

# По-сложни цикли



СофтУни

Преподавателски екип



SoftUni



Софтуерен университет  
<https://softuni.bg>

1. For цикъл с едно условие
2. Безкраен цикъл
3. Прекъсване на цикъл
4. Продължаване на цикъл



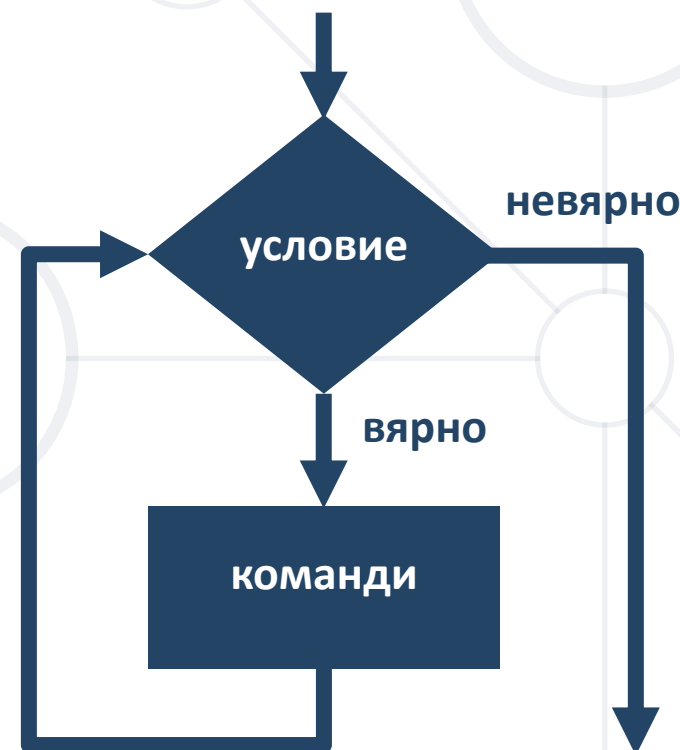
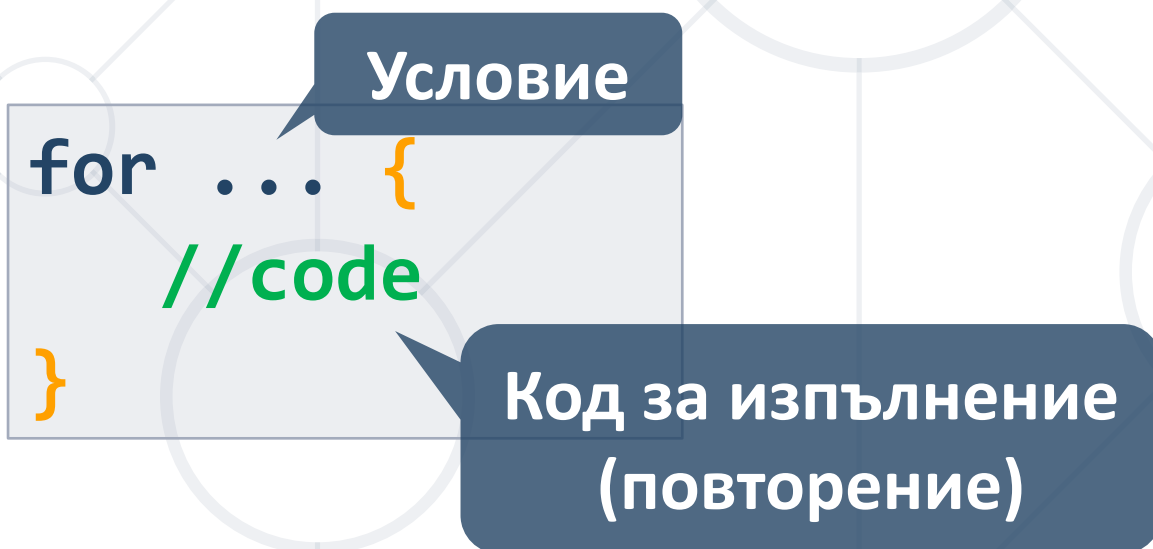


```
for ... {  
  
}
```

**For цикъл с едно условие**

# For цикъл с едно условие - конструкция

- В програмирането често се налага да изпълним блок с команди неопределен брой пъти
  - За целта използваме **цикъл, който зависи от условие**



# For цикъл с едно условие – пример

```
var a int = 5
for a <= 10 {
    fmt.Printf("a = %d\n", a)
    a++
}
```

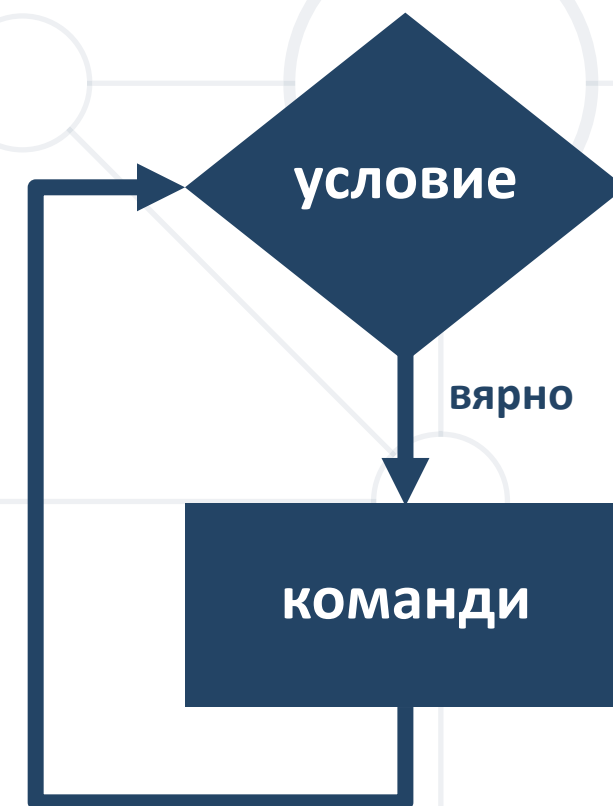


```
a = 5
a = 6
a = 7
a = 8
a = 9
a = 10
Press any key to continue . . .
```

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

Условието е винаги  
вярно

```
for true {  
    fmt.Println("Infinite loop")  
}
```





**Прекъсване чрез оператор `break`**

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

```
for true {  
    fmt.Println("Infinite loop")  
    if ... {  
        break  
    }  
}
```

Условие за прекъсване на  
цикъла



# Прекратяване на цикъл – пример



```
for true {  
    var input string  
    fmt.Scanln(&input)  
    if input == "Stop" {  
        break  
    }  
    fmt.Println("Infinite loop")  
}
```

Условие за  
прекратяване на  
повторението

- Напишете програма, която:
  - Чете от потребителя текст (низ)
  - Приключва четенето когато получи командата "Stop"
- Примерен вход и изход:



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

```
Nakov  
SoftUni  
Sofia  
Bulgaria  
SomeText
```

- Напишете програма, 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

- Напишете програма, която:
  - Чете от потребителя 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

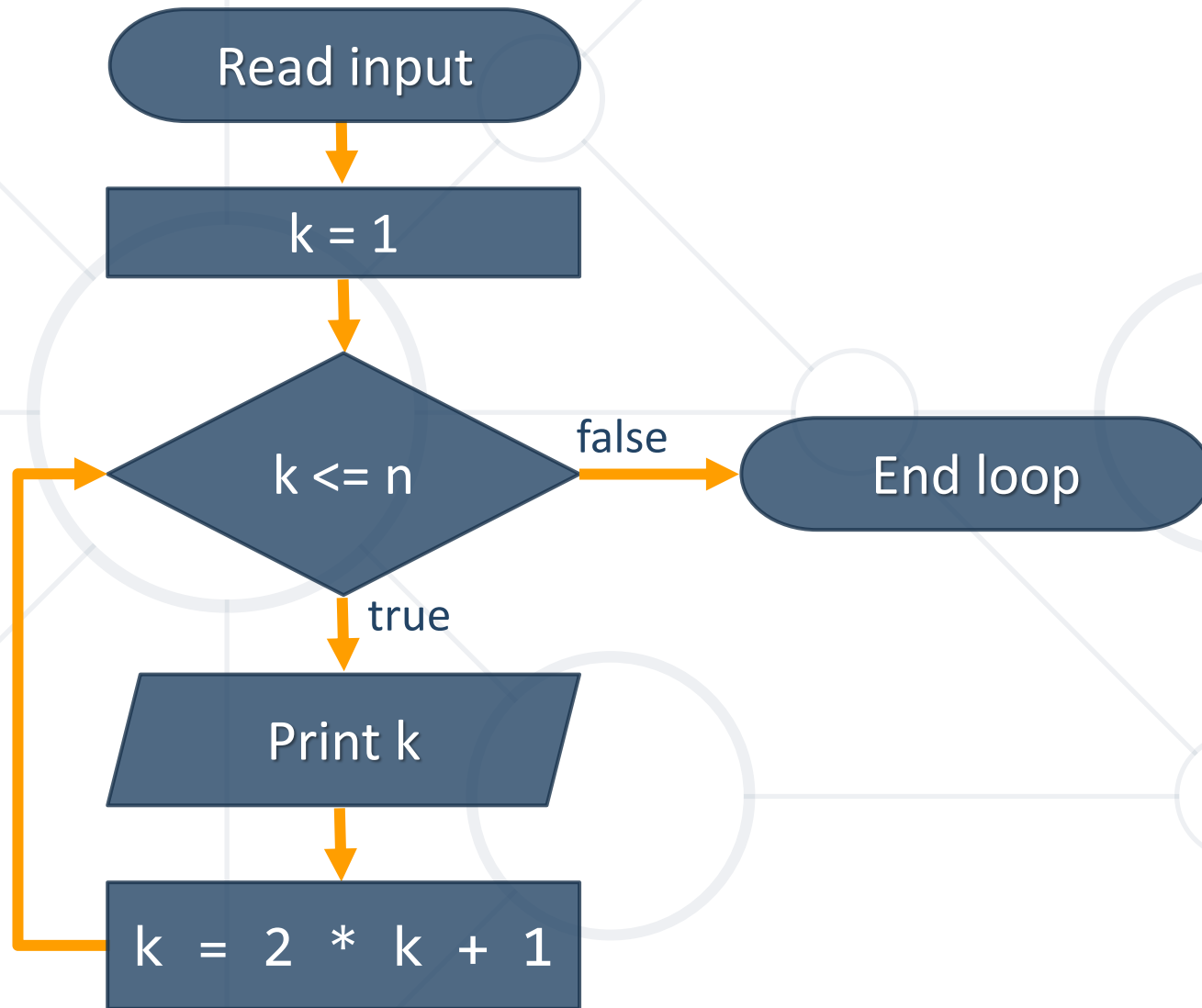


21

# Редица числа $2k + 1$ – условие

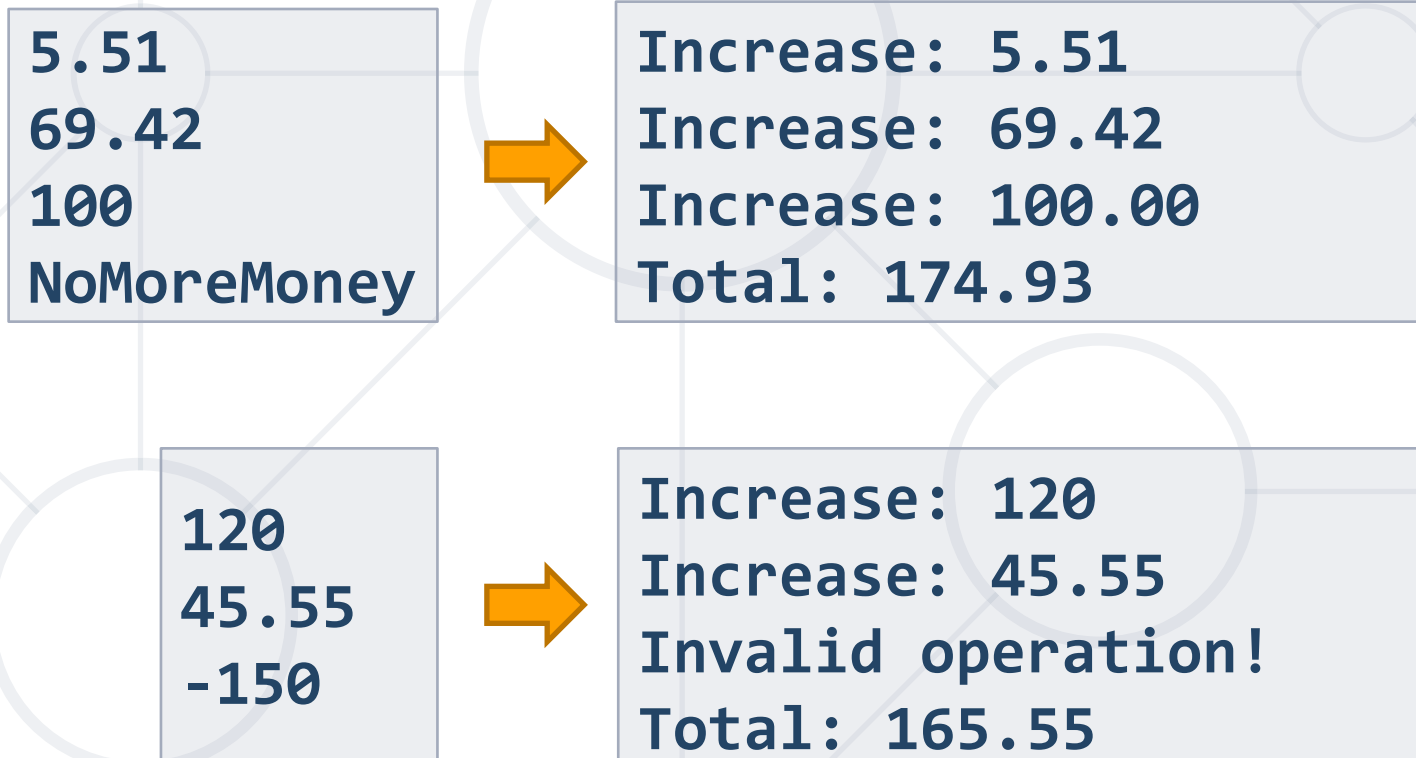
- Напишете програма, която:
  - Прочита цяло число  $n$
  - Отпечатва всички числа  $\leq n$  от редицата: 1, 3, 7, 15, 31, ...
  - Всяко следващо число е равно на **предишното** \* 2 + 1

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

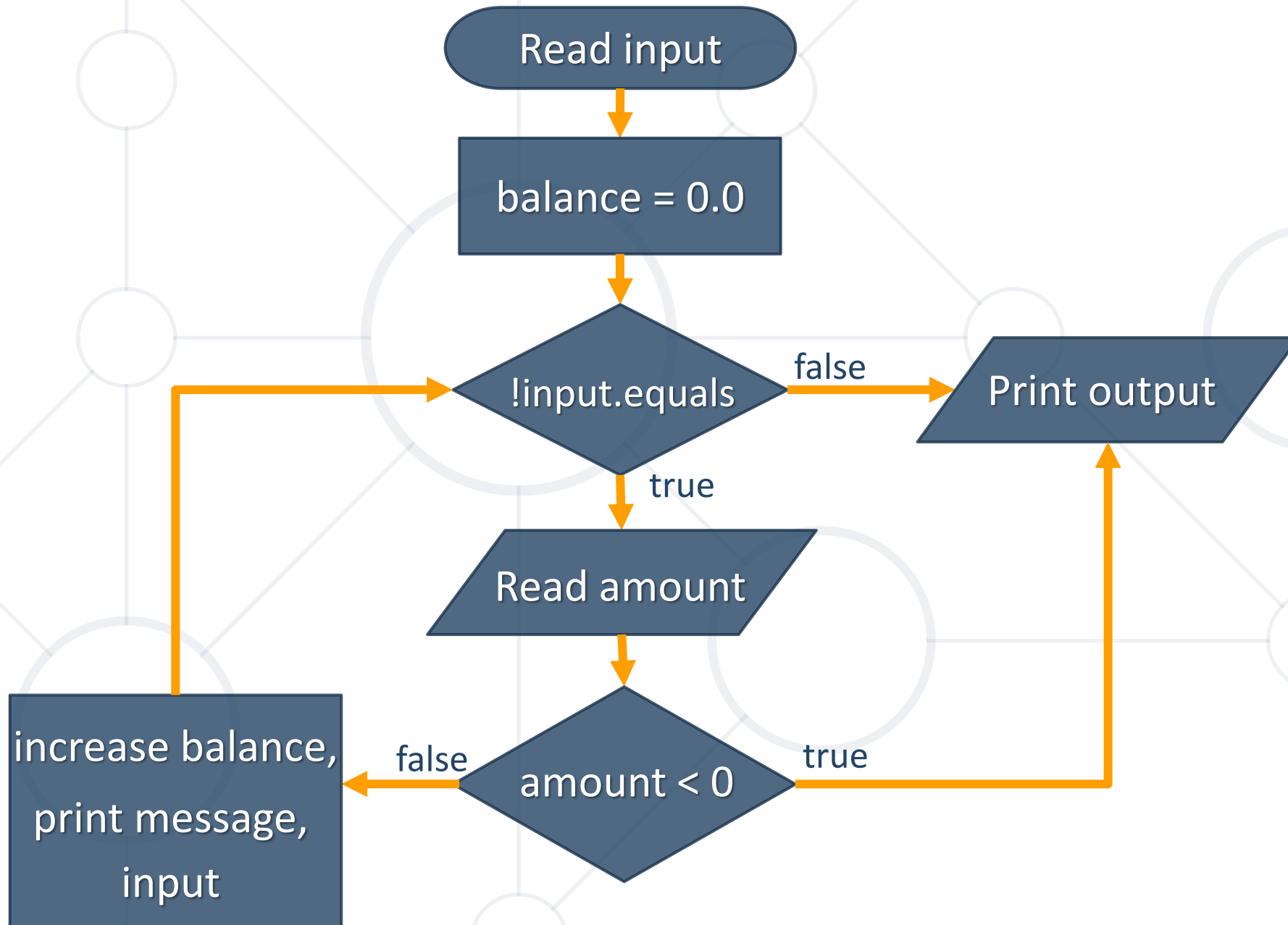


- Напишете програма, която:
  - Чете  $n$  – на брой числа, които представляват вноски по банкова сметка до получаване на командата **"NoMoreMoney"**
  - При всяка вноска принтира:  
**"Increase: {сумата}"**
  - Ако се въведе отрицателно число да се изпише **"Invalid operation!"** и програмата да приключи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

- Примерен вход и изход:







# Най-голямо число – условие

- Напишете програма, 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-малко число – условие

- Напишете програма, 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



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

```
var i int = 0
for i < 10 {
    if i % 2 == 0 {
        i++
        continue
    }
    fmt.Println(i)
    i++
}
```

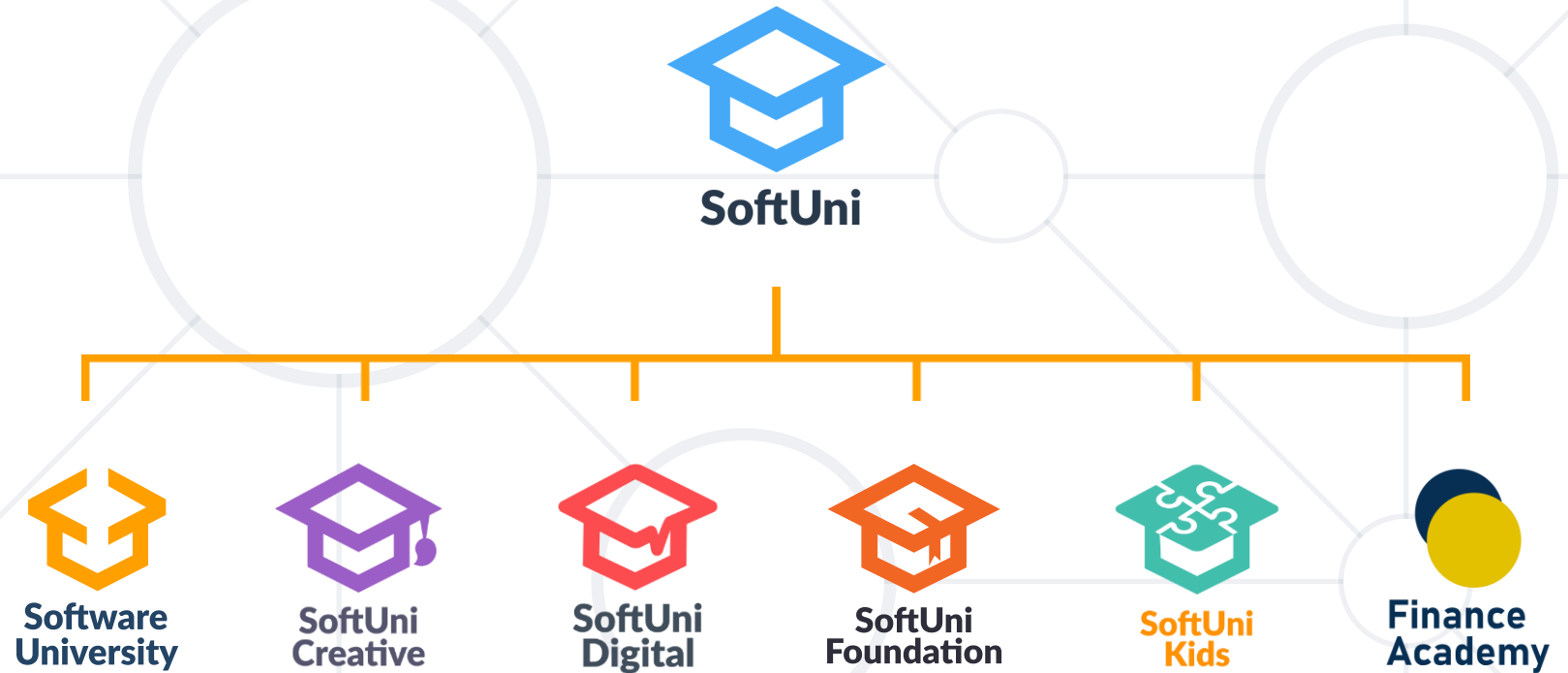


```
"C:\Program
1
3
5
7
9
```

- Можем да повтаряме блок от код с **for** 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](http://softuni.bg)
- Фондация "Софтуерен университет"
  - [softuni.foundation](http://softuni.foundation)
- Софтуерен университет @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

