

Повторения (цикли)

Прости повторения с While-цикъл



СофтУни

Преподавателски екип



SoftUni

Софтуерен университет
<https://softuni.bg>

1. Преговор
2. Масиви – четене от масив
3. Повторения (цикли)
 - **while**-цикъл – конструкция
 - **Безкраен while-цикъл**
 - Прекъсване на цикъл
 - Продължаване на цикъл





Преговор

1. Какъв ще е резултатът от изпълнението на следния код:

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

123

Infinite loop

111

Compile time
error

2. Какъв ще е резултатът от изпълнението на следния код:

```
for( ; ; ) {  
    console.log("SoftUni");  
}
```

Infinite loop

Runtime error

Compile
time error

SoftUni

3. Какъв ще е резултатът от изпълнението на следния код:

```
for (let i = 0; i < 2; i += 0.5) {  
  console.log(i + ", ");  
}
```

0, 1

0, 0.5, 1, 1.5

Compile time
error

Infinite loop

4. Какъв ще е резултатът от изпълнението на следния код:

```
let text = "input";  
console.log(text[0]);
```

input

i

Compile time
error

text

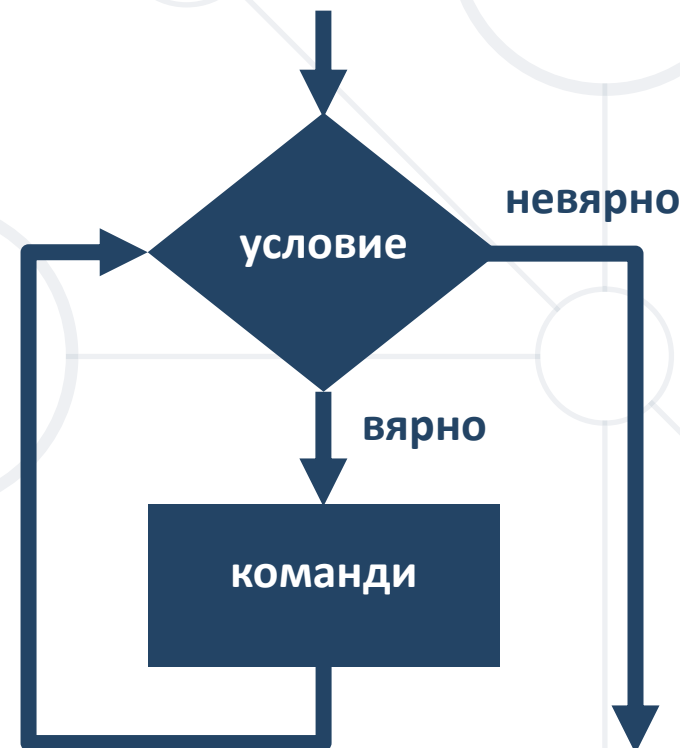
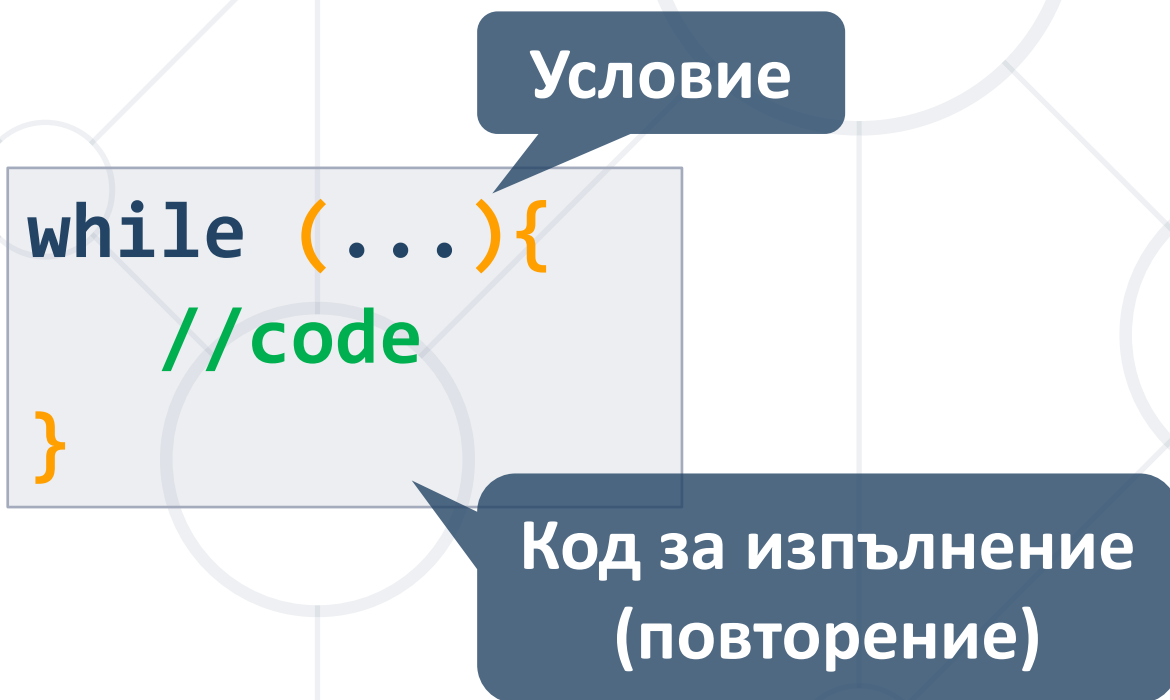


while

while-цикъл

Повторения (цикли) – While-цикъл

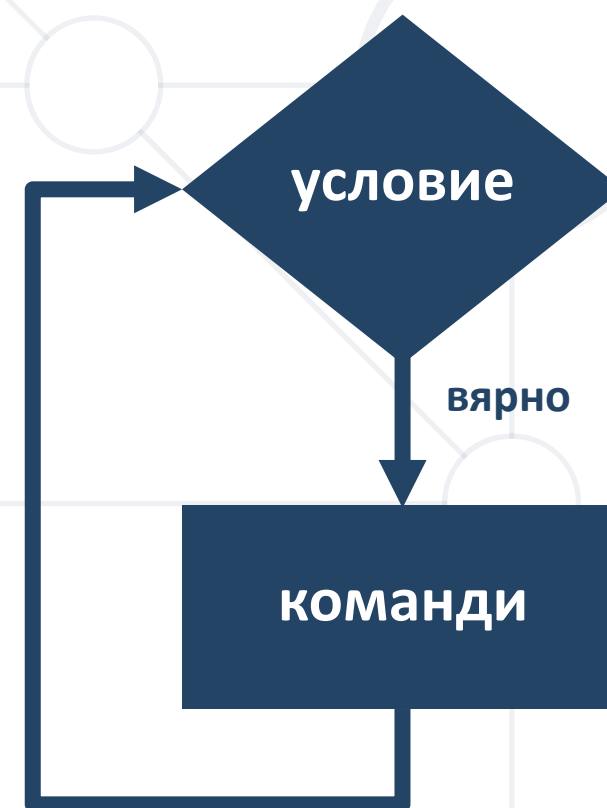
- В програмирането често се налага да изпълним блок с команди няколко пъти
 - За целта използваме цикли – **while**, **for** и други



- Безкраен цикъл – повтаряне на блок от код безкраен брой пъти:

Условието е винаги
вярно

```
while (true) {  
    console.log("Infinite loop");  
}
```





Прекъсване чрез оператор `break`

Безкрайни цикли

- Оператор **break** – прекъсва цикъла
- Не може да съществува самостоятелно извън цикъл

```
while (true) {  
    console.log("Infinite loop");  
    if (...) {  
        break;  
    }  
}
```

Условие за прекъсване на
цикъла

while-цикъл – пример

Условие за прекратяване
на повторението

```
let a = 5;  
while (true) {  
  if (a > 10) {  
    break;  
  }  
  console.log("a = " + a);  
  a++;  
}
```



```
a = 5  
a = 6  
a = 7  
a = 8  
a = 9  
a = 10
```

while-цикъл – пример



```
let index = 0;  
while (true) {  
  let str = input[index];  
  index++;  
  if (str === "Stop") {  
    break;  
  }  
  
  console.log("Infinite loop");  
}
```

Условие за прекратяване
на повторението

- Напишете функция, която:
 - Получава масив от стрингове
 - Приключва четенето когато получи стринга "Stop"
- Примерен вход и изход:



```
Nakov  
SoftUni  
Sofia  
Bulgaria  
SomeText  
Stop  
AfterStop
```

```
Nakov  
SoftUni  
Sofia  
Bulgaria  
SomeText
```

```
let index = 0;
while (true) {
  let str = input[index];
  index++;
  if (str === "Stop") {
    break;
  }
  console.log(str);
}
```


A background network diagram consisting of a grid of light gray lines intersecting at various points. At these intersections, there are several circles of different sizes, some of which are also light gray. The overall pattern resembles a stylized molecular structure or a network graph.

`while`

`while`-цикъл

Повторение докато е вярно дадено условие

while-цикъл – пример

Условие за прекратяване
на повторението

```
let a = 5;  
  
while (a <= 10) {  
  console.log("a = " + a);  
  a++;  
}
```



```
a = 5  
a = 6  
a = 7  
a = 8  
a = 9  
a = 10
```

while-цикъл – пример



```
let index = 0;  
let str = input[index];  
while (str !== "Stop") {  
    console.log("Infinite loop");  
    index++;  
    str = input[index];  
}
```

Условие за
прекратяване на
повторението

- Напишете функция, която:
 - Първоначално прочита потребителско име и парола на потребителски профил
 - Прочита парола за вход и проверява дали е коректна
 - При:
 - Невалидна парола, прочита нова
 - При коректно въведена парола, прекратява изпълнение

```
function password(input) {  
    let username = input[0];  
    let password = input[1];  
    let data = input[2];  
    let index = 3;  
    while (data !== password) {  
        data = input[index];  
        index++;  
    }  
    console.log(`Welcome ${username}!`);  
}
```

- Напишете функция, която:
 - Чете цели числа
 - Приключва четенето когато получи сума равна на първоначално въведеното число
 - Извежда сумата на всички прочетени числа
- Примерен вход и изход:

100
10
20
30
45



105

20
1
2
3
4
5
6



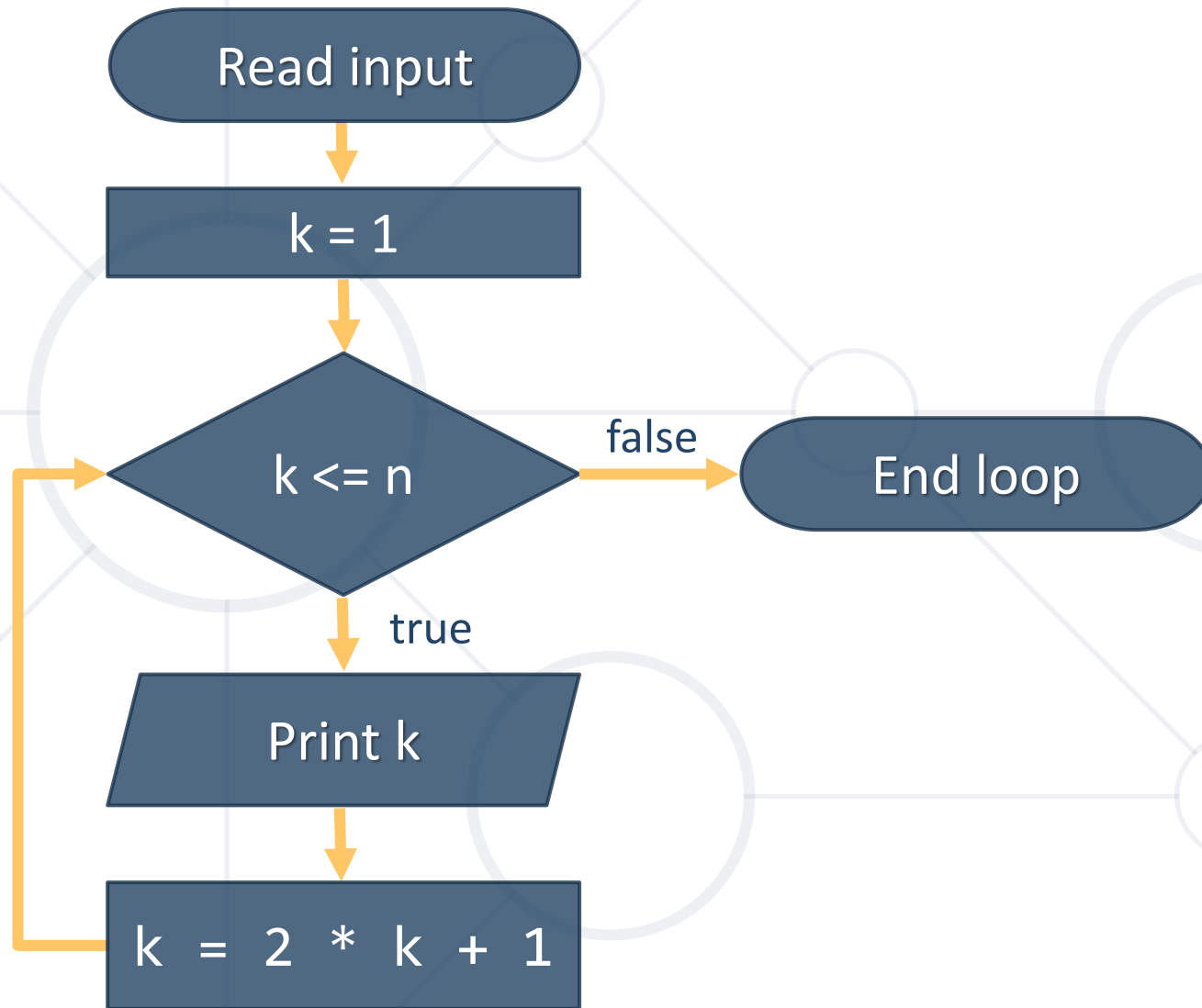
21

```
let n = Number(input[0]);  
let sum = 0;  
let index = 1;  
  
while (sum < n) {  
    let currentNum = Number(input[index]);  
    sum += currentNum;  
    index++;  
}  
console.log(sum);
```

Редица числа $2K+1$ – условие

- Напишете функция, 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, $(1 * 2) + 1 = 3$, $(3 * 2) + 1 = 7$, $(7 * 2) + 1 = 15$...



Редица числа $2K+1$ – решение

```
function sequence(input) {  
  let number = Number(input[0]);  
  let k = 1;  
  while (k <= number) {  
    console.log(k);  
    k = k * 2 + 1;  
  }  
}
```

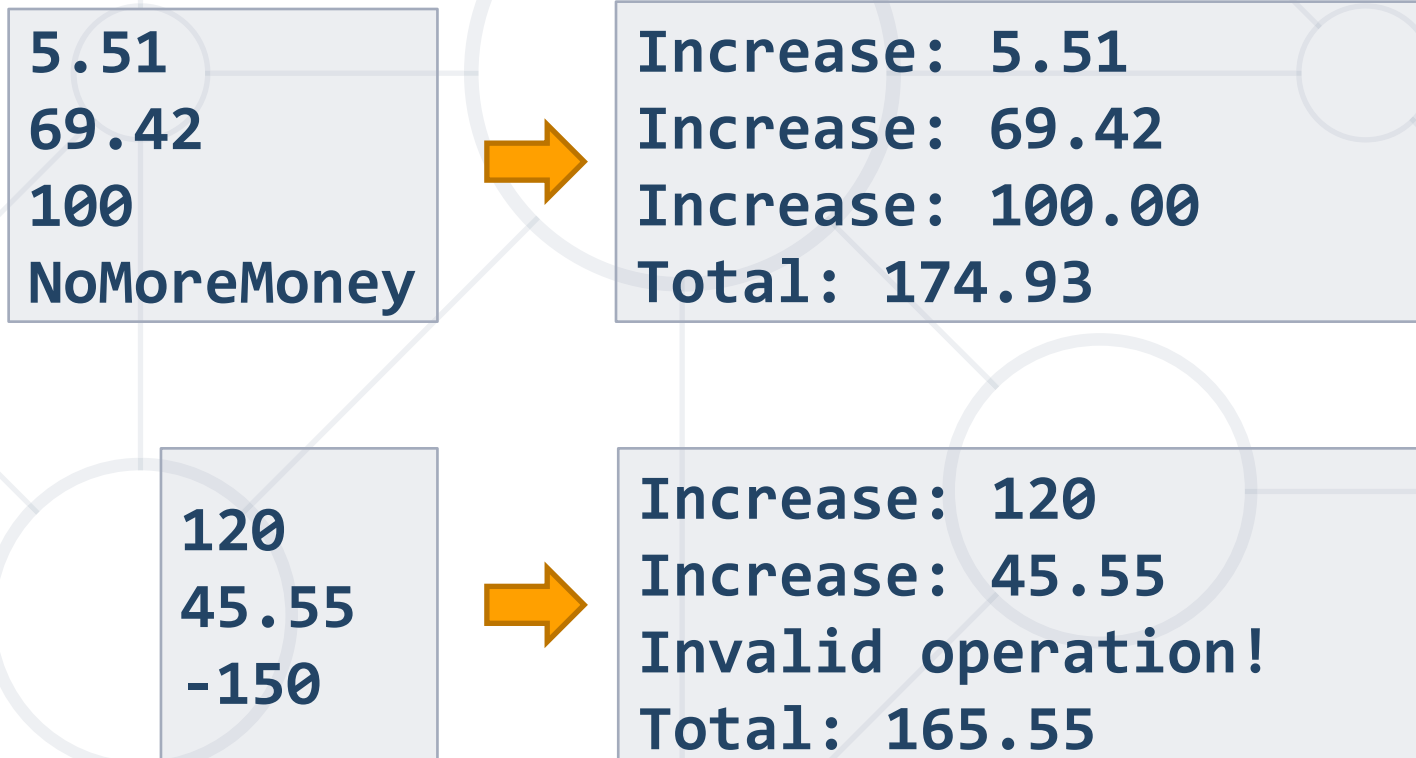
Повторение докато е в
сила условието $k \leq n$

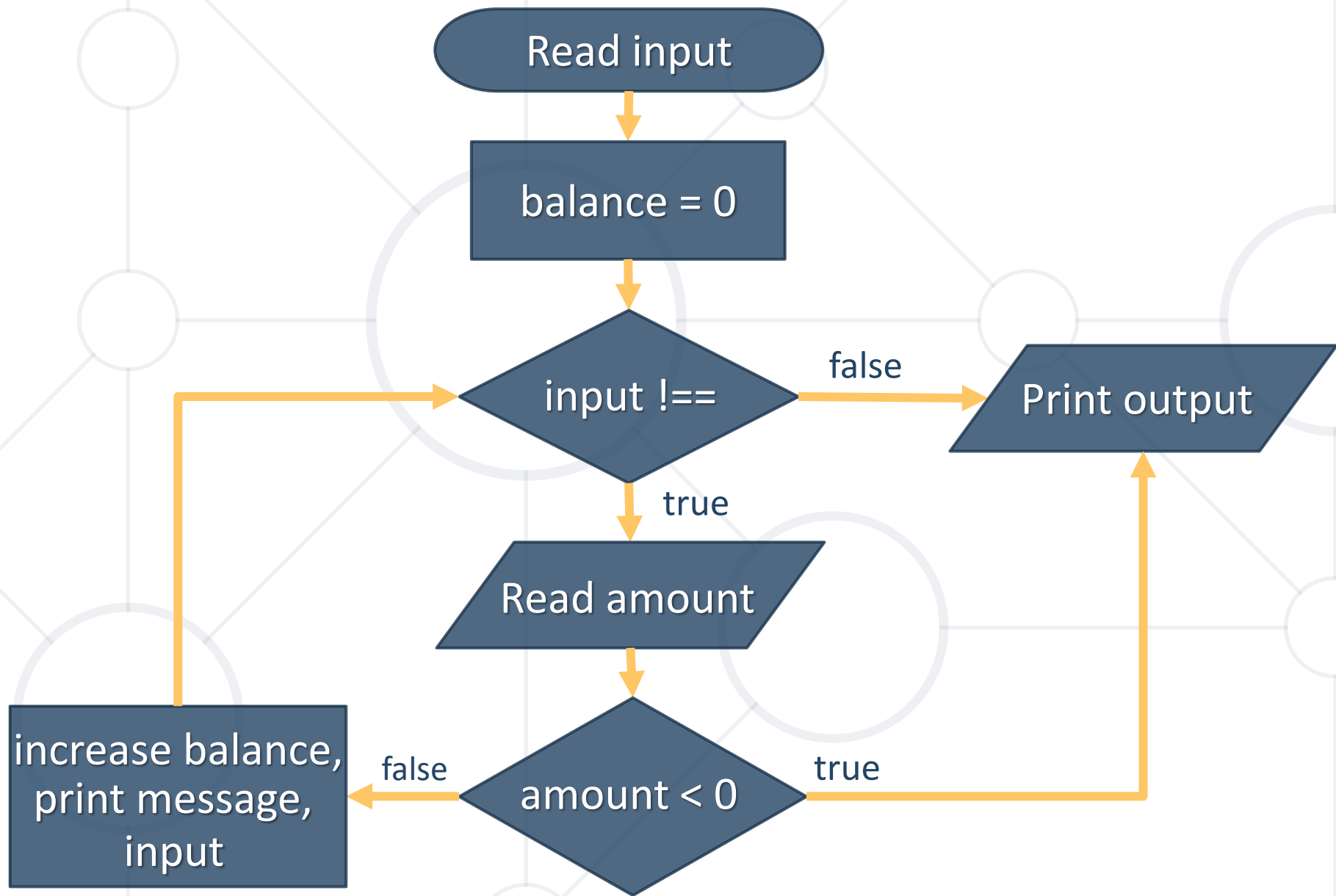
Тестване на решението: <https://judge.softuni.bg/Contests/Index/2407#3>

- Напишете функция, която:
 - Чете n – на брой числа, които представляват вноски по банкова сметка до получаване на командата **"NoMoreMoney"**
 - При всяка вноска принтира:
"Increase: {сумата}"
 - Ако се въведе отрицателно число да се изпише **"Invalid operation!"** и програмата да приключи
 - Накрая на програмата трябва да се изпише:
"Total: {общата сума в сметката}"

Баланс на сметка – условие(3)

- Примерен вход и изход:





Баланс на сметка – решение

```
function accountBalance(input) {  
  let deposit = input[0];  
  let balance = 0;  
  let index = 1;  
  while (deposit !== "NoMoreMoney") {  
    let amount = Number(deposit);  
    if (amount < 0) { //TODO: Print message and exit the loop  
    }  
    balance += amount;  
    console.log(`Increase: ${amount.toFixed(2)}`);  
    deposit = input[index];  
    index++;  
  }  
  console.log(`Total: ${balance.toFixed(2)}`);  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Index/2407#4>

Най-голямо число – пример

- Напишете функция, която:
 - Прочита **n** последователни пъти числа, докато получи команда **"Stop"**
 - Намира най-голямото измежду тях
- Примерен вход и изход:

100
99
80
70
Stop



100

-10
20
-30
Stop



20

45
-20
7
99
Stop



99

5

3

Най-голямо число – решение

```
let inputElement = input[0];
let index = 1;
let max = Number.MIN_SAFE_INTEGER;
while (inputElement !== "Stop") {
    let num = Number(inputElement);
    if (num > max) {
        max = num;
    }
    inputElement = input[index];
    index++;
}
console.log(max);
```

Тестване на решението: <https://judge.softuni.bg/Contests/Index/2407#5>

Най-малко число – условие

- Напишете функция, която:
 - Прочита **n** последователни пъти числа, докато получи команда **"Stop"**
 - Намира най-малкото измежду тях
- Примерен вход и изход:

100
99
80
70
Stop



70

-10
20
-30
Stop



-30

45
-20
7
99
Stop



-20



```
let inputElement = input[0];  
let index = 1;  
let min = Number.MAX_SAFE_INTEGER;  
while (inputElement !== "Stop") {  
    //TODO: Use logic similar  
    to the previous problem  
}
```

- Оператор **continue** – преминава към следващата итерация на цикъла

```
let i = 0;  
while (i < 10) {  
  if (i % 2 === 0) {  
    i++;  
    continue;  
  }  
  console.log(i);  
  i++;  
}
```



```
"C:\Program  
1  
3  
5  
7  
9
```

- Напишете функция, която:
 - Изчислява **средната оценка** на ученик от цялото му обучение
 - Ако годишната му оценка е:
 - **≥ 4.00** , ученикът преминава е следващия клас
 - **< 4.00** , той ще повтори класа
 - Ако бъде скъсан повече от един път, той бива изключен и програмата приключва. Отпечатва се името и в кой клас е изключен:
"{име на ученика} **has been excluded at** {класа, в който е бил изключен} **grade**"
 - При **завършване** да се отпечата:
"{име на ученика} **graduated. Average grade:** {средната оценка от цялото обучение}"

Завършване – условие (2)

Gosho
5
5.5
6
5.43
5.5
6
5.55
5
6
6
5.43
5




Gosho graduated.
Average grade: 5.53

Mimi
5
6
5
6
5
6
6
2
3



Mimi has been
excluded at 8 grade

```
function graduation(input) {  
  let name = input[0];  
  let grades = 1;  
  let sum = 0;  
  let excluded = 0;  
  let index = 0;  
  while (grades <= 12) {  
    index++;  
    let grade = Number(input[index]);  
    if (grade < 4.00) {  
      // TODO: increase excluded count and break if is more than 1  
      continue;  
    }  
    // TODO: add grade to sum and increase grades count  
  }  
  let average = sum / 12;    //TODO: print the output  
}
```

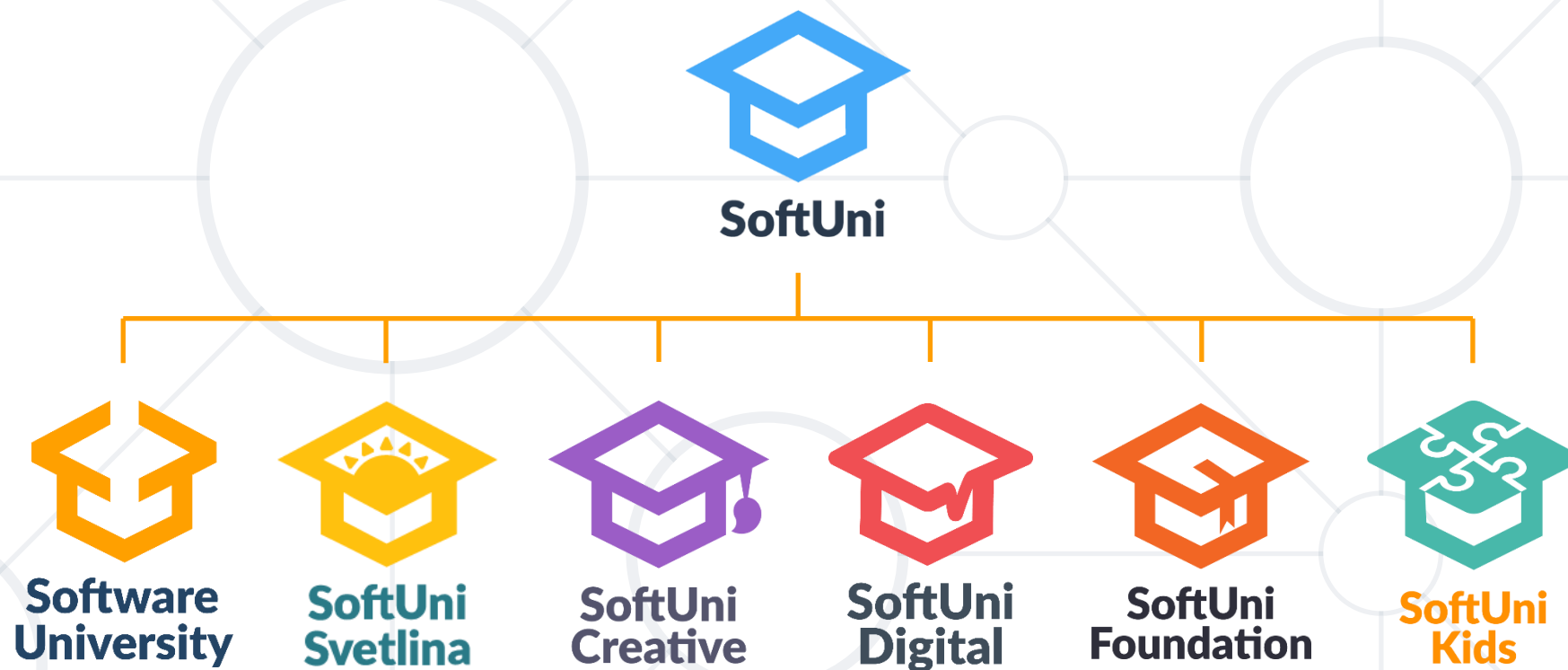


Тестване на решението: <https://judge.softuni.bg/Contests/Index/2407#7>

- Можем да повтаряме блок от код с **while**-цикъл
- Можем да прекъсваме цикли с оператора **break**
- Можем да преминем към следваща итерация с оператора **continue**



Въпроси?



- Този курс (презентации, примери, демонстрационен код, упражнения, домашни, видео и други активи) представлява **защитено авторско съдържание**
- Нерегламентирано копиране, разпространение или използване е незаконно
- © СофтУни – <https://softuni.org>
- © Софтуерен университет – <https://softuni.bg>



- Софтуерен университет – качествено образование, професия и работа за софтуерни инженери
 - softuni.bg
- Фондация "Софтуерен университет"
 - softuni.foundation
- Софтуерен университет @ Facebook
 - facebook.com/SoftwareUniversity
- Дискуссионни форуми на СофтУни
 - forum.softuni.bg



Software University

