

# Повторения (цикли)

Прости повторения с While цикъл



СофтУни

Преподавателски екип



SoftUni

Софтуерен университет  
<https://softuni.bg>

1. Преговор

2. While цикъл

- Конструкция
- Безкраен while цикъл
- Прекъсване на цикъл
- Продължаване на цикъл





**Преговор**

1. Какъв ще е резултатът 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

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, 1

0, 0.5, 1, 1.5

TypeError

Infinite loop



**while**

**While цикъл**



# While цикъл - конструкция

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



# While цикъл – пример

Условие за прекратяване  
на повторението


```
a = 5
```

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



```
a = 5  
a = 6  
a = 7  
a = 8  
a = 9  
a = 10  
Process finished with exit code 0
```

# While цикъл – пример



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

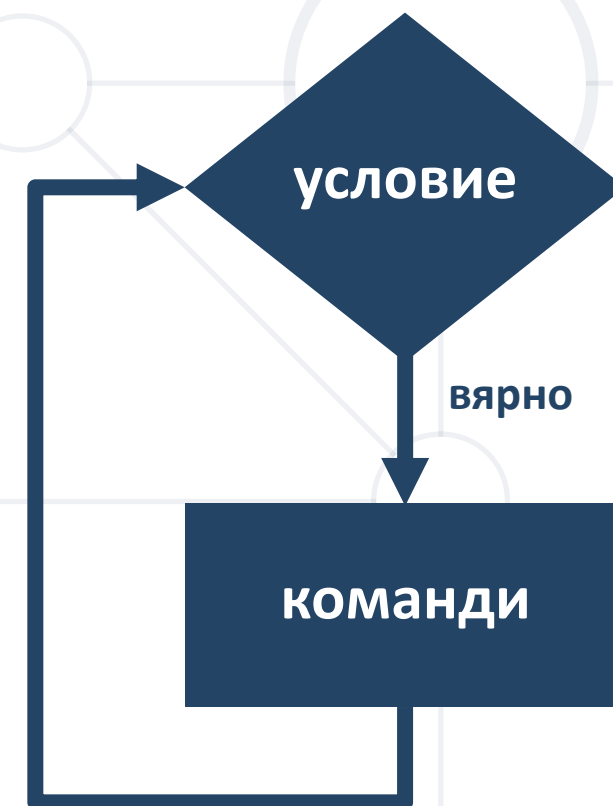
Условие за прекратяване  
на повторението

```
    print("Loop")  
    line = input()
```

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

Условието е винаги  
вярно

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





**Прекъсване чрез оператор `break`**

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

```
while True:  
    print("Loop")  
    if ...:  
        break
```

Условие за прекъсване на  
цикъла

# Прекратяване на цикъл – пример

Условие за прекратяване  
на повторението

```
a = 5
```

```
while True:  
    if a > 10:  
        break  
    print("a = " + str(a))  
    a += 1
```



```
a = 5  
a = 6  
a = 7  
a = 8  
a = 9  
a = 10  
Process finished with exit code 0
```

# Прекратяване на цикъл – пример



```
while True:  
    line = input()  
    if line == "Stop":  
        break  
  
print("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
```

```
while True:  
    line = input()  
  
    if line == "Stop":  
        break  
  
    print(line)
```

- 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}!")
```

- 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

```
n = int(input())  
sum = 0  
  
while sum < n:  
    current_num = int(input())  
    sum += current_num  
  
print(sum)
```

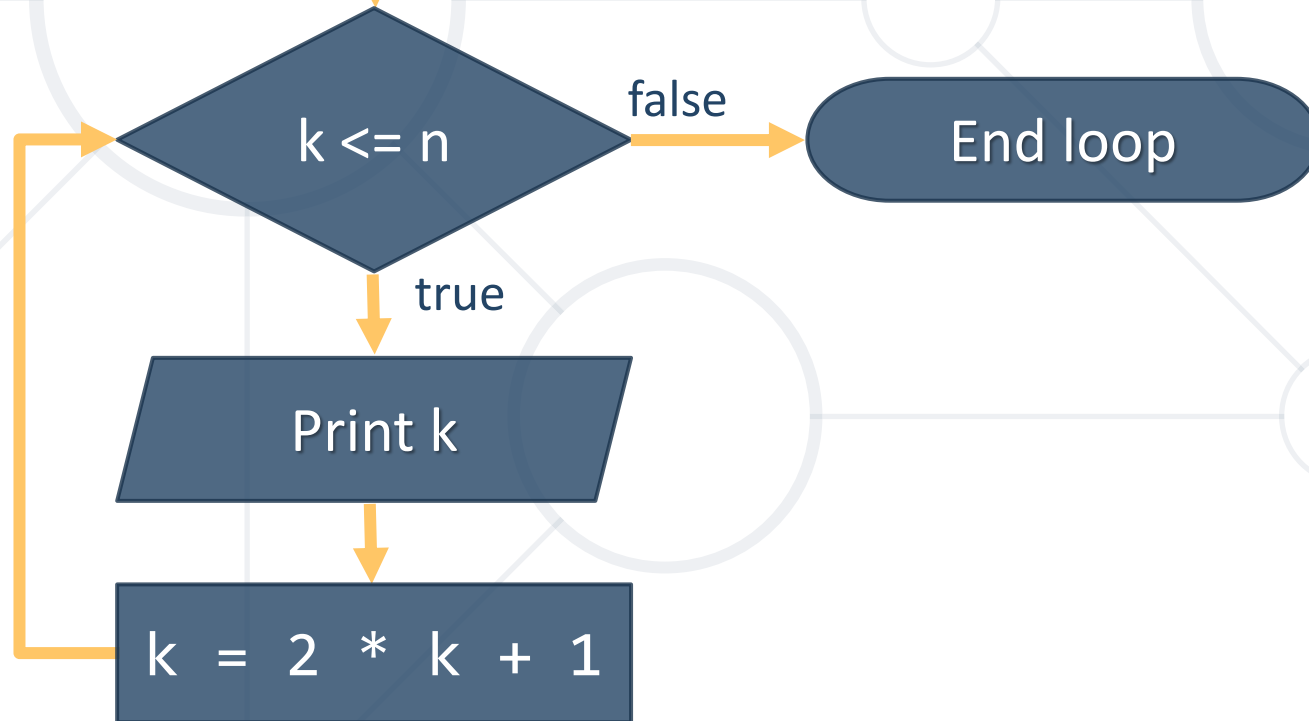
# Редица числа $2k + 1$ – условие

- 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** ...

Read input

$k = 1$



End loop

Read input

$k = 1$

$k \leq n$

false

true

Print  $k$

$k = 2 * k + 1$



# Редица числа $2k + 1$ – решение

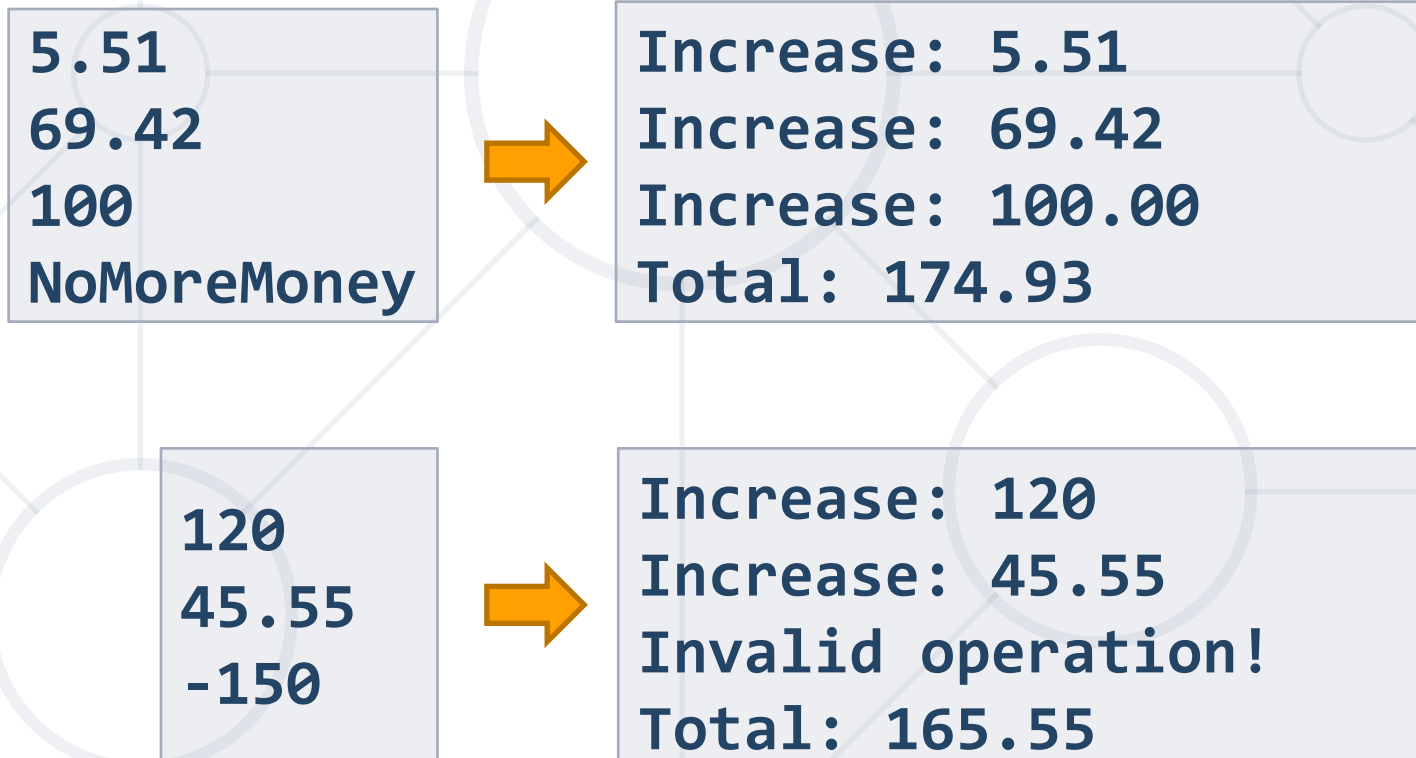
```
number = int(input())  
k = 1  
while k <= number:  
    print(k)  
    k = k * 2 + 1
```

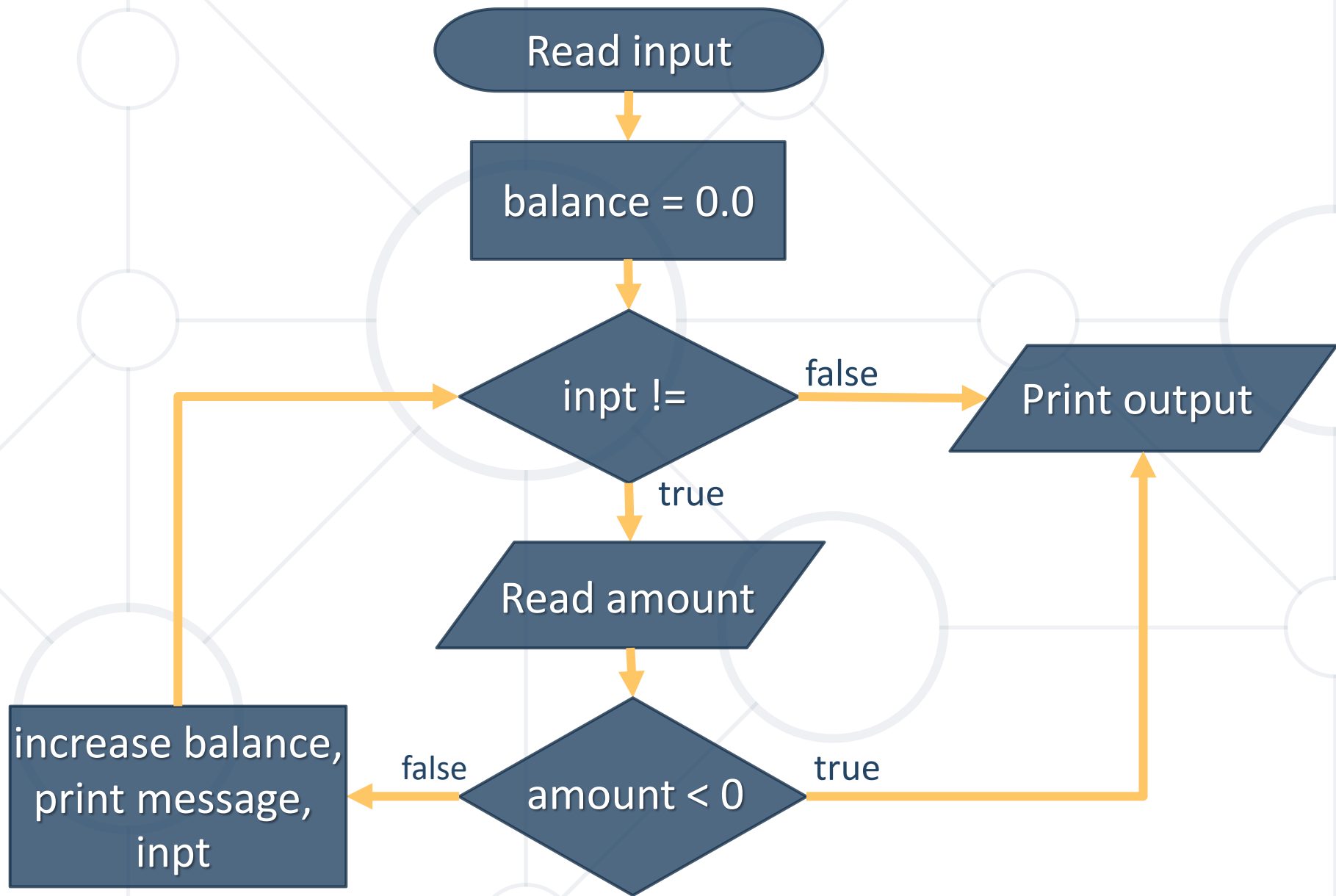
Повторение докато е в  
сила условието  $k \leq n$

- 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}"**

# Баланс на сметка – условие(3)

- Примерен вход и изход:





```
inpt = input()
balance = 0.0
while inpt != "NoMoreMoney":
    amount = float(inpt)
    if amount < 0:
        # TODO: Print output and exit the loop

    balance += amount
    print(f"Increase: {amount:.2f}")
    inpt = input()

print(f"Total: {balance:.2f}")
```

# Най-голямо число – пример

- 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-голямо число – решение

```
input = input()
max = -1000000000000000

while input != "Stop":
    num = int(input)

    if num > max:
        max = num
        input = input()

print(max)
```

# 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





# Най-малко число – решение

```
input = input()
min = 10000000000000000

while input != "Stop":
    # TODO: Use logic similar to the previous problem

print(min)
```

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

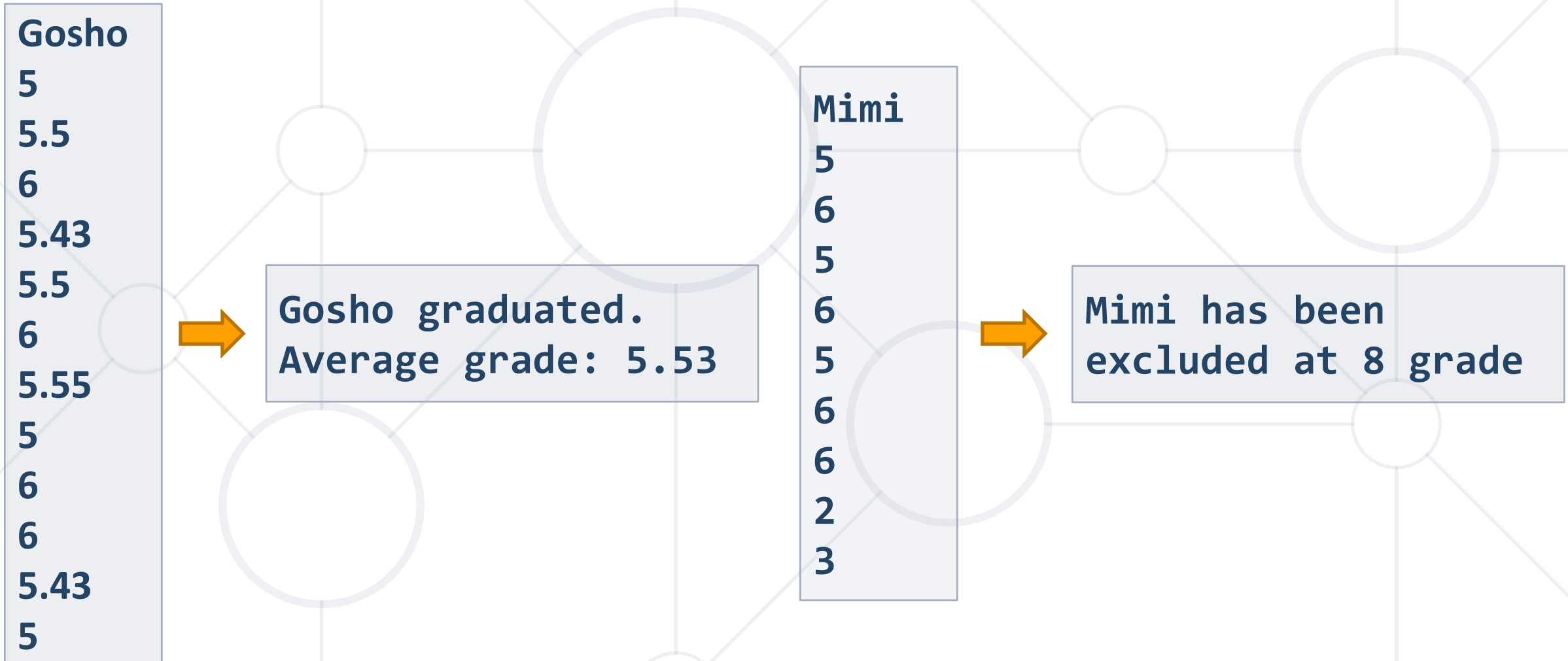
```
i = 0
while i < 10:
    if i % 2 == 0:
        i += 1
        continue
    print(i)
    i += 1
```




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

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

# Завършване – условие (2)



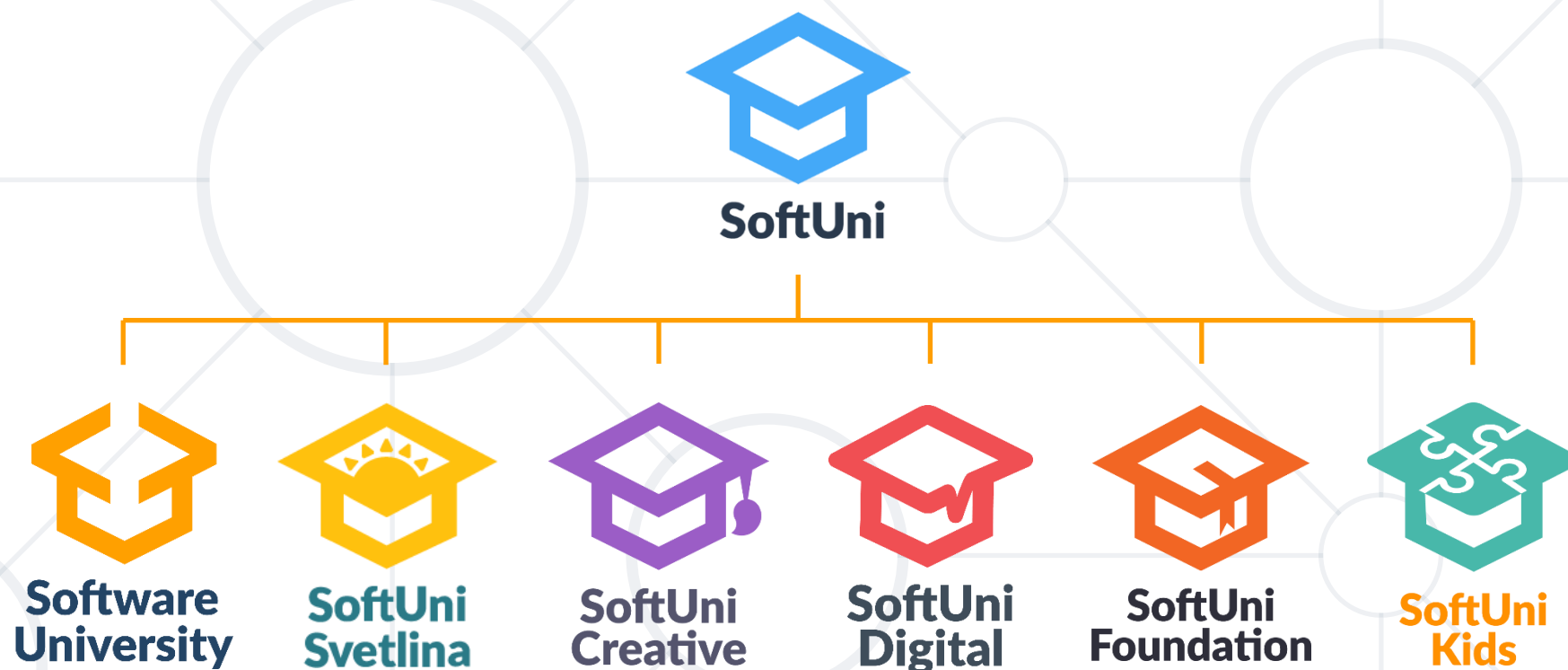
```
name = input()
grades = 1.0
sum_grades = 0.0
excluded = 0
while grades <= 12:
    grade = float(input())
    if grade < 4.00:
        #TODO: increase excluded count
        and break if is more than 1
        continue
    #TODO: add grade to sum and increase grades count
average = sum_grades / 12
#TODO: print the output
```



- 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](http://softuni.bg)
- Фондация "Софтуерен университет"
  - [softuni.foundation](http://softuni.foundation)
- Софтуерен университет @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)
- Дискуссионни форуми на СофтУни
  - [forum.softuni.bg](http://forum.softuni.bg)



Software University

