

# Повторения (цикли)

Прости повторения с `while`-цикъл



СофтУни

Преподавателски екип



Software  
University



SoftUni  
Foundation



Софтуерен университет  
<http://softuni.bg>

# Имате въпроси?

sli.do

**#pb-feb**

1. Преговор
2. Увеличаване и намаляване на стойността на променливи
3. Повторения (цикли)
  - **while** - цикъл – конструкция
4. Безкрайни цикли
  - Прекъсване на цикли





**Преговор**

1. Кои числа ще се изпринтират като резултат от изпълнението на следния код:

```
for i in range(2, 10, 7):  
    print(i)
```

2 и 9

2... 10

2... 7

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

```
for i in range(5, 0):  
    print(i)
```

No output

Infinite loop

5... 0

0... 5

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

```
for i in range(97, 100):  
    print(chr(i))
```

a b c

Compile time  
error

97 98 99

Infinite loop

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

```
for i in range(0, 2, 0.5):  
    print(i)
```

0, 0.5, 1, 1.5

0, 1

Compile time  
error

Infinite loop



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.

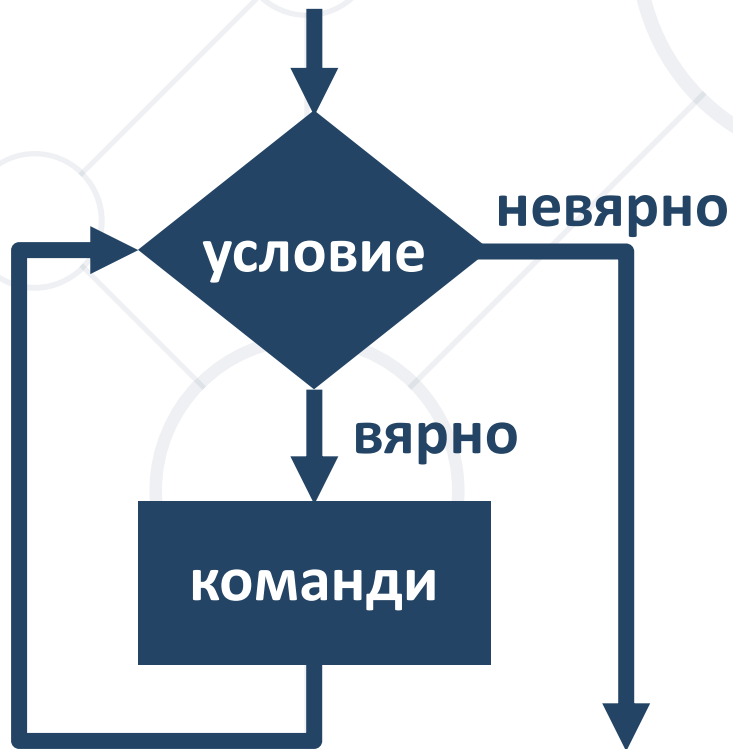
**while**

**while-цикъл**

**Повторение докато е вярно дадено условие**

# Повторения (цикли) – while-цикъл

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



```
while ... :  
    //code
```

Условие

Код за изпълнение  
(повторение)

# while-цикъл – пример

```
a = 5
```

```
while a <= 10:  
    print("a = " + str(a))  
    a += 1
```

Условие за прекратяване  
на повторението



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

```
Process finished with exit code 0
```

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



```
graph LR; Input["Nakov  
SoftUni  
Sofia  
Bulgaria  
SomeText  
Stop"] --> Output["Няма изход"]
```

Nakov  
SoftUni  
Sofia  
Bulgaria  
SomeText  
Stop

*Няма изход*

```
inpt = input()
while inpt != "Stop":
    inpt = input()
```

- 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

```
username = input()
password = input()

data = input()
while data != password:
    data = input()

print(f"Welcome: {username}!")
```

Тестване на решението: <https://judge.softuni.bg/Contests/Compete/Index/1014#1>

- Напишете програма, която:
  - Чете от потребителя цели числа
  - Приключва четенето когато получи командата **"Stop"**
  - Извежда сумата на всички прочетени числа
- Примерен вход и изход:

10  
20  
30  
45  
Stop



105

1  
2  
3  
4  
5  
6  
Stop



11



```
inpt = input()
sum = 0

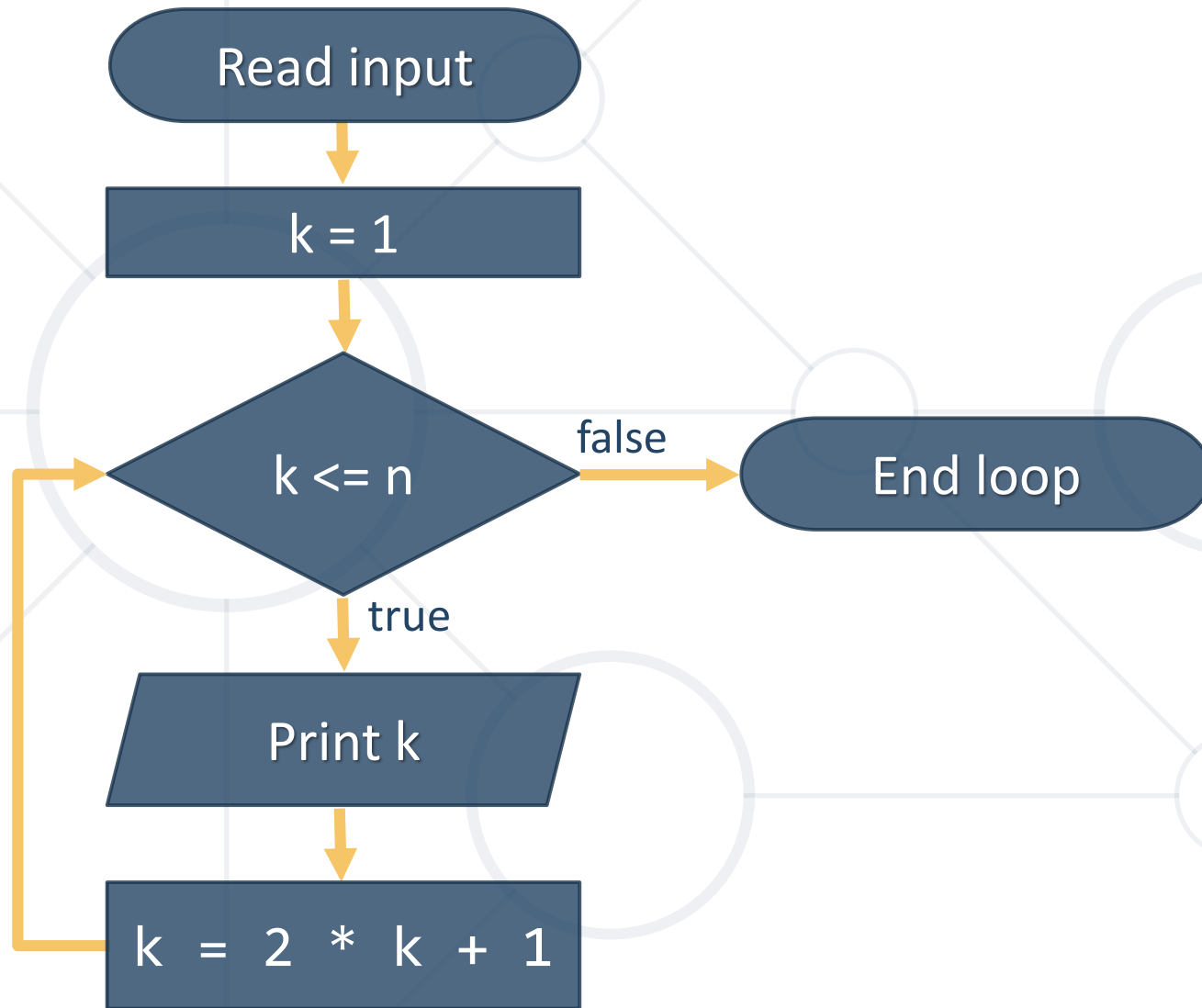
while inpt != "Stop":
    currentNum = int(inpt)
    sum += currentNum
    inpt = input()

print(sum)
```

# 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** ...



# Редица числа $2k+1$ - решение

```
number = int(input())  
k = 1  
while k <= number:  
    print(k)  
    k = k * 2 + 1
```


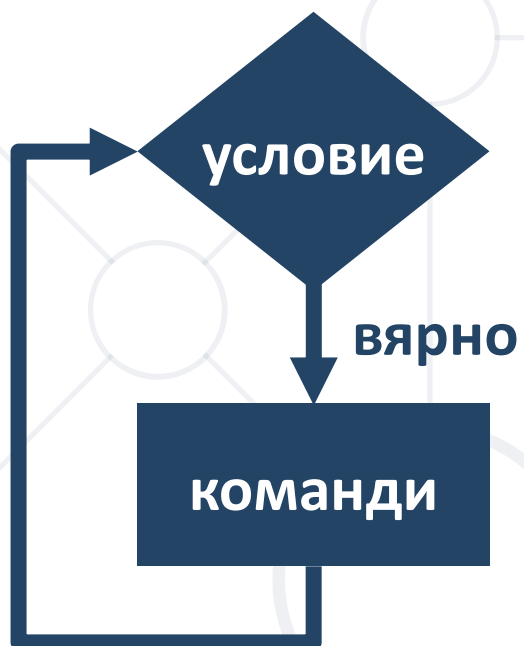
Повторение докато е в  
сила условието  $k \leq n$



# Прекъсване чрез оператор `break`

## Безкрайни цикли

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



Условието е винаги  
вярно

```
while True:  
    print("Infinite loop")
```

- Оператор **break** – прекъсва цикъла

```
while True:  
    print("Infinite loop")  
    if ...:  
        break
```

Условие за прекъсване на  
цикъла

**BREAK**

- Напишете програма, която:
  - Чете  $n$  – на брой числа, които представляват вноски по банкова сметка
  - 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 balance: {общата сума в сметката}"**



# Баланс на сметка - условие(3)

- Примерен вход и изход:

3  
5.51  
69.42  
100

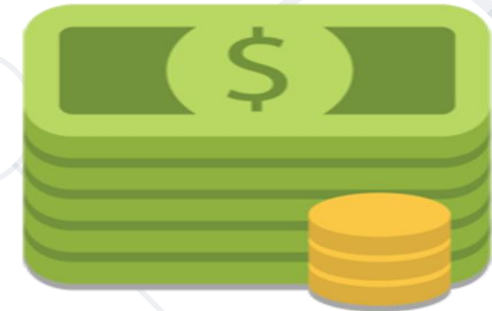


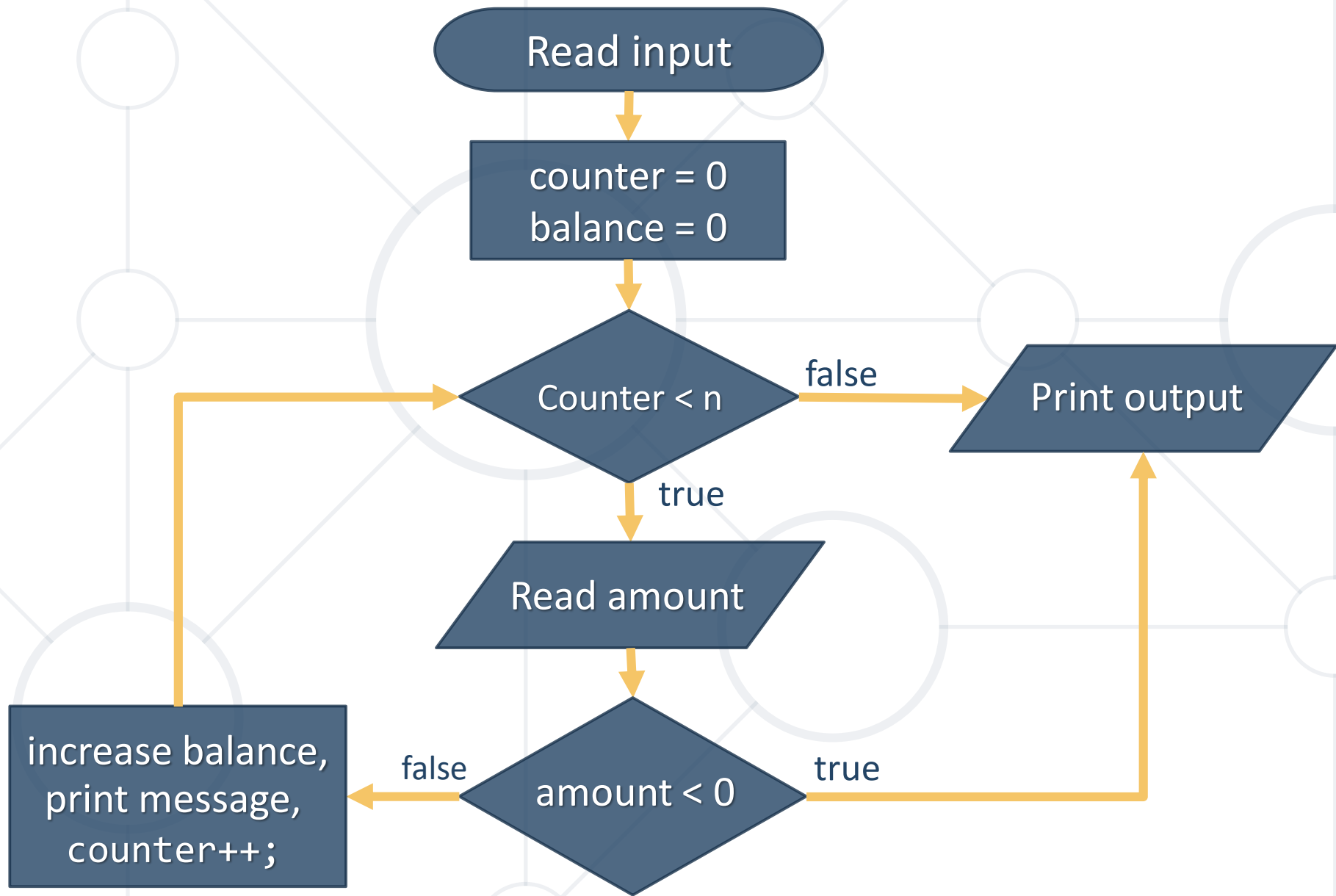
Increase: 5.51  
Increase: 69.42  
Increase: 100.00  
Total: 174.93

5  
120  
45.55  
-150



Increase: 120.00  
Increase: 45.55  
Invalid operation!  
Total: 165.55

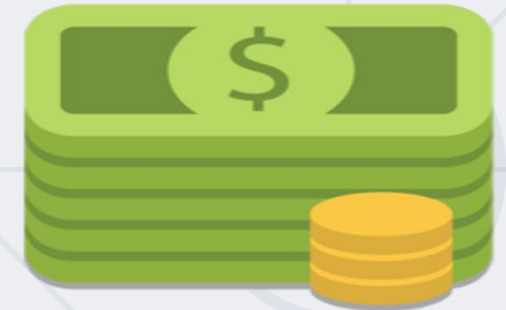




# Баланс на сметка - решение

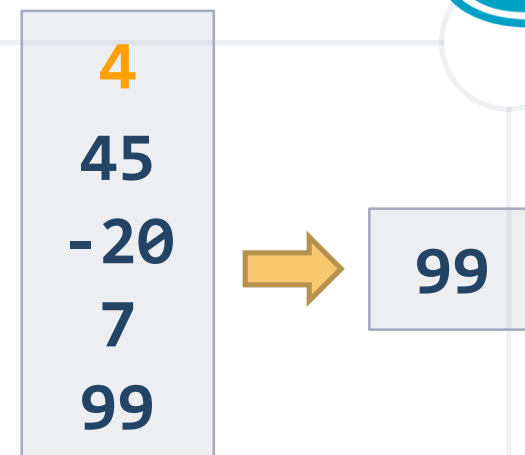
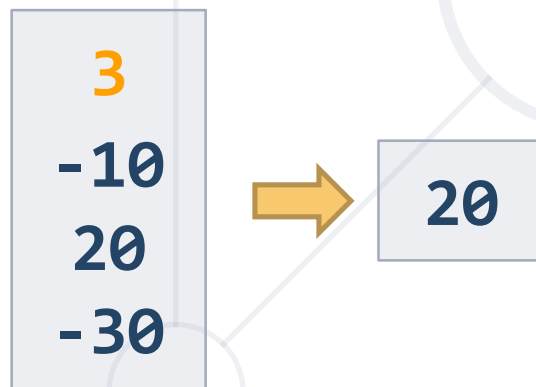
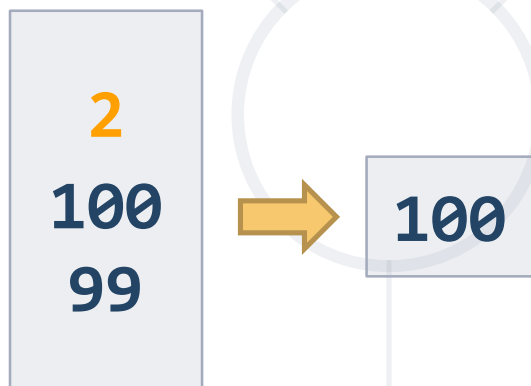
```
n = int(input())
counter = 0
balance = 0.0
while counter < n:
    amount = float(input())
    if amount < 0:
        # TODO: Print output and exit the loop

    balance += amount
    print("Increase: {0:.2f}".format(amount))
    counter += 1
print("Total: {0:.2f}".format(balance))
```



# Най-голямо число - пример

- Напишете програма, която:
  - Прочита цяло число(**n**) от потребителя
  - Прочита **n** последователни пъти числа
  - Намира най-голямото измежду тях
- Примерен вход и изход:



# Най-голямо число - решение

```
n = int(input())
count = 0
max = -1000000000000000

while count < n:
    num =
    int(input())
    count += 1

    if num > max:
        max = num

print(max)
```

# Най-малко число - условие

- Напишете програма, която:
  - Получава цяло число(*n*) от потребителя
  - Взима *n* последователни пъти числа
  - Намира най-малкото измежду тях
- Примерен вход и изход:

2  
99  
100



99

3  
-10  
20  
-30



-30

4  
45  
-20  
7  
99



-20



```
n = int(input())
counter = 0
min = 1000000000000000
while counter < n:
    #TODO: Use logic similar to the previous problem
print(max)
```



- Напишете програма, която:
  - Изчислява **средната оценка** на ученик от цялото му обучение
  - Ако годишната му оценка е:
    - **$\geq 4.00$** , ученикът преминава е следващия клас
    - **$< 4.00$** , той повтаря класа
  - При **завършване** да се отпечата:  
"{име на ученика} graduated. Average grade: {средната оценка от цялото обучение}"

# Завършване - условие (2)

- Примерен вход и изход:

Pesho  
4  
5.5  
6  
5.43  
4.5  
6  
5.55  
5  
6  
6  
5.43  
5



Pesho graduated.  
Average grade: 5.37

Ani  
5  
5.32  
6  
5.43  
5  
6  
5.5  
4.55  
5  
6  
5.56  
6



Ani graduated.  
Average grade: 5.45

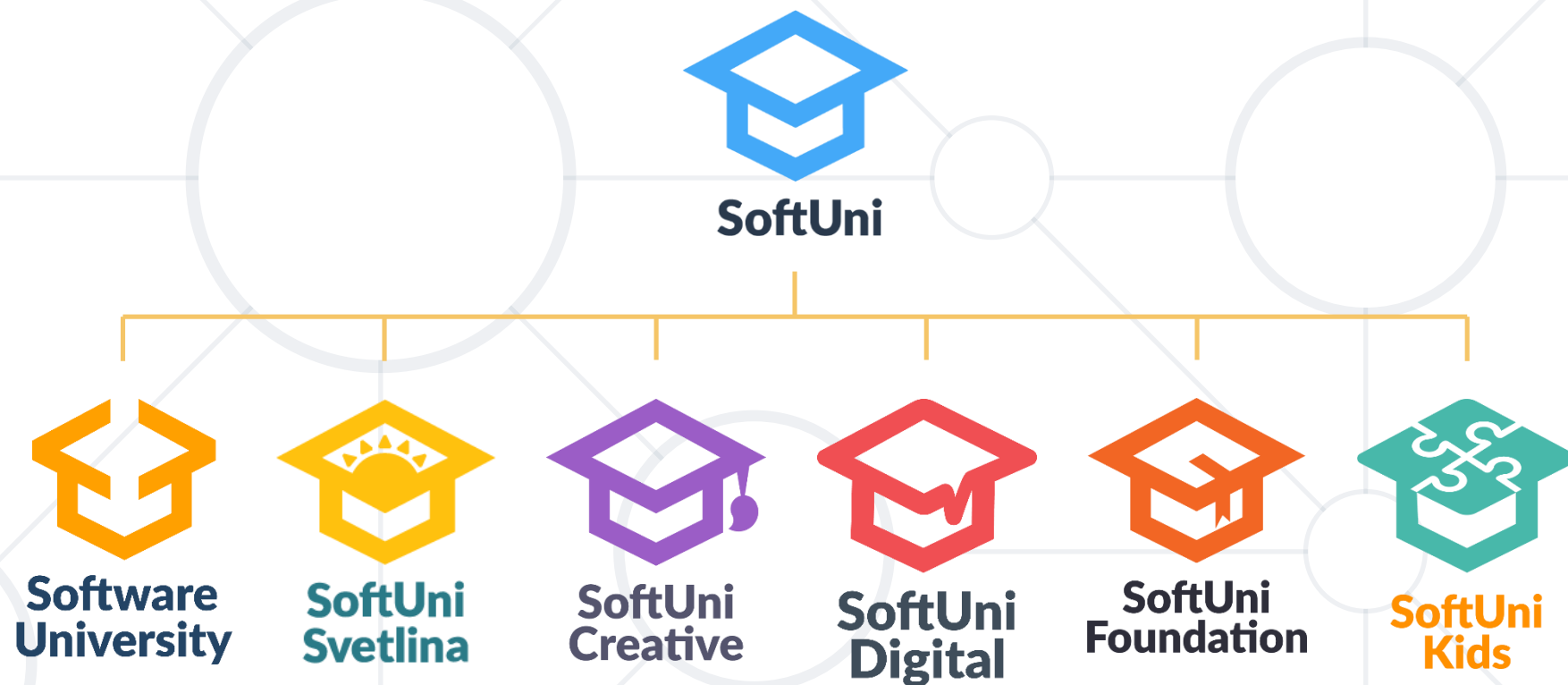
```
name = input()
grades = 1.0
sum_grades = 0.0
while grades <= 12:
    grade = float(input())
    if grade >= 4.00:
        #TODO: add grade to sum and increase grades count
average = sum_grades / 12
#TODO: print the output
```



- Използваме **while** - цикли, за да повтаряме действие, докато е в сила дадено условие
- Можем да прекъсваме циклите с оператора **break**



# Въпроси?



# SoftUni Diamond Partners



**XS**software



**SBTech**  
*we know sports*



telenor



**SoftwareGroup**  
*doing it right*

**NETPEAK**



**SmartIT**



**Postbank**

*Решения за твоето утре*

**SUPER  
HOSTING  
.BG**

**INDEAVR**

*Serving the high achievers*



**INFRAGISTICS®**



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



# SoftUni Organizational Partners

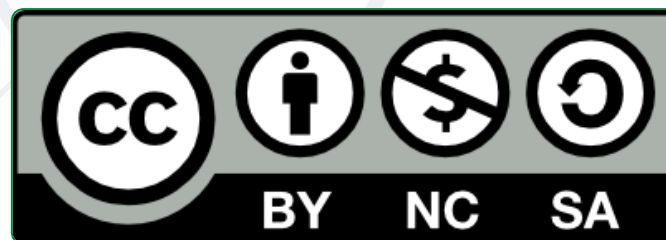


OneBit  
SOFTWARE



WORLD  
OF  
MYTHS

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



# Обучения в СофтУни

- 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)



**Software  
University**

