

# Повторения (цикли)

Прости повторения с While-цикъл



СофтУни

Преподавателски екип



SoftUni

Софтуерен университет

<https://softuni.bg>

# Имате въпроси?

[sli.do](https://sli.do)

**#pb-jan**

1. Преговор
2. **While**-цикъл – конструкция
3. Безкраен **while**-цикъл
4. Прекъсване на цикъл
5. Продължаване на цикъл





**Преговор**

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

```
for (int i = 1; i <= 3; ) {  
    cout << i;  
}
```

123

111

Infinite loop

Compile time  
error

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

```
for(; ;) {  
    cout << "SoftUni";  
}
```

Compile  
time error

Runtime error

Infinite loop

SoftUni

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

```
for (int i = 0; i < 2; i += 0.5) {  
    cout << i << ", ";  
}
```

0, 0.5, 1, 1.5

0, 1

Infinite loop

Compile time  
error



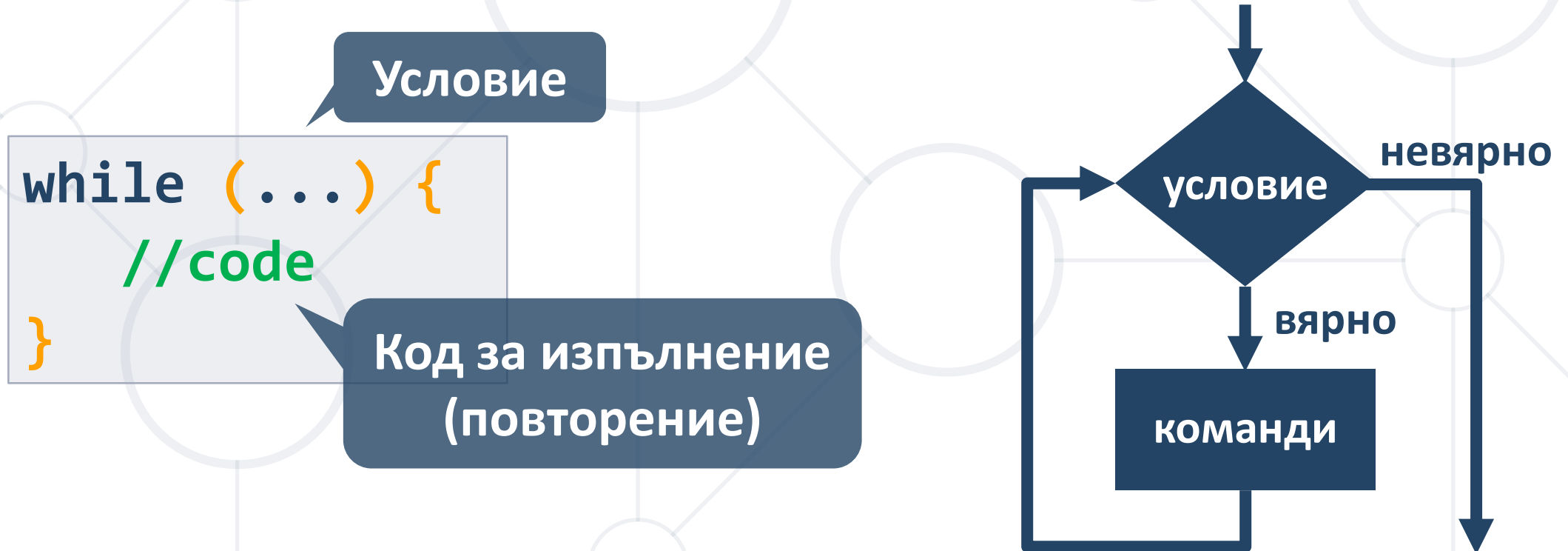
**while**

**while-цикъл**



# Повторения (цикли) – While-цикъл

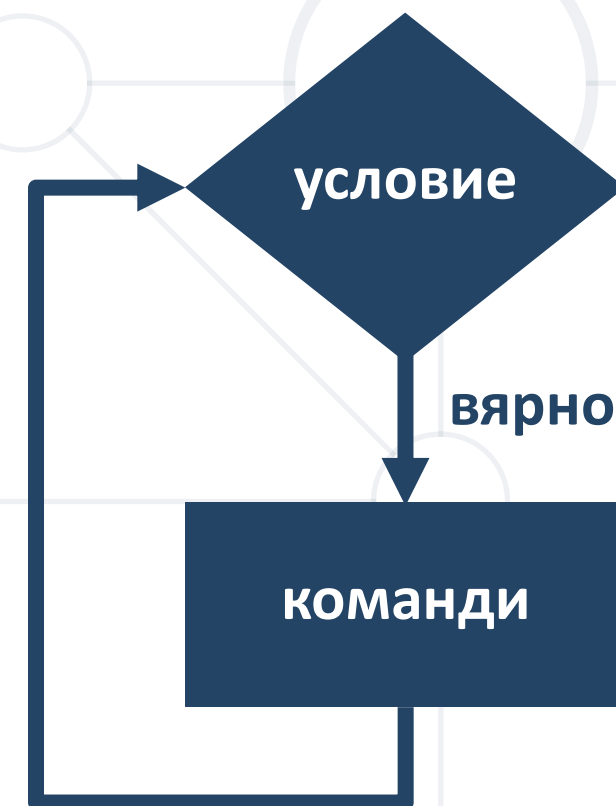
- В програмирането често се налага да изпълним блок с команди няколко пъти
  - За целта използваме цикли – **while**, **for** и други



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

Условието е винаги  
вярно

```
while (true) {  
    cout << "Infinite loop" << endl;  
}
```





# Прекъсване чрез оператор `break`

Безкрайни цикли

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


```
while (true) {  
    cout << "Infinite loop" << endl;  
    if (...) {  
        break;  
    }  
}
```

Условие за прекъсване на  
цикъла

# While-цикъл – пример

Условие за прекратяване  
на повторението

```
int a = 5;  
while (true) {  
    if (a > 10) {  
        break;  
    }  
    cout << "a = " << a << endl;  
    a++;  
}
```



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

# while-цикъл – пример



```
while (True) {  
    string input;  
    cin >> input;  
    if (input != "Stop") {  
        break;  
    }  
  
    cout << "Infinite loop" << endl;  
}
```

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
```

*Няма изход*

```
string input;  
cin >> input;
```

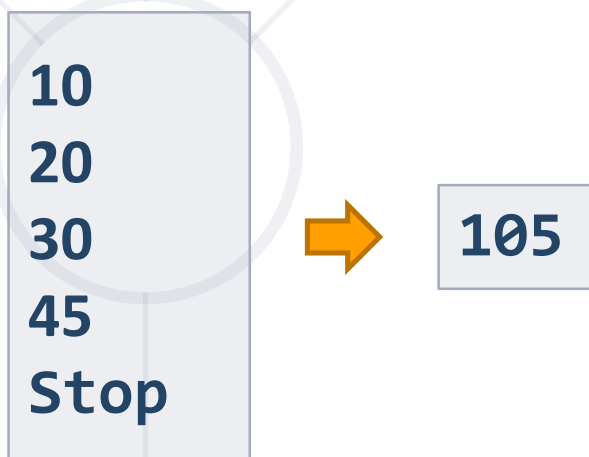
Когато командата стане  
"Stop", цикълът ще приключи

```
while (input != "Stop") {  
    cout << input << endl;  
    cin >> 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

- 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



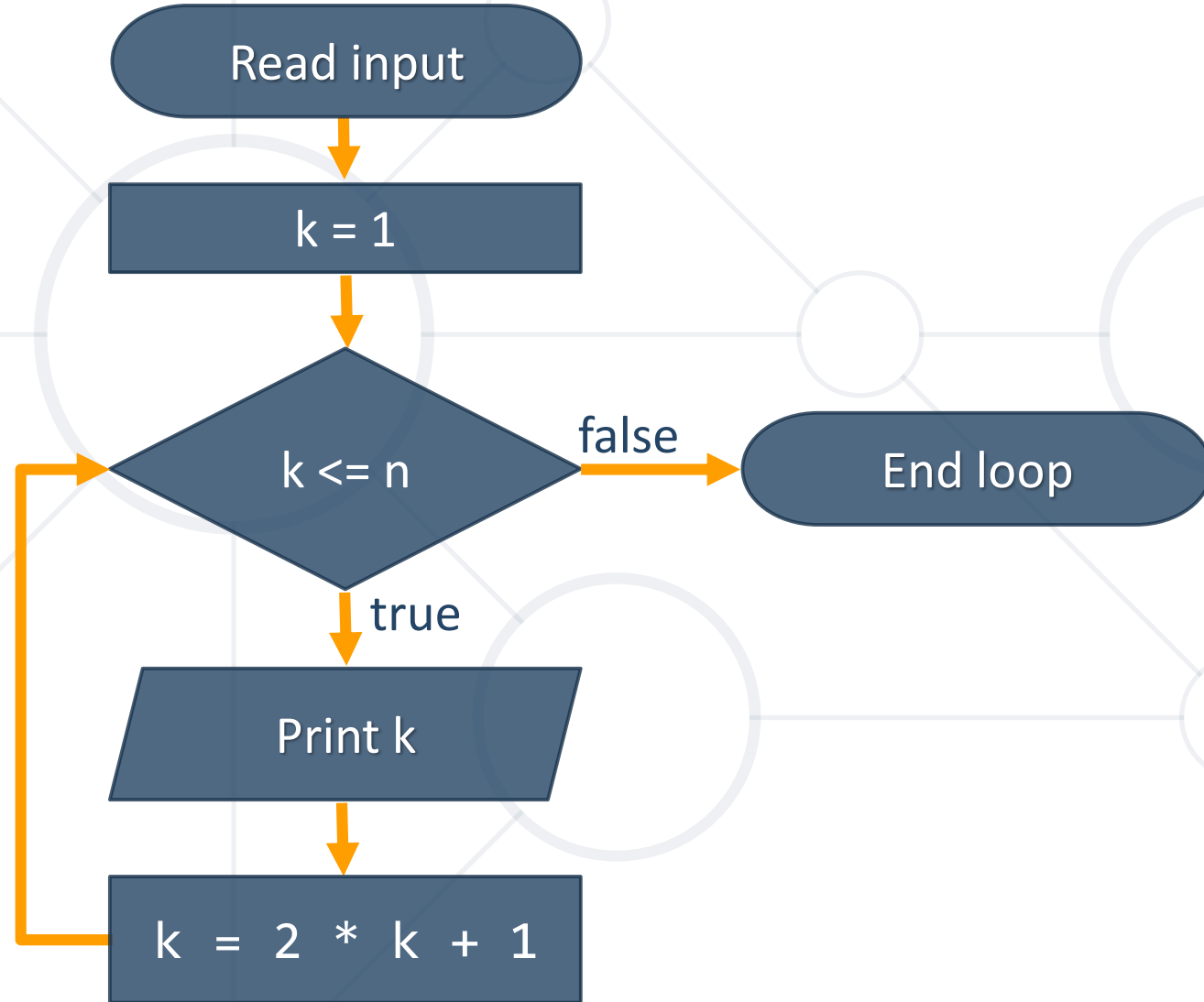
```
1
2
3
4
5
6
Stop
```

11

# 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**,  $(1 * 2) + 1 = \mathbf{3}$ ,  $(3 * 2) + 1 = \mathbf{7}$ ,  $(7 * 2) + 1 = \mathbf{15}$  ...



# Редица числа $2k+1$ - решение

```
int n; cin >> n;  
int k = 1;  
while (k <= n) {  
    cout << k << endl;  
    k = 2 * k + 1;  
}
```


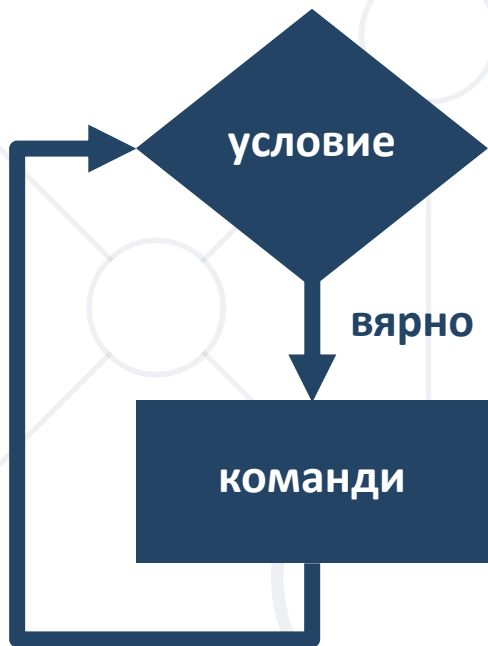
Повторение докато е в  
сила условието  $k \leq n$



# Прекъсване чрез оператор `break`

## Безкрайни цикли

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



Условието е винаги  
вярно

```
while (true) {  
    cout << "Infinite loop" << endl;  
}
```

- Оператор **break** – прекъсва цикъла

```
while (true) {  
    cout << "Infinite loop" << endl;  
    if (...) {  
        break;  
    }  
}
```

Условие за прекъсване на  
цикъла



- 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: {общата сума в сметката}"**



# Баланс на сметка - условие(3)

- Примерен вход и изход:

3  
5.51  
69.42  
100



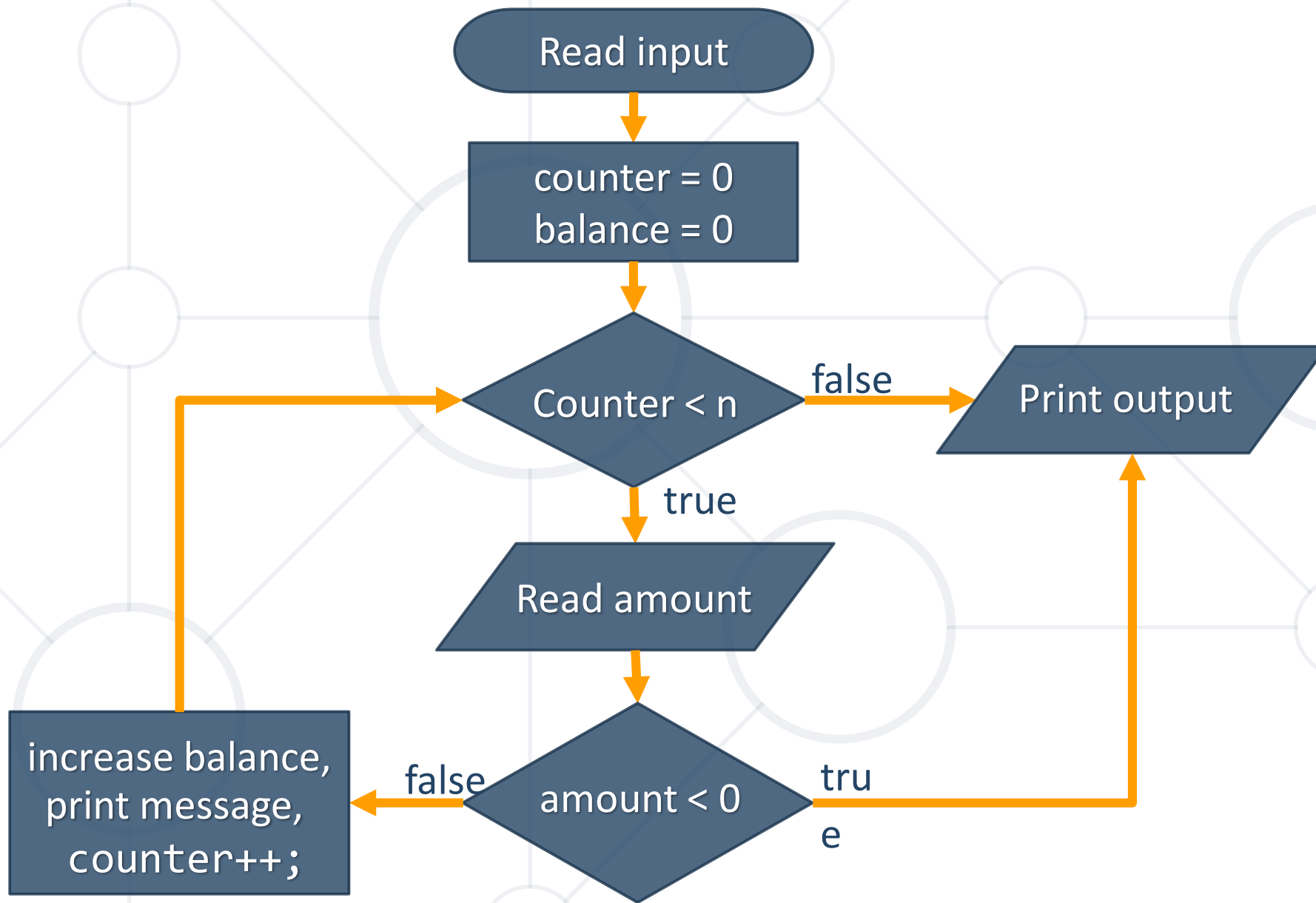
Increase: 5.51  
Increase: 69.42  
Increase: 100.00  
Total: 174.93

5  
120  
45.55  
-150



Increase: 120  
Increase: 45.55  
Invalid operation!  
Total: 165.55



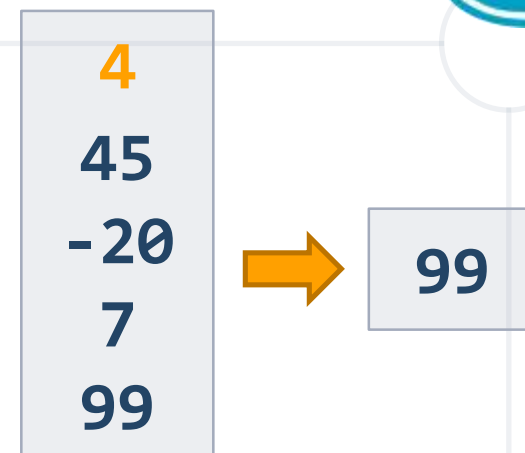
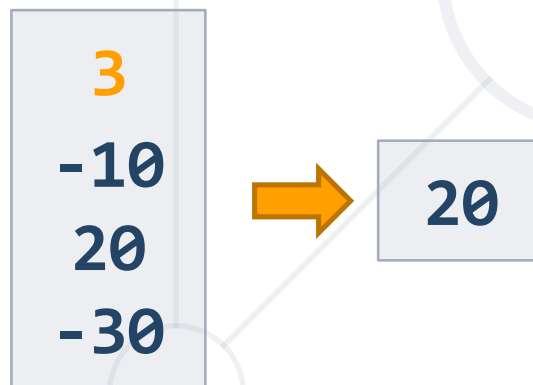
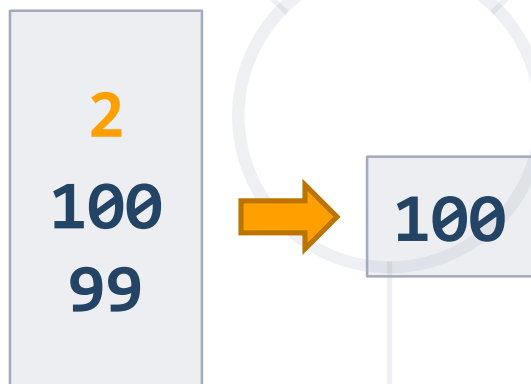


# Баланс на сметка - решение

```
int n; cin >> n;
int counter = 0;
double balance = 0.0;
while (counter < n) {
    double amount; cin >> amount;
    if (amount < 0) // TODO: Print output and exit the loop
        balance += amount;
    cout << "Increase: " << amount << endl;
    counter++;
}
cout << "Total: " << balance << endl;
```

# 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:



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

```
for (int i = 0; i < 10; i++) {  
    if (i % 2 == 0)  
    {  
        continue;  
    }  
    cout << i << endl;  
}
```



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

# Най-малко число - условие

- Напишете програма, която:
  - Прочита цяло число(*n*) от потребителя
  - Прочита *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



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



# Завършване - условие (2)

- Примерен вход и изход:

Peter  
4  
5.5  
6  
5.43  
4.5  
6  
5.55  
5  
6  
6  
5.43  
5



Peter 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

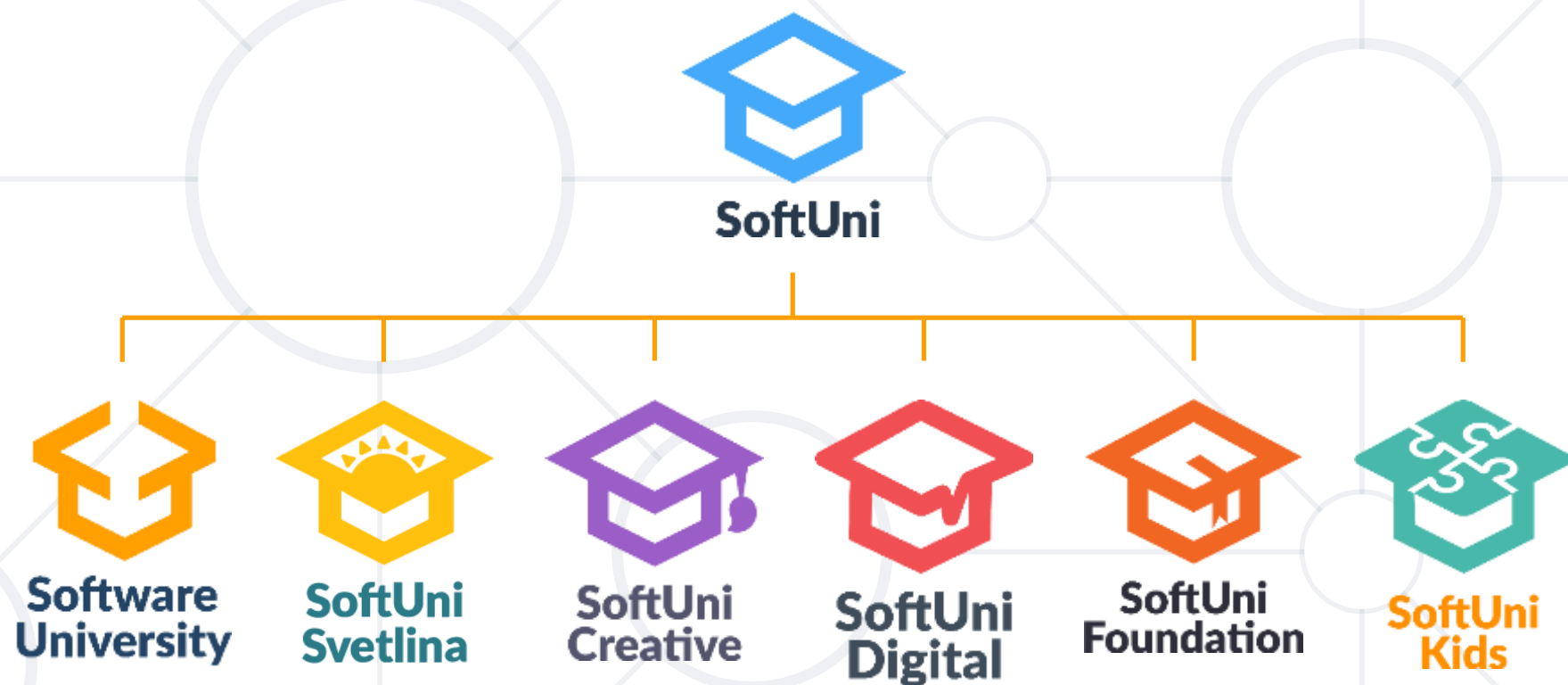
```
string name; cin >> name;
double grades = 1;
double sum = 0;
while (grades <= 12) {
    double grade; cin >> grade;
    if (grade < 4.00) {
        continue;
    }
    // TODO: add grade to sum and increase grades count
}
double average = sum / 12;
cout.setf(ios::fixed);
cout.precision(2);
cout << name << " graduated. Average grade: " << average << endl;
```



- 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

