

Повторения (цикли)

Прости повторения с `while`-цикъл



СофтУни

Преподавателски екип



Software
University



SoftUni
Foundation



Софтуерен университет
<http://softuni.bg>

Имате въпроси?

sli.do

#pb-jan

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





Преговор

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

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

123

Infinite loop

111

Compile time
error

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

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

123

Infinite loop

111

Compile time
error

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

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

Infinite loop

Runtime error

Compile
time error

SoftUni

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

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

Infinite loop

Runtime error

Compile
time error

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

Compile time
error

Infinite loop

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

Compile time
error

Infinite loop



**Увеличаване и намаляване на стойността
на променливи**

- Инкрементиране - увеличаването на стойността на дадена променлива
 - Извършва се чрез оператори за инкрементиране - префиксни и постфиксни

Пример	Име	Резултат
++a	Пре-инкрементация	Увеличава стойността с единица и връща a
a++	Пост-инкрементация	Връща a и увеличава стойността с единица

- Извършва се само върху променливи, които имат числена стойност

- Пре-инкрементация

```
int a = 1;  
cout << ++a << endl;    // 2  
cout << a << endl;      // 2
```

Стойността на променливата а се увеличава с 1 и след това се принтира

- Пост-инкрементация

```
int a = 1;  
cout << a++ << endl;    // 1  
cout << a << endl;      // 2
```

Първо се принтира променливата а и след това се увеличава с 1

- Декрементиране – намаляването на стойността на дадена променлива
 - Извършва се чрез оператори за декрементиране – префиксни и постфиксни

Пример	Име	Резултат
--a	Пре-декрементация	Намалява стойността с единица и връща a
a--	Пост-декрементация	Връща a и намалява стойността с единица

- Извършва се само върху променливи, които имат числена стойност

- Пре-декрементация

```
int a = 1;  
cout << --a << endl;    // 0  
cout << a << endl;      // 0
```

Стойността на променливата a се намалява с 1 и след това се принтира

- Пост-декрементация

```
int a = 1;  
cout << a-- << endl;    // 1  
cout << a << endl;      // 0
```

Първо се принтира променливата a и след това се намалява с 1

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.

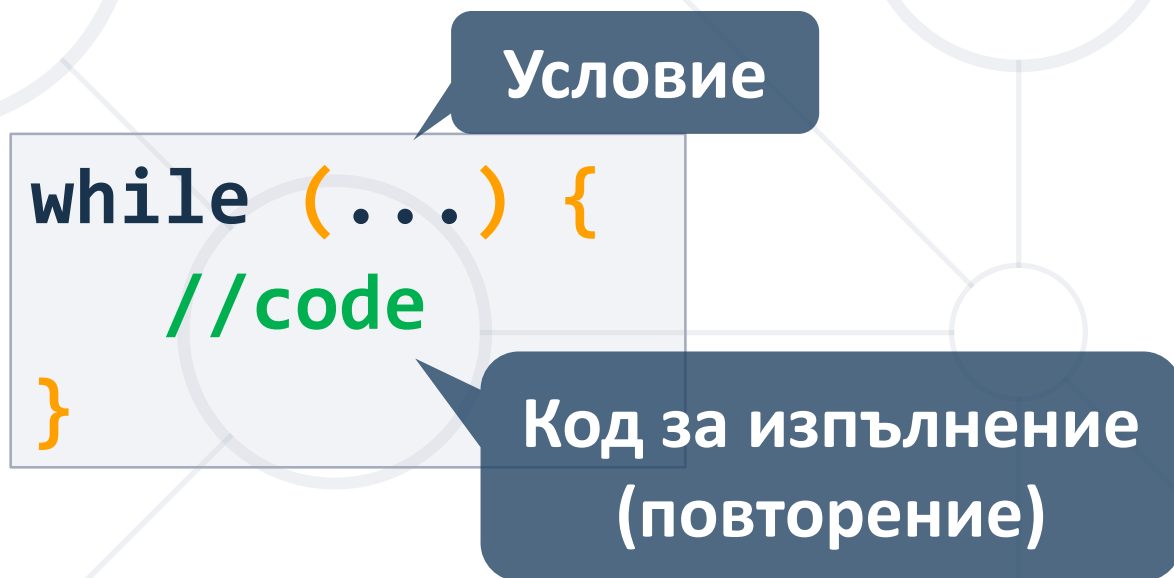
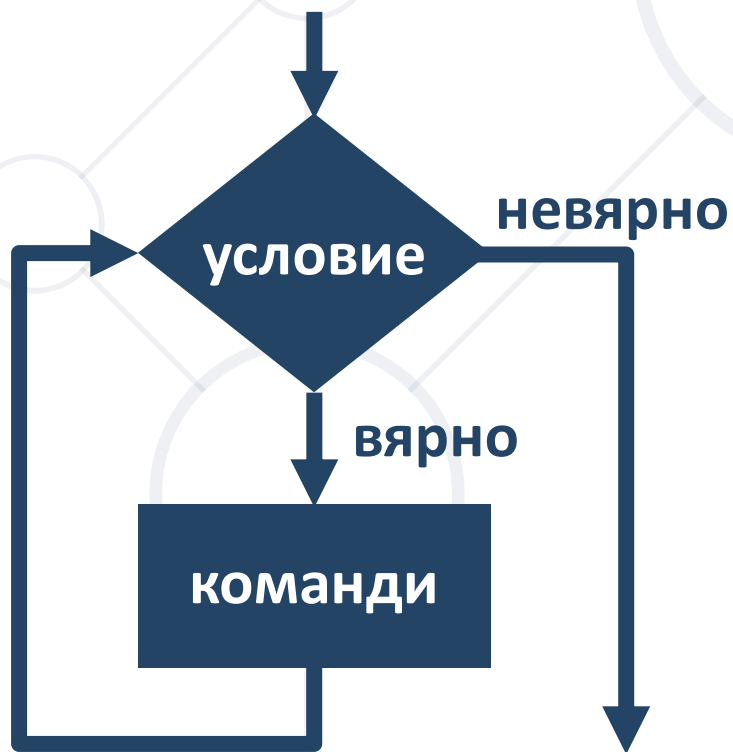
while

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

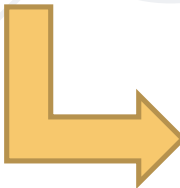


while-цикъл – пример

```
int a = 5;
```

Условие за прекратяване
на повторението

```
while (a <= 10) {  
    cout << "a = " << a << endl;  
    a++;  
}
```



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

- 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

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

```
while (input != "Stop") {  
    cin >> input;  
}
```

Когато командата стане
"Stop", цикълът ще приключи

- 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

```
string username, password;  
cin >> username >> password;  
  
string input;  
cin >> input;  
while (input != password) {  
    cin >> input;  
}  
  
cout << "Welcome " << username << "!" << endl;
```

Тестване на решението: <https://judge.softuni.bg/Contests/Compete/Index/1175#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



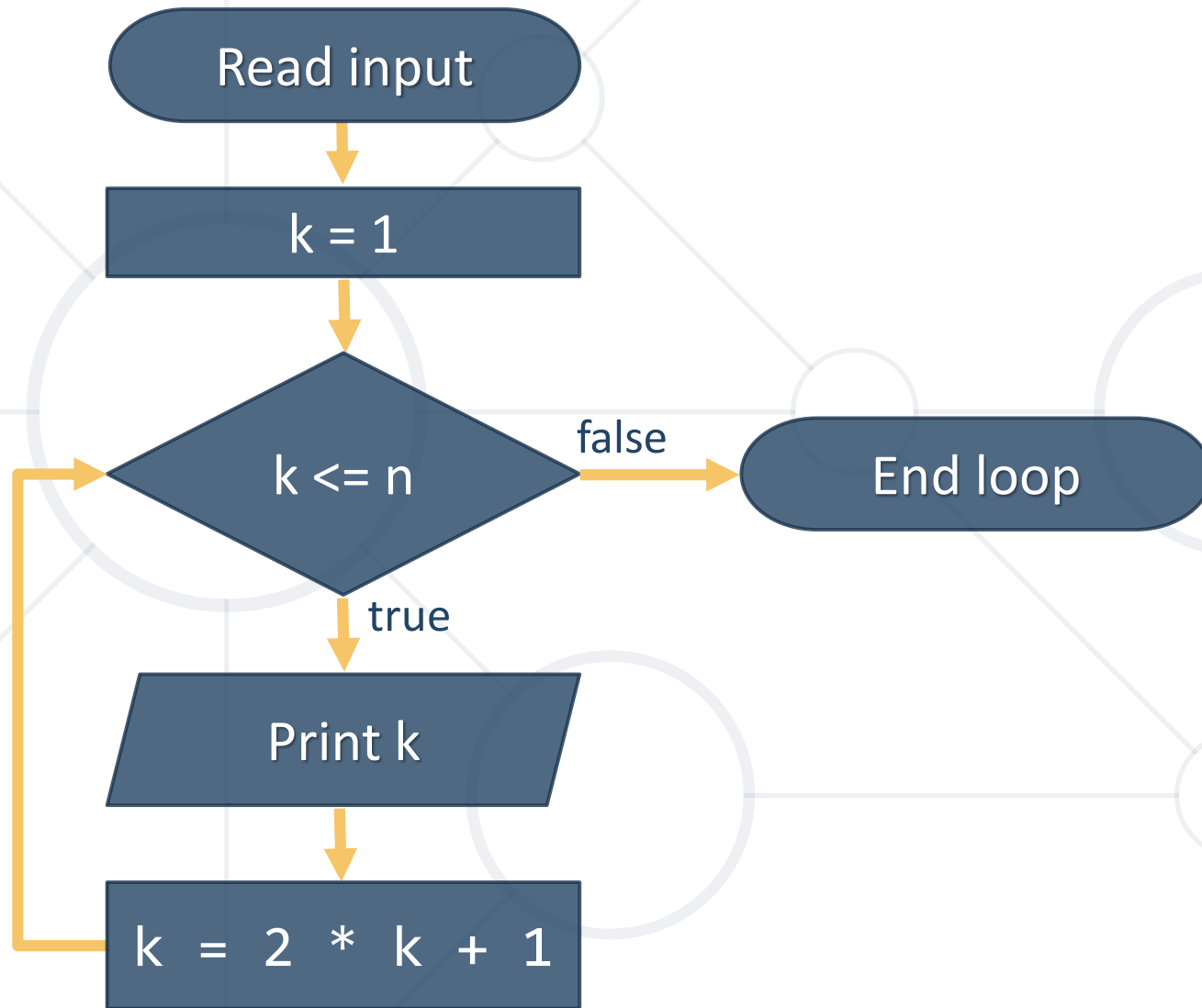
11

```
string input;  
cin >> input;  
int sum = 0;  
  
while (input != "Stop")  
{  
    int currentNum = stoi(input);  
    sum += currentNum;  
    cin >> input;  
}  
cout << sum << endl;
```


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

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


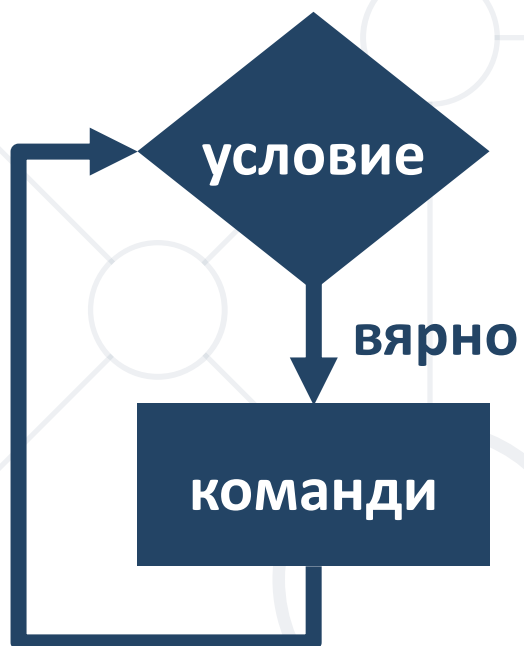
Повторение докато е в
сила условието $k \leq n$



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



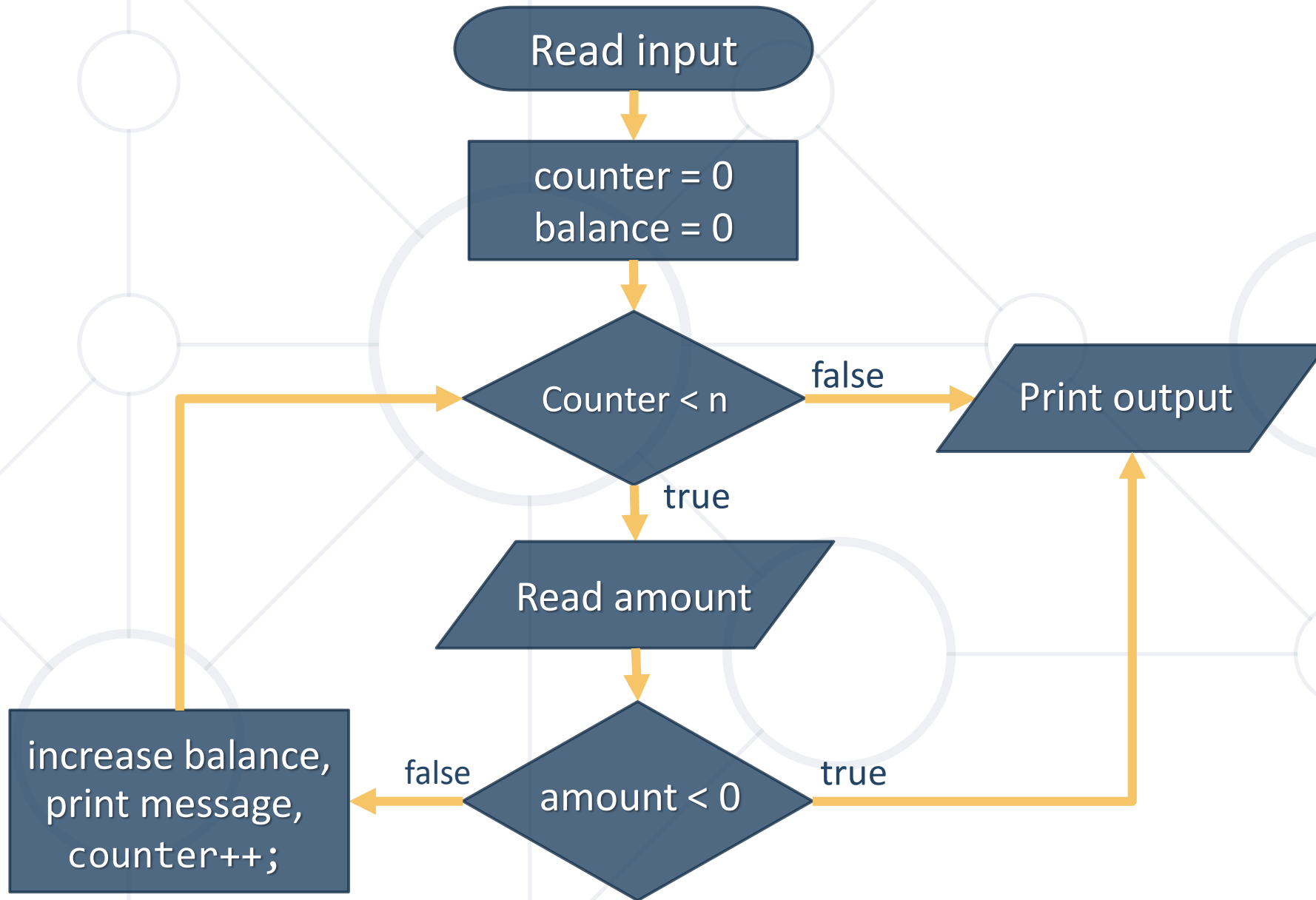
Increase: 5.51
Increase: 69.42
Increase: 100.00
Total: 174.93

5
120
45.55
-150



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

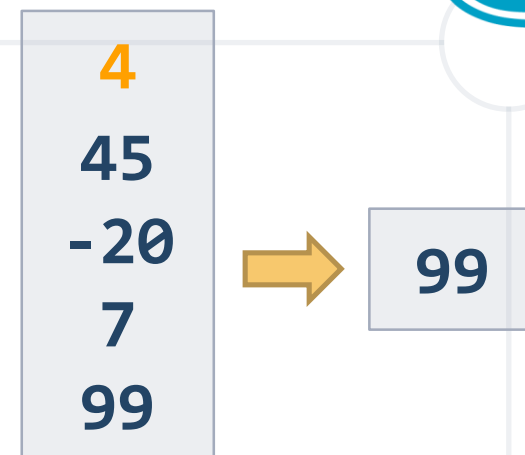
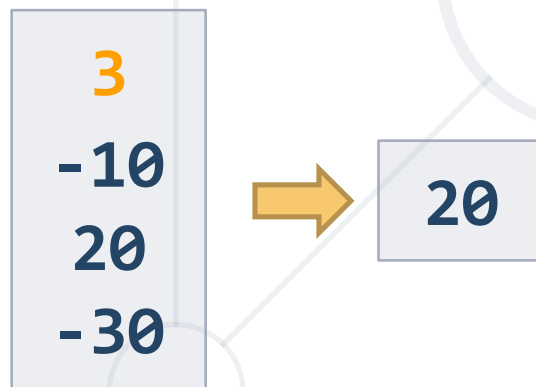
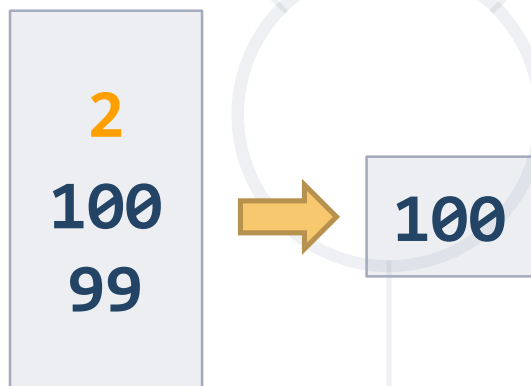




```
int n; cin >> n;
int counter = 0;
double balance = 0.0;
while (counter < n) {
    double amount; cin >> amount;
    if (amount < 0) // TODO: Print output and exit the loop
        balance += amount;
    cout << "Increase: " << amount << endl;
    counter++;
}
cout << "Total: " << balance << endl;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

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:

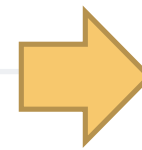


Най-голямо число - решение

```
int n;  
cin >> n;  
int count = 0;  
int max = INT_MIN;  
while (count < n) {  
    int num;  
    cin >> num;  
    count++;  
    if (num > max) {  
        max = num;  
    }  
}  
cout << "max = " << max << endl;
```

- 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



```
int n;  
cin >> n;  
int count = 0;  
int min = INT_MAX;  
while (count < n) {  
    //TODO: Use logic similar to the previous problem  
}
```


- Напишете програма, 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
 - 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

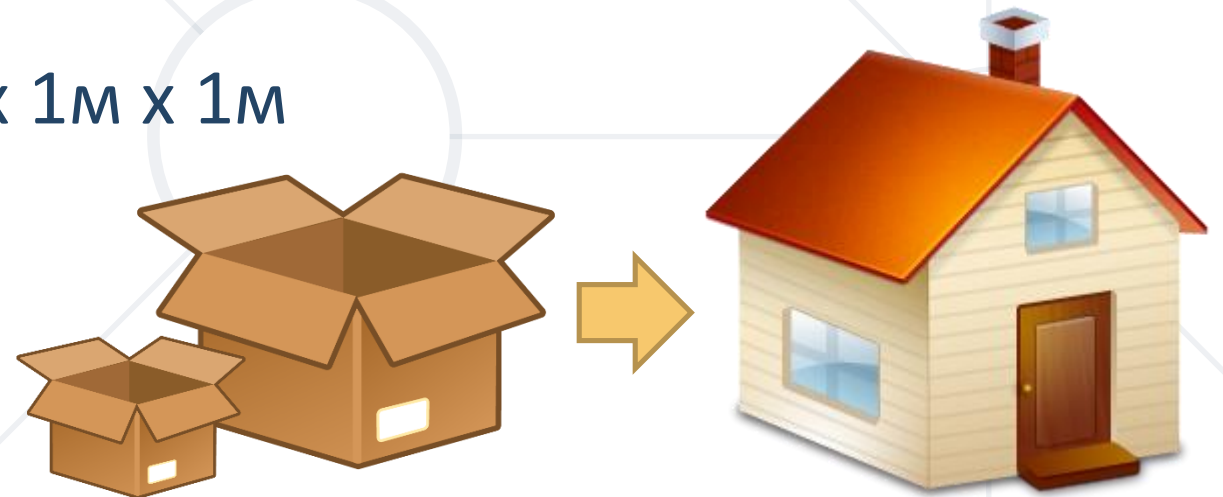


Ani graduated.
Average grade: 5.45

```
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;
```



- Напишете програма, която:
 - Прочита 3 цели числа – ширина, дължина, височина
 - Прочита брой кашони до получаване на команда "Done"
 - Изчислява дали кашоните могат да се преместят в помещение с прочетените размери
 - 1 кашон е с размери 1м x 1м x 1м



- Ако помещението не може да събере кашоните, трябва да се принтира:
 - "No more free space! You need {брой недостигащи куб. метри} Cubic meters more."
- При получаване на команда "Done" и налично свободно място:
 - "{брой свободни куб. метри} Cubic meters left."

Преместване - условие (3)

- Примерен вход и изход:

10
1
2
4
6
Done



10 Cubic meters left.

2
2
2
9



No more free space! You need 1 Cubic meters more.



```
int width; cin >> width;
// TODO: Read the length and height

int volume = width * length * height;
bool hasVolume = true;

string command; cin >> command;
while (!(command == "Done"))
{
    int box = std::stoi(command);
    volume -= box;
    ...
}
```

Преместване - решение (2)

```
...  
    if (volume < 0) {  
        hasVolume = false;  
        break;  
    }  
    cin >> command;  
}  
  
if (hasVolume)  
    cout << volume << " Cubic meters left." << endl;  
else  
    cout << "No more free space! You need " <<  
    std::abs(volume) << " Cubic meters more." << endl;
```

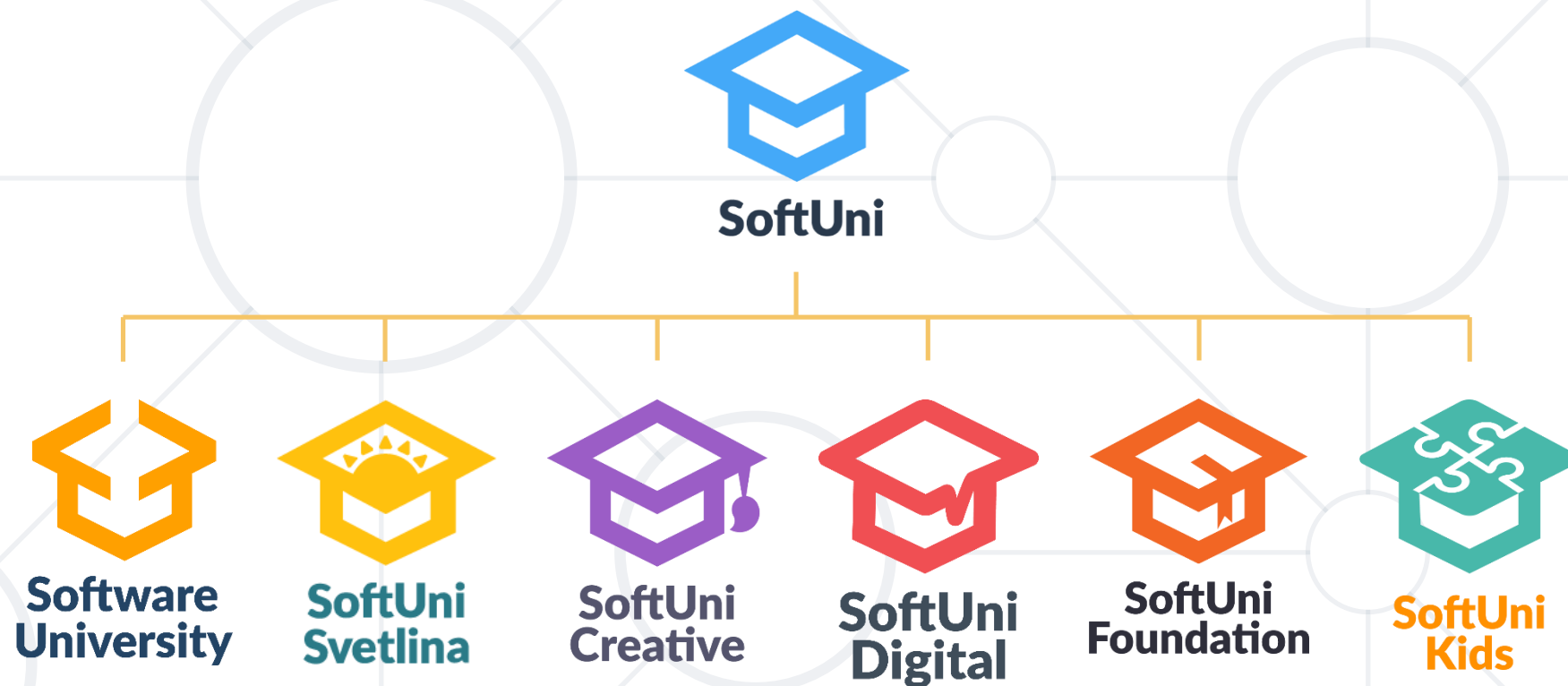
Цикълът прекъсва

Прочитае отново
командата

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



Въпроси?



SoftUni Diamond Partners



XSsoftware



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



- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"



Обучения в СофтУни

- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - [facebook.com/SoftwareUniversity](https://www.facebook.com/SoftwareUniversity)
- Software University Forums
 - forum.softuni.bg

