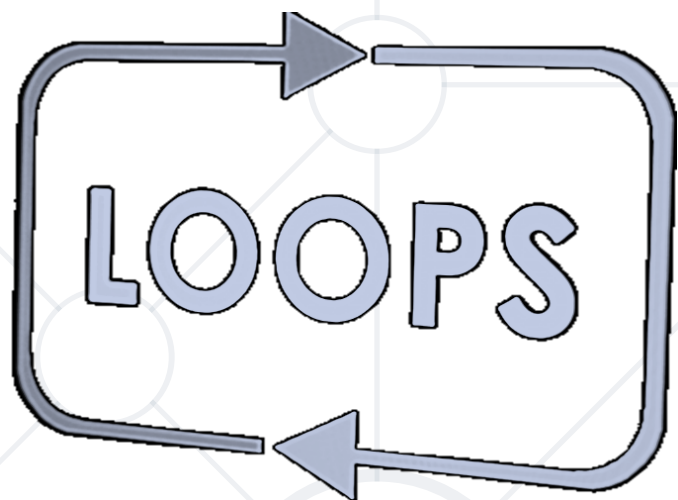


# Повторения (цикли)

Прости повторения с for-цикъл



СофтУни

Преподавателски екип



Software  
University



SoftUni  
Foundation



Софтуерен университет  
<http://softuni.bg>

# Имате въпроси?

sli.do

**#pb-march**

1. Преговор
2. Какво е цикъл?
3. for-цикъл – конструкция
4. Цикли със стъпка
5. Работа с текст
6. Техники за използване на for-цикли





**Преговор**

1. Каква ще е стойността на променливата **a** след изпълнението на следната програма:

```
let a = 5;  
switch (a){  
  case 5:  
  case 6:  
    a = a + 1;  
    break;  
  default:  
    a = a + 2;  
    break;  
}
```

0

5

6

7

2. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
console.log(!(5 == 5) && (4 + 1 == 5));
```

True

False

Runtime  
error

Compile time  
error

3. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
console.log(!(3 == 3) || (3 == 5));
```

True

Runtime  
error

False

Compile time  
error

4. Какво ще се отпечата на конзолата, ако изпълним следната проверка:

```
console.log(!(3 > 5) || (1 == 1));
```

Compile time  
error

Runtime  
error

False

True



5. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

```
let number = 101;  
if (number >= 1)  
    console.log("Larger than 1");  
if (number <= 101)  
    console.log("Less than 101");  
console.log("Equal to 101");
```

Larger than 1

Less than 101

Equal to 101

Compile  
time error

6. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

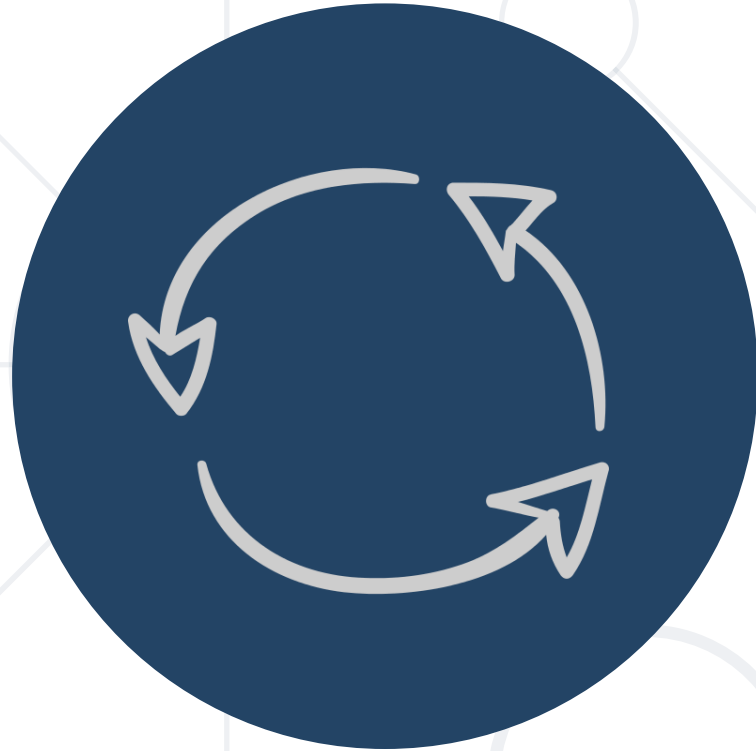
```
let role = "Administrator";  
let password = "SoftUni";  
if (role == "SoftUni") {  
    if (password == "SoftUni") {  
        console.log("Welcome!");  
    }  
}
```

"Welcome!"

Runtime error

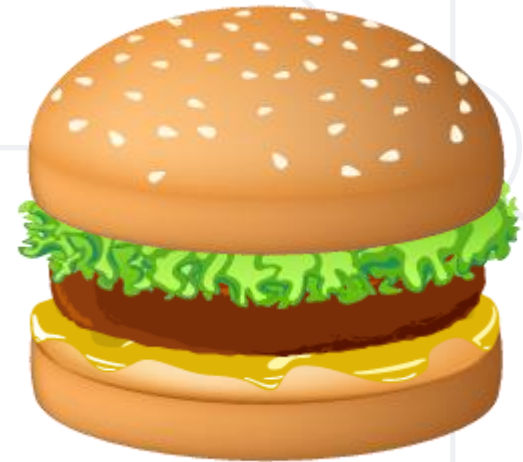
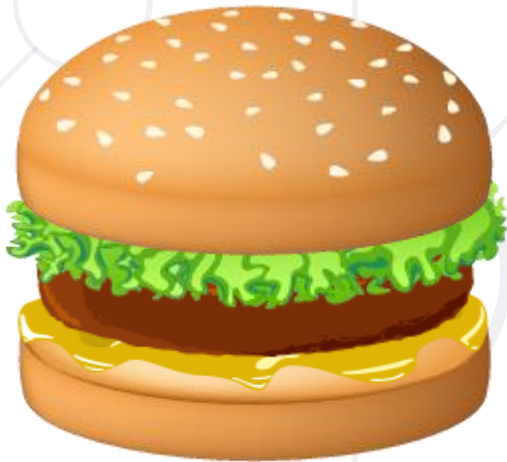
No output

Compile time  
error



**Цикли**

- Често ни се налага да повтаряме едно и също действие многократно
- Ако искаме да направим 3 бургера, бихме повторили едни и същи действия 3 пъти:



# Какво е цикъл? (2)

- Циклите в програмирането ни позволяват да повтаряме **едни и същи действия** определен брой пъти:

```
for (let i = 1; i <= 3; i++) {  
  console.log("Bottom bun");  
  console.log("Mustard");  
  console.log("Meat");  
  console.log("Lettuce");  
  console.log("Top bun");  
}
```



# **Повторения на блокове код**

## **Конструкция за for-цикъл**

- Можем да повтаряме действия до определен момент чрез **for**-цикли

Ключова дума за  
конструкцията

Начална  
стойност

Крайна  
стойност

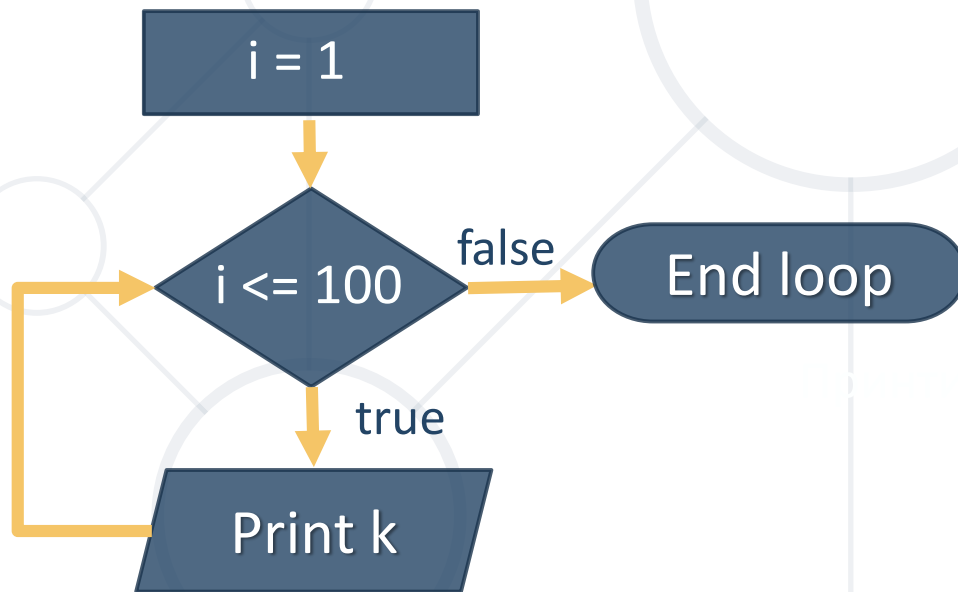
```
for (let i = 1; i <= 10; i++) {  
  console.log(i);  
}
```

Инкрементация  
на индекса (i)

Тяло на цикъла: блок от код за  
повторение

# Числа от 1 до 100

- Напишете програма, която:
  - Извежда числата [1...100] всяко на нов ред



Тестване на решението: <https://judge.softuni.bg/Contests/Compete/Index/2321#0>



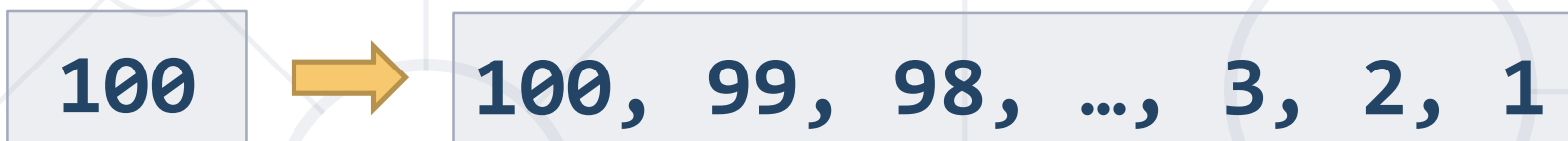


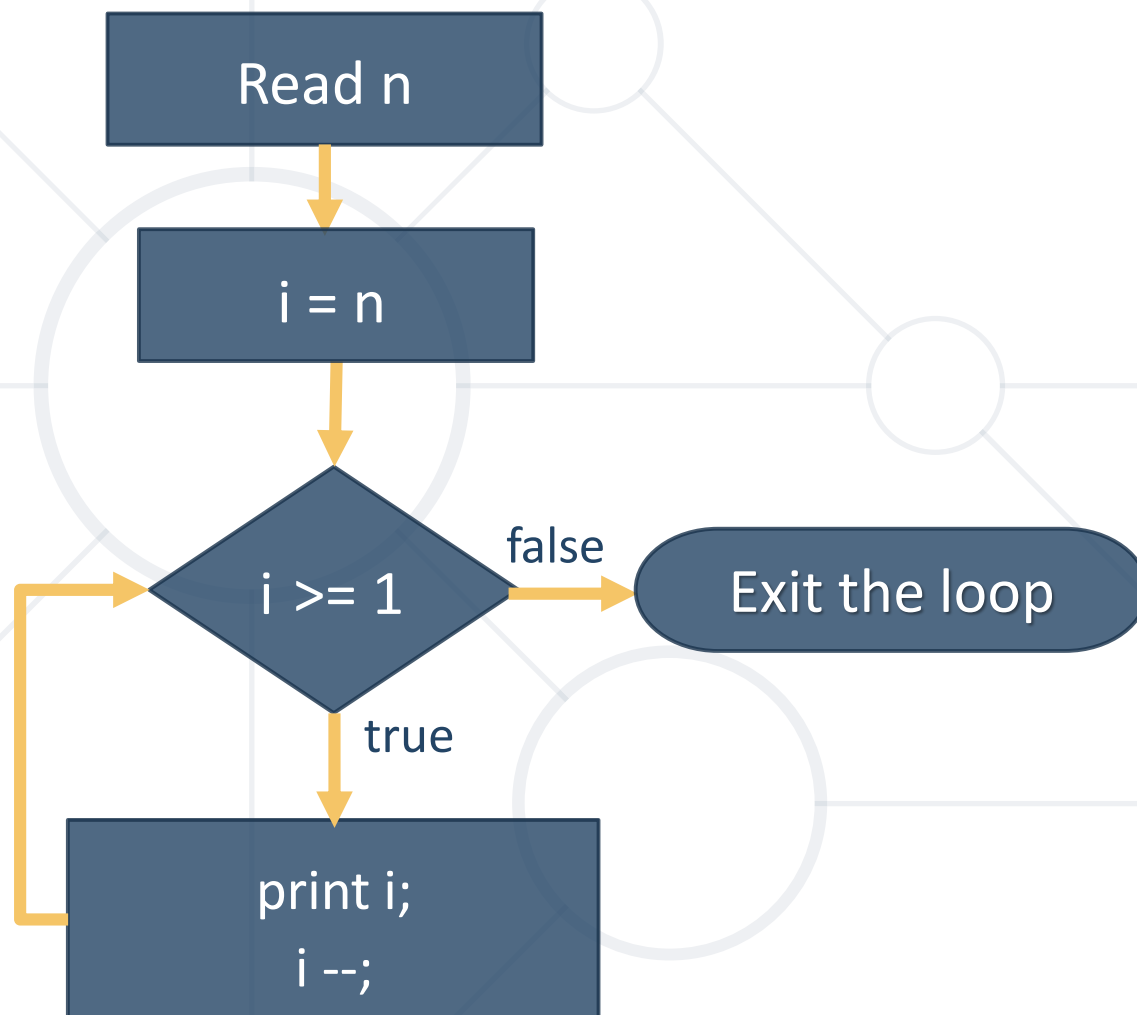
# Работа с по-сложни for-цикли

## Цикли със стъпка

# Числата от N до 1 в обратен ред – условие

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





# Числата от N до 1 в обратен ред – решение

```
let n = Number(number)
for (let i = n; i >= 1; i--) {
  console.log(i);
}
```

Намаляваща стъпка: -1

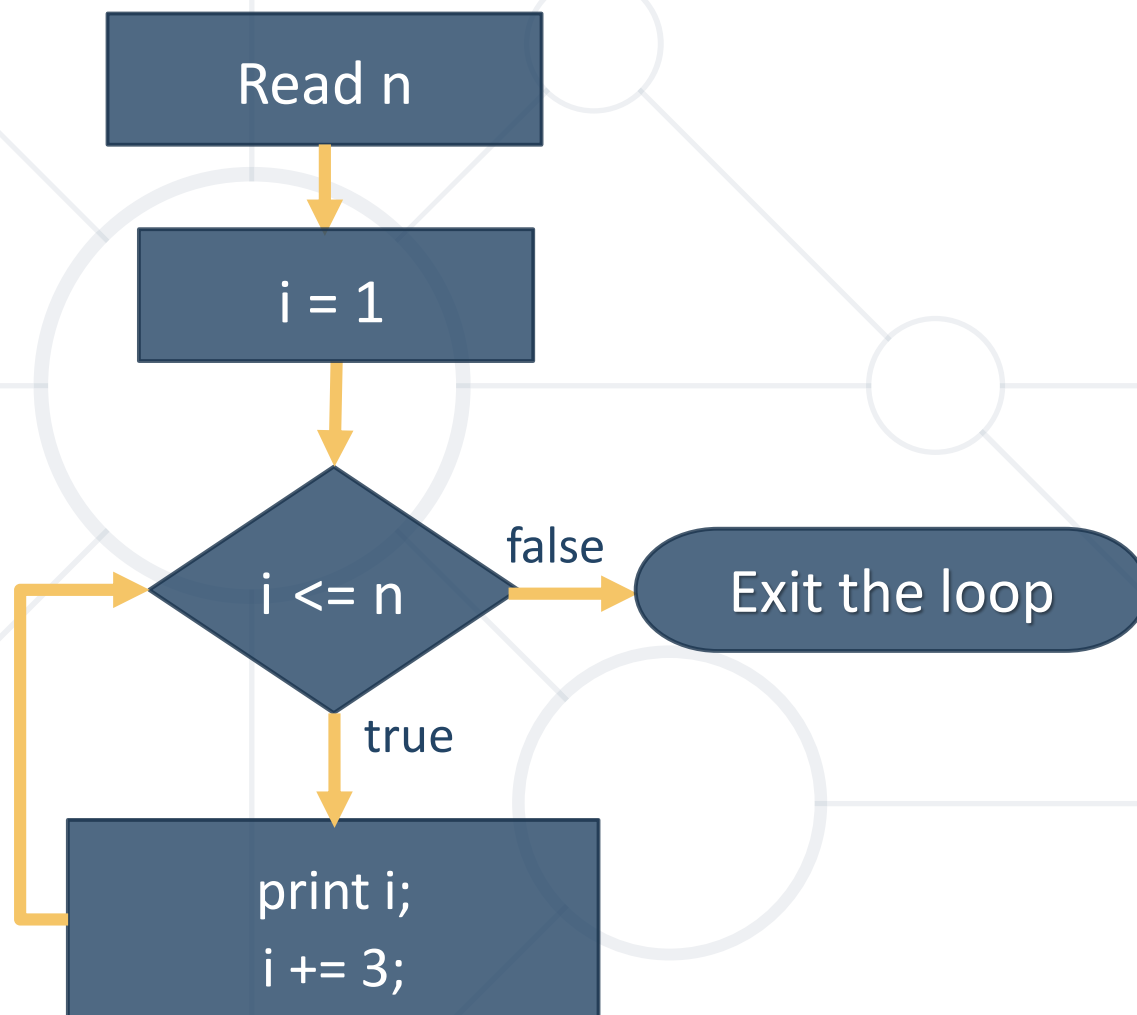
Обърнато условие:  $i \geq 1$

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/2321#1>

# 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:





# Числата от 1 до N през 3 – решение

```
let n = Number(number);  
for (let i = 1; i <= n; i += 3) {  
    console.log(i);  
}
```

Задаване на  
стъпка 3

# Четни степени на 2 – условие

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

10 → 1, 4, 16 , ..., 1024

7 → 1, 4, 16 , ..., 64



# Четни степени на 2 – решение

```
let n = Number(number);  
let num = 1;  
for (int i = 0; i <= n; i += 2) {  
    console.log(num);  
    num = num * 2 * 2;  
}
```

Ползваме  
стъпка 2



# Работа с текст

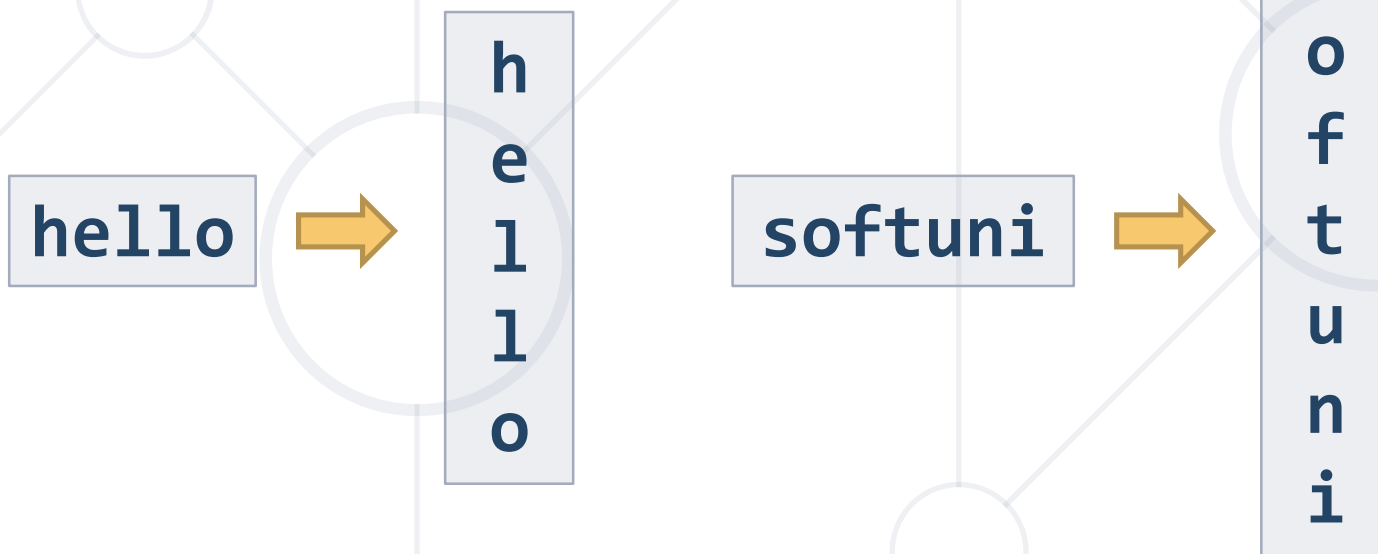
- Можем да вземем дължината на текст

```
let text = 'SoftUni';           // въвеждаме SoftUni  
let length = text.length;      // 7
```

- Можем да вземем символ от текст по индекс

```
let text = 'SoftUni';           // въвеждаме SoftUni  
let letter = text[4];           // U
```

- Напишете програма, която
  - чете текст(стринг)
  - печата всеки символ от текста на отделен ред
- Примерен вход и изход:



```
function characterSequence(input){  
  for (let i = 0; i < input.length; i++) {  
    console.log(input[i]);  
  }  
}
```

Взимаме дължината  
на текста

Взимаме всеки  
символ по индекс *i*

# Сумиране на гласни букви - условие

- Напишете програма, която:
  - Прочита от потребителя текст
  - Извежда сумата на гласните букви според таблицата по-долу:

а	е	и	о	и
1	2	3	4	5

- Примерен вход и изход:

hello



6

(e+o = 2+4 = 6)

hi



3

(i = 3)

bamboo



9

(a+o+o = 1+4+4 = 9)

beer



4

(e+e = 2+2 = 4)

# Сумиране на гласни букви - решение

```
function vowelsSum(input){  
  let sum = 0;  
  
  for (let i = 0; i < input.length; i++) {  
    switch (input[i]) {  
      case 'a': sum += 1; break;  
      case 'e': sum += 2; break;  
      // TODO: Add cases for the other vowels.  
    }  
  }  
  console.log("Vowels sum = " + sum);  
}
```

- Напишете програма, която:
  - Прочита от потребителя **текст, контролно число и бюджет**
  - Сумира буквите и ги умножава по **контролното число**
  - Проверява дали бюджета покрива сумата
  - Всички гласни букви имат стойност 3, а съгласните стойност 1
  - Гласни букви са: "а", "е", "і", "о", "u" и "у".

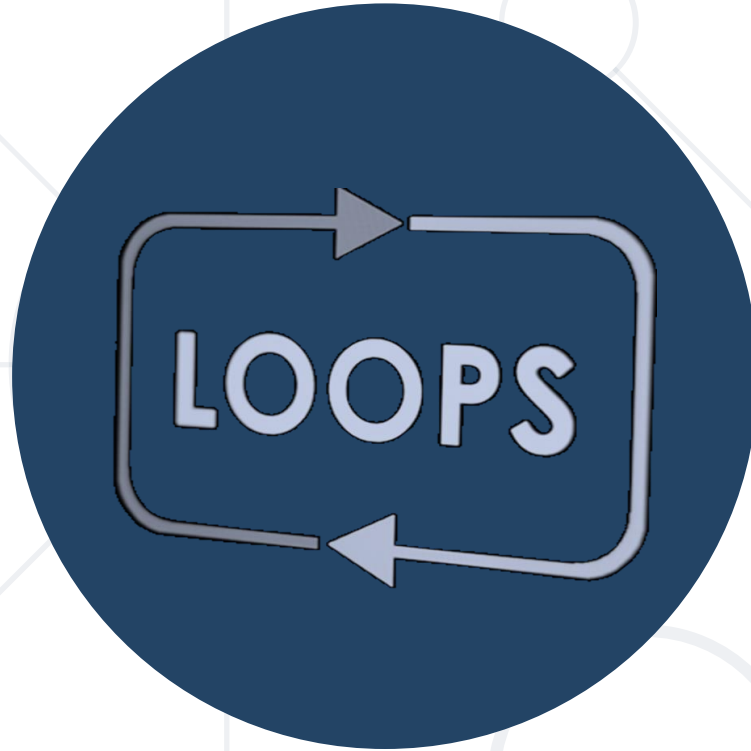


# Сбор от букви - пример

- Примерен вход и изход:



```
function lettersSum(productName, controlNumber, budget) {  
    let productValue = 0;  
    for (let i = 0; i < productName.length; i++) {  
        const currentChar = productName[i].toLowerCase();  
        // TODO: Add conditions for currentChar.  
    }  
    productValue *= controlNumber;  
  
    if (productValue > budget) {  
        console.log(`Cannot buy ${productName}. Product value:  
${productValue.toFixed(2)}`);  
    } else {  
        console.log(`${productName} bought. Money left: $${(budget -  
productValue).toFixed(2)}`);  
    }  
}
```



**Повторения на блокове код**  
**Решаване на задачи в клас (лаб)**

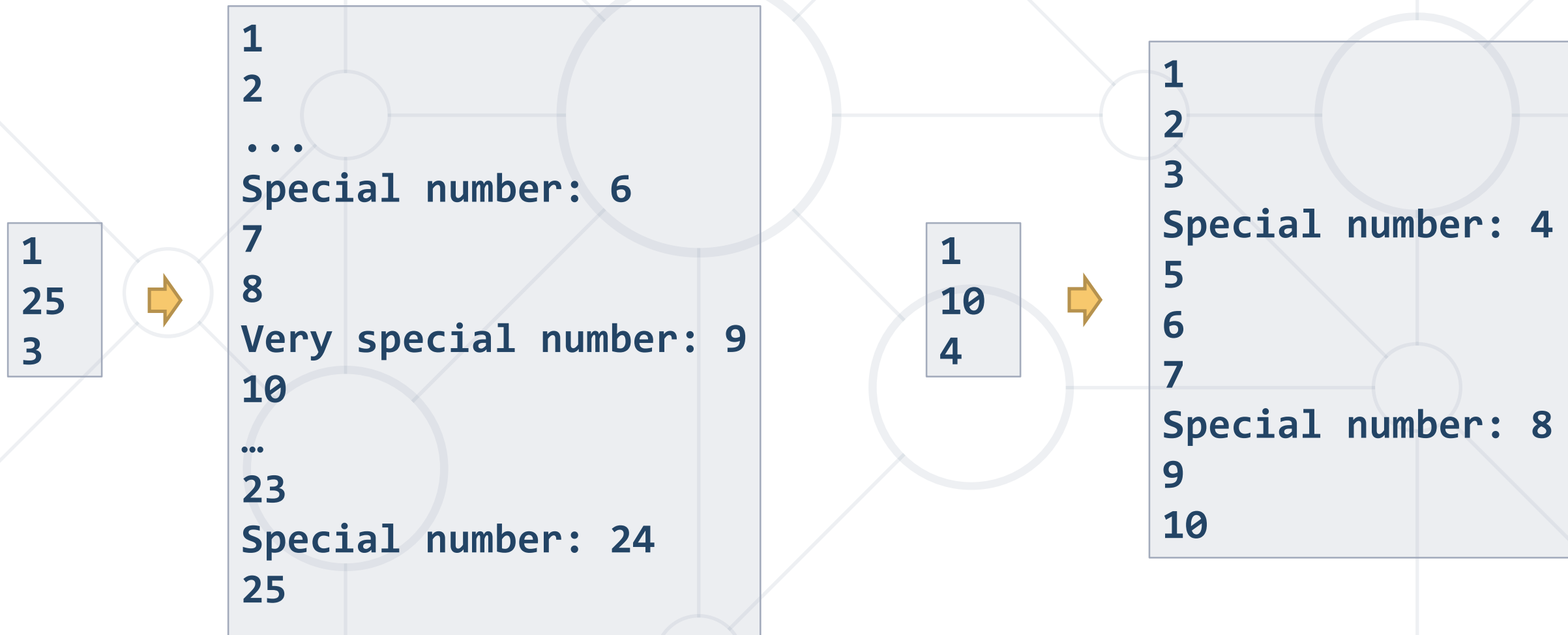


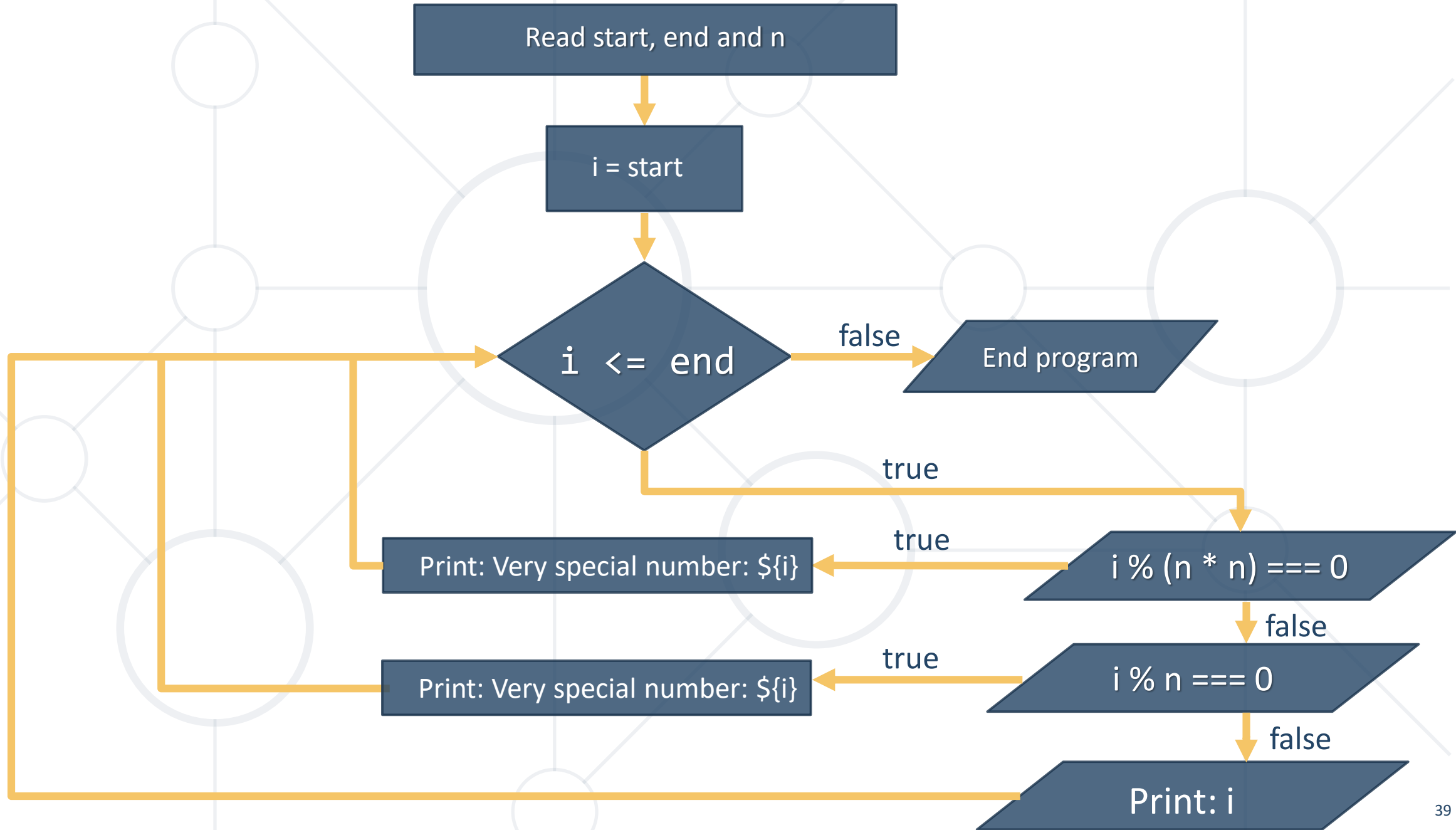
# Техники за използване на for-цикли

## Задачи с цикли

- Напишете програма, която:
  - Чете **start**, **end** и **n**
  - Ако число което се дели на  $n^2$  без остатък се отпечатва по следния начин: "**Special number: {съответното число}**"
  - Ако числото което се дели на **n** без остатък се отпечатва по следния начин: "**Special number: {съответното число}**"
  - Ако не се дели на **n** без остатък се отпечатва самото число

# Специални числа – примерен вход и изход





```
function verySpecialNumbers(start, end, n) {  
  for(let i = start; i <= end; i++) {  
    if (i % (n * n) === 0) {  
      console.log(`Very special number: ${i}`);  
    } else if (i % n === 0) {  
      console.log(`Special number: ${i}`);  
    } else {  
      console.log(i);  
    }  
  }  
}
```

1 3 7



- Напишете програма, която:
  - Прочита цяло число  $n$  от потребителя
  - Проверява дали числото е четно или нечетно
  - Принтира всички числа(четни или нечетни) до  $n$  по следния начин: "Current number: { i }. Cube: { i }"

# Число на трета степен - примерен вход и изход

- Примерен вход и изход:

5



Current number: 1. Cube: 1  
Current number: 3. Cube: 27  
Current number: 5. Cube: 125

6



Current number: 2. Cube: 8  
Current number: 4. Cube: 64  
Current number: 6. Cube: 216

# Число на трета степен - решение

```
function numberOnThirdPower(n) {  
    let start = 0;  
  
    if (n % 2 === 0) {  
        start = 2;  
    } else {  
        start = 1;  
    }  
  
    for (let i = start; i <= n; i += 2) {  
        console.log(`Current number: ${i}. Cube: ${i * i * i}`);  
    }  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/2321#8>



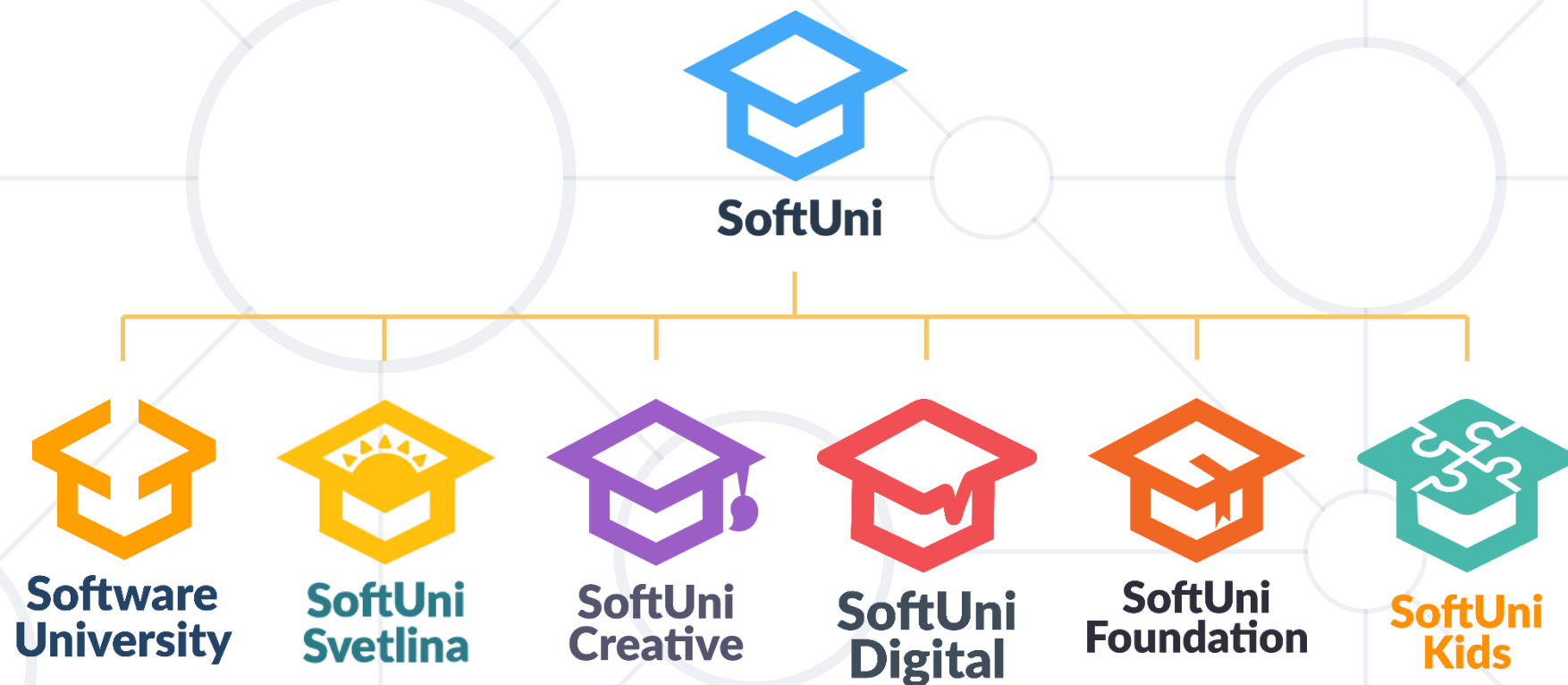
# **По-сложни задачи с цикли**

## **Решаване на задачи в клас (лаб)**

- Можем да повтаряме блок код с **for**-цикъл
- Цикли със стъпка
  - Цикли с намаляваща стъпка
- Можем да вземем символ по индекс от текст



# Въпроси?



# SoftUni Diamond Partners



**XS**software



**SBTech**  
*we know sports*



telenor



**SoftwareGroup**  
*doing it right*

**NETPEAK**



**SmartIT**



**Postbank**

*Решения за твоето утре*



**INDEAVR**

*Serving the high achievers*



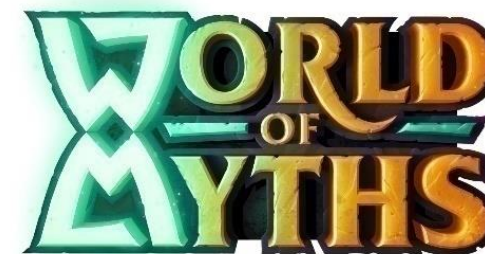
**INFRAGISTICS®**



**STEMO®**  
*Computer Systems & Software*

**SUPERHOSTING.BG**

# SoftUni Organizational Partners





- 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 C#" от Светлин Наков и колектив с лиценз CC-BY-SA

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

