

Loops

For Loop, While Loop



СофтУни

Преподавателски екип



SoftUni

Софтуерен университет
<https://softuni.bg>

Table of Contents

1. Incrementing a Variable
2. What is a Loop?
3. For Loop
4. Working with Text
5. While Loop
6. Interrupting a Loop



Have a Question?

sli.do

#prgm-for-qa



**Incrementing and Decrementing the Value of
Variables**

- **Incrementing** – increasing the value of a given variable
променлива
 - It is performed by using incrementing operators: **prefix** and **postfix**
 - Applied only to variables that have a numerical value

Example	Name	Result
++a	Pre -incrementation	Increases the value by 1 and returns a
a++	Post -incrementation	Returns a and increases the value by 1

Incrementation (2)

- **Pre**-incrementation

```
int a = 1;  
Console.WriteLine(++a); // 2  
Console.WriteLine(a);   // 2
```

The value of the variable is increased by 1 and is printed after

- **Post**-incrementation

```
int a = 1;  
Console.WriteLine(a++); // 1  
Console.WriteLine(a);   // 2
```

The variable a is first printed and is increased by 1 after

- **Decrementing** – decreasing the value of a given variable
 - It is performed by using decrementing operators: **prefix** and **postfix**
 - Applied only to variables that have a **numerical** value

Example	Name	Result
--a	Pre -decrementation	Decreases the value by 1 and returns a
a--	Post -decrementation	Returns a and decreases the value by 1

Decrementation (2)

- **Pre**-decrementation

```
int a = 1;  
Console.WriteLine(--a); // 0  
Console.WriteLine(a);  // 0
```

The value of the variable is decreased by 1 and is printed thereafter

- **Post**-decrementation

```
int a = 1;  
Console.WriteLine(a--); // 1  
Console.WriteLine(a);  // 0
```

The value of the variable is printed and is decreased by 1 thereafter



Repetition of Blocks of Code

For-loop construction

What is a Loop? (1)

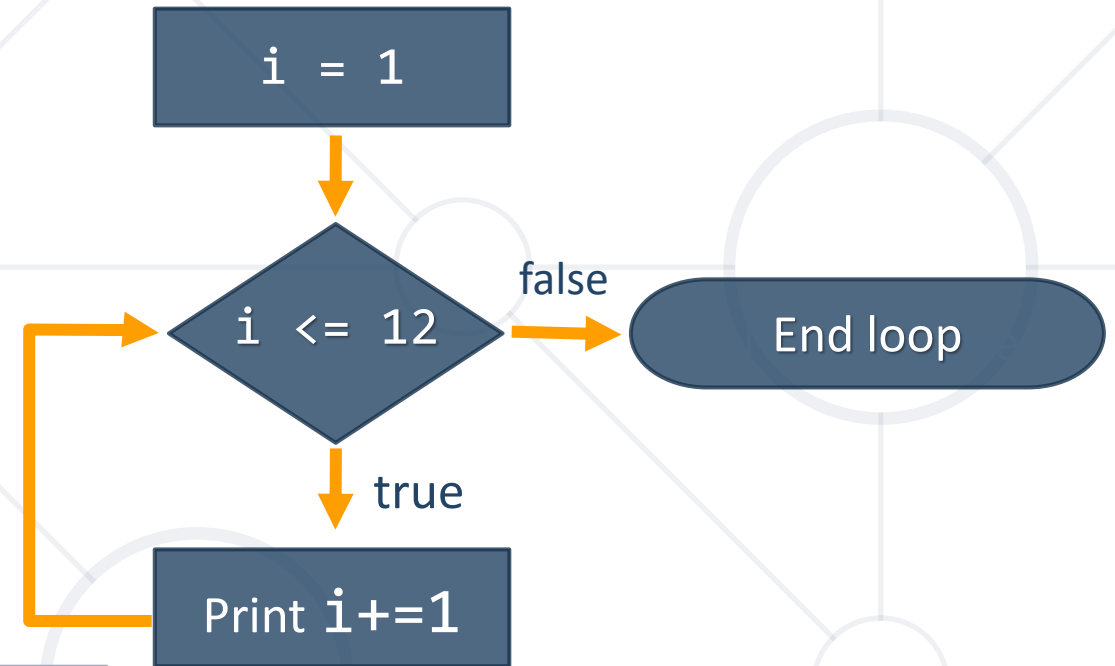
- We often need to **repeat** the same action **multiple times**
- When we graduate we count to 12



What is a Loop? (2)

- Loops in programming allow us to repeat **the same actions** a specific number of times:

```
for (int i = 1; i <= 12; i += 1)
{
    Console.WriteLine(i);
}
```



For-Loop – Construction

- We can repeat actions until a certain point using **for**-loops

Keyword for the construction

Initial value

Final value

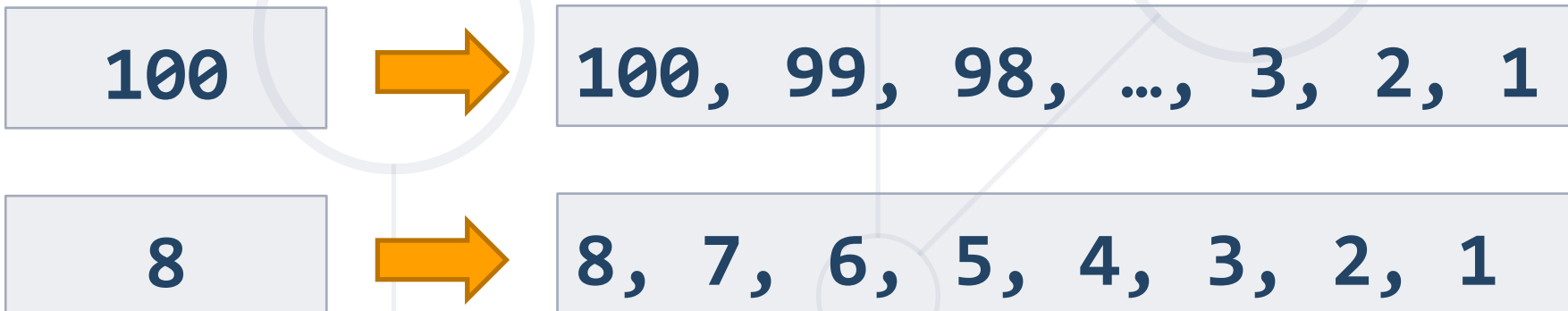
```
for (int i = 1; i <= 12; i += 1)
{
    Console.WriteLine(i);
}
```

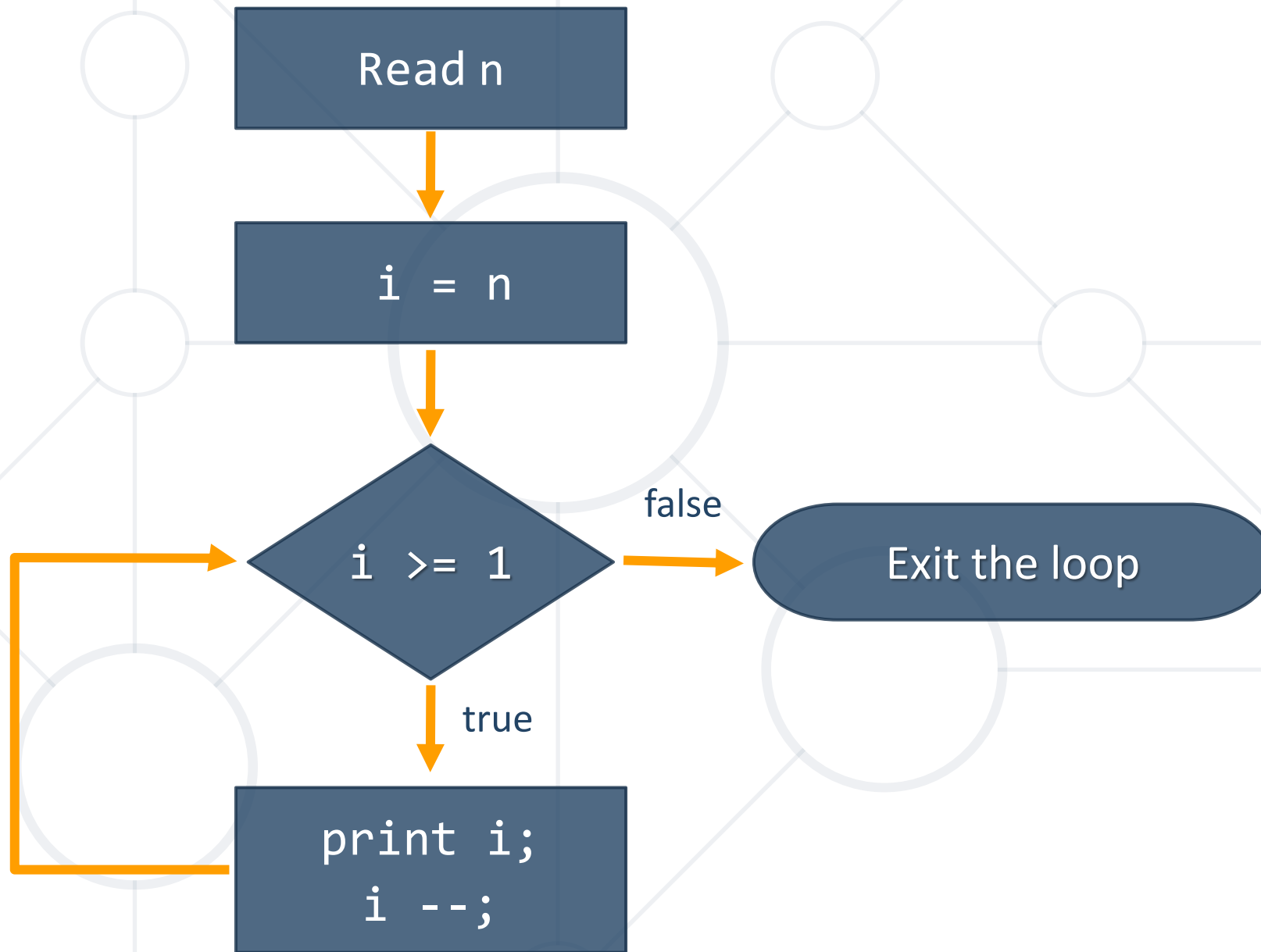
Step

Body of the loop: Block of code for repetition

Numbers from 1 to N Reversed - Problem

- Write a Console program, that:
 - Reads from the Console an integer value - **n**
 - Prints on the Console the numbers from **n** to **1** in reversed order (step -1)
- Example input and output:





Numbers from 1 to N Reversed – Solution

```
int n = int.Parse(Console.ReadLine());  
for (int i = n; i >= 1; i--)  
{  
    Console.WriteLine(i);  
}
```

Reversed condition: `i >= 1`

Decreasing step: `-1`

Numbers from 1 to N Step 3 - Problem

- Write a console program, that:
 - Reads an integer number - **n**
 - Prints on the console the numbers from **1** to **n** with step **3**
- Example input and output:



Read n

i = 1

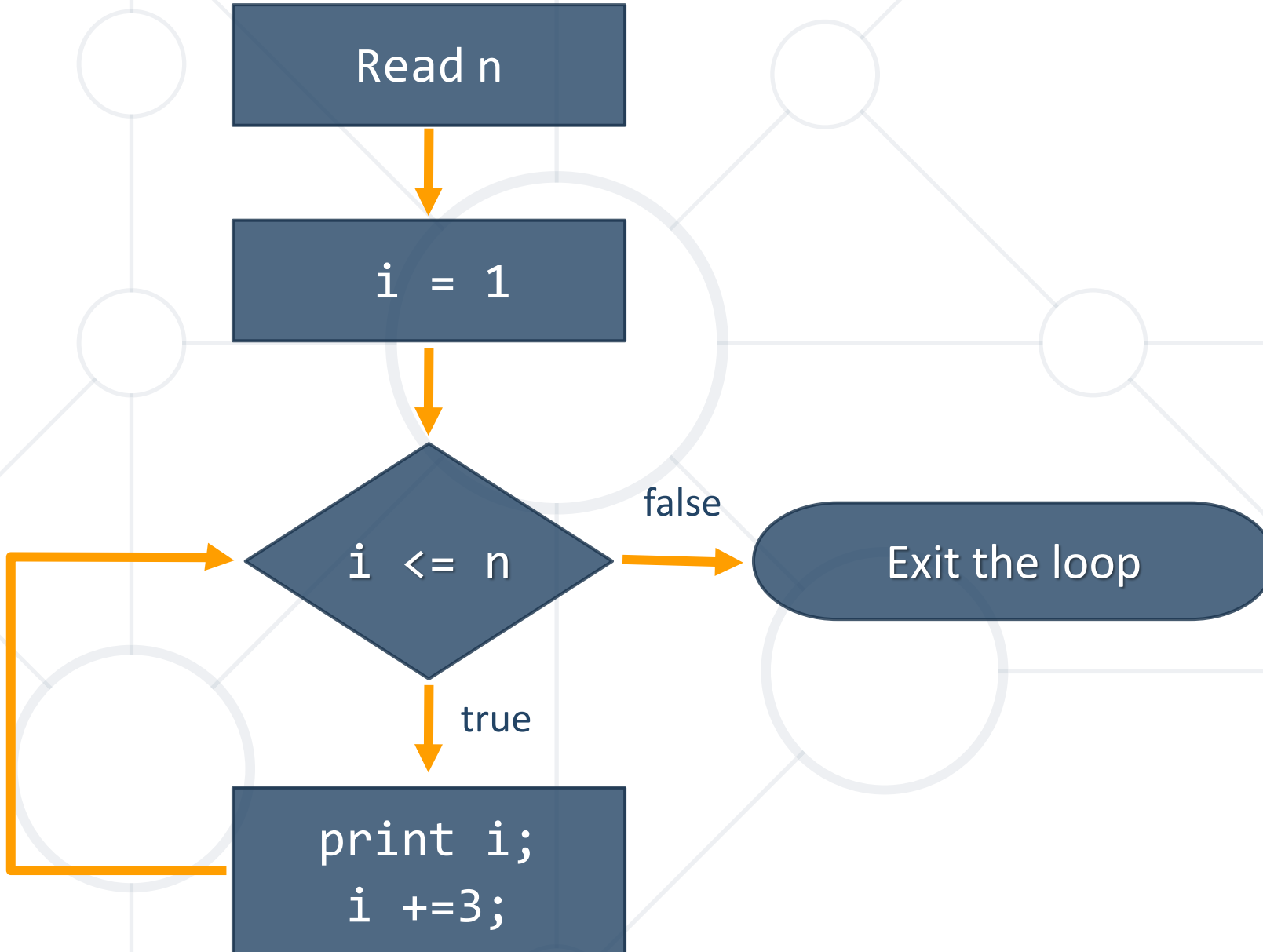
i <= n

false

Exit the loop

true

print i;
i += 3;



Numbers from 1 to N Step 3 – Solution

```
int n = int.Parse(Console.ReadLine());  
for (int i = 1; i <= n; i += 3)  
{  
    Console.WriteLine(i);  
}
```

Assign step to 3

Even Powers of 2 – Problem

- Write a console program, that:
 - Reads an integer value **n** from the Console
 - Prints the even powers of **2** to **2ⁿ**: $2^0, 2^2, 2^4, 2^8, \dots, 2^n$
- Example input and output:

10 → 1, 4, 16 , ..., 1024

7 → 1, 4, 16 , ..., 64

Even Powers of 2 – Solution

```
int n = int.Parse(Console.ReadLine());  
int num = 1;  
for (int i = 0; i <= n; i += 2)  
{  
    Console.WriteLine(num);  
    num = num * 2 * 2;  
}
```

Assign to
step 2



Working with text

Iterating over String

- We can take the length of a text

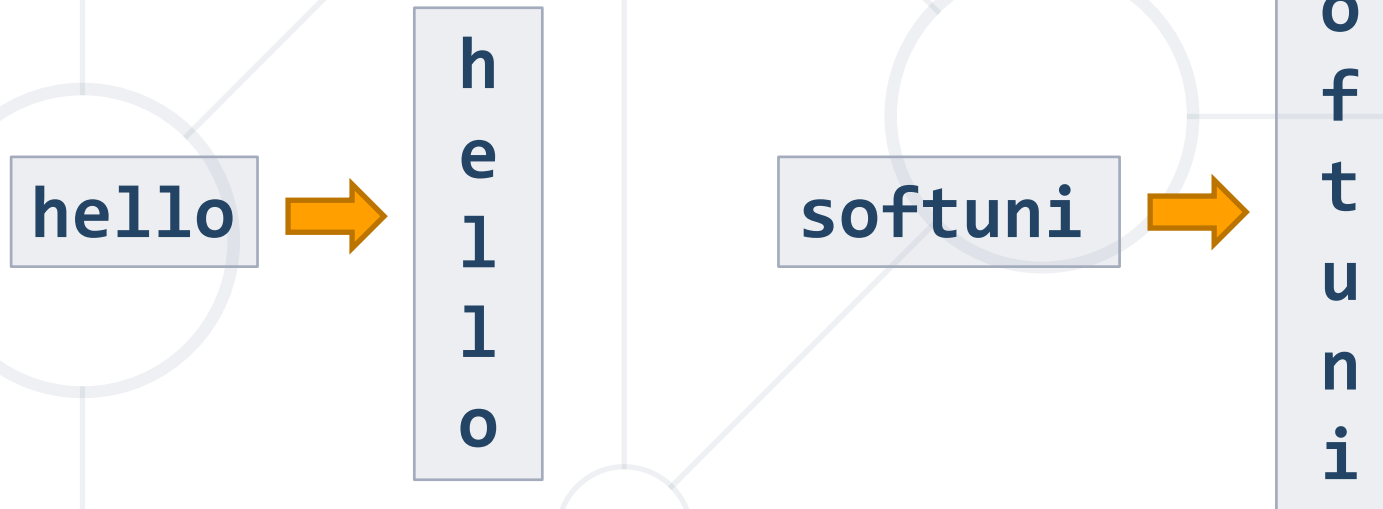
```
string text = "SoftUni"  
int length = text.Length; // 7
```

- We can take a single symbol (char) from a text by a given index

```
string text = "SoftUni" // U  
char letter = text[4];
```

String of Symbols – Problem

- Write a console program, that:
 - Reads text from the console (**string**)
 - Prints every symbol (**char**) from the text, each on a new line
- Example input and output:



String of Symbols – Solution

```
string input = Console.ReadLine();
```

We take the length of the text

```
for (int i = 0; i < input.Length; i++)
```

```
{
```

```
    Console.WriteLine(input[i]);
```

We take every single symbol by index i

```
}
```


Sum of Vowels – Problem

- Write a console program, that:
 - Reads string input from the console
 - Outputs the sum of the vowels as shown in the table below:

a	e	i	o	u
1	2	3	4	5

- Example input and output:

hello



6

(e+o = 2+4 = 6)

hi



3

(i = 3)

bamboo



9

(a+o+o = 1+4+4 = 9)

beer



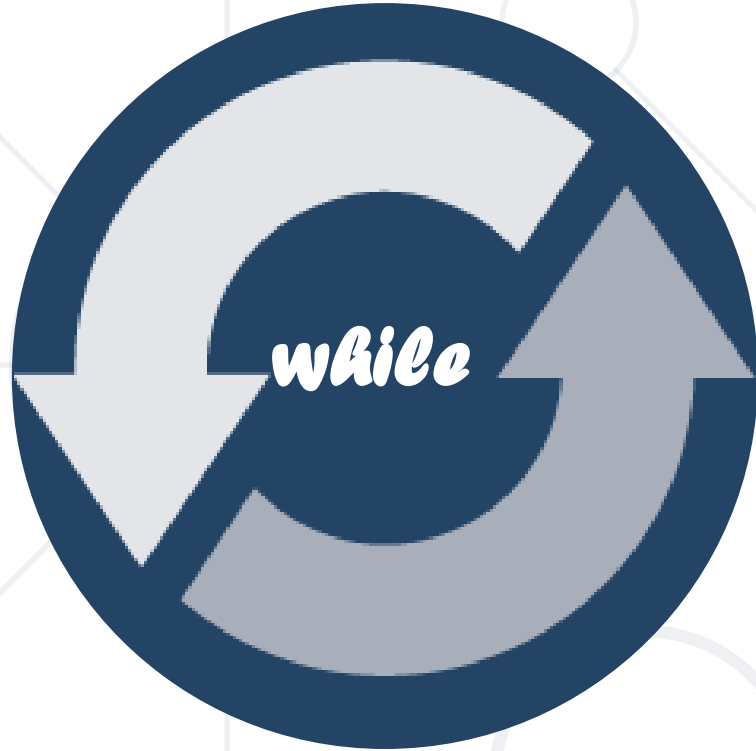
4

(e+e = 2+2 = 4)

Sum of Vowels – Solution

```
string input = Console.ReadLine();
int sum = 0;

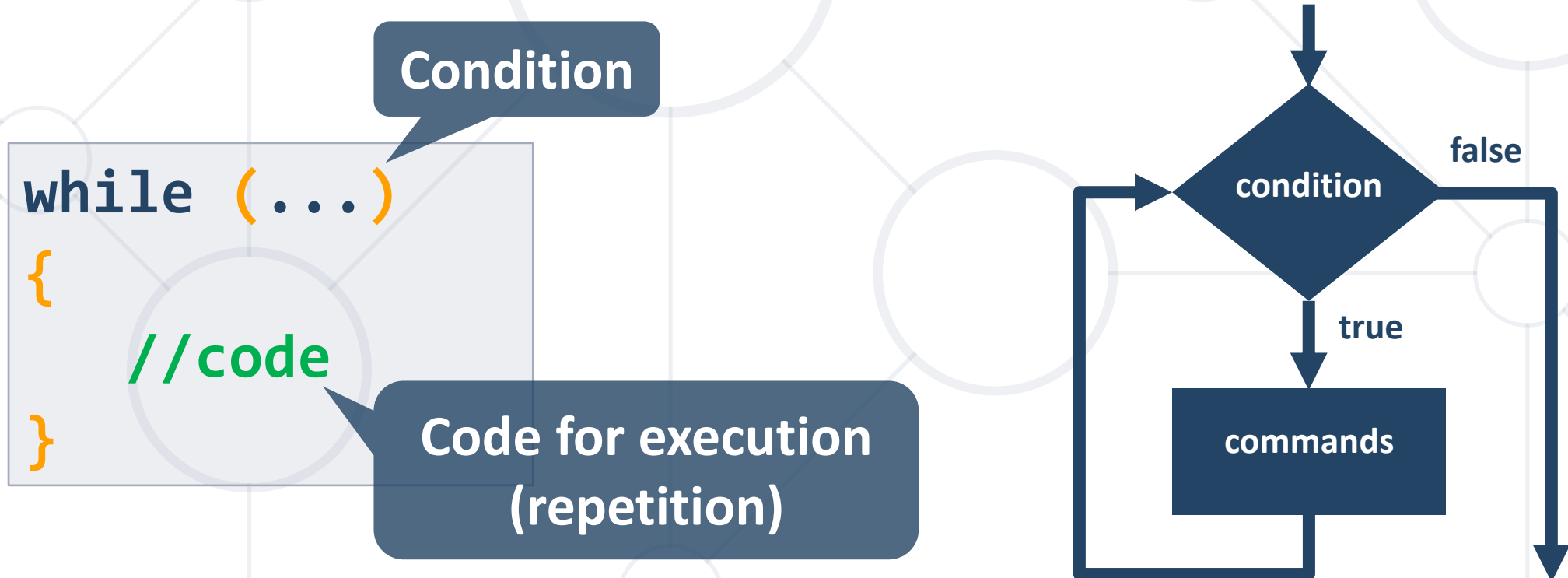
for (int i = 0; i < input.Length; i++)
{
    switch (input[i])
    {
        case 'a': sum += 1; break;
        case 'e': sum += 2; break;
        // TODO: Add cases for the other vowels
    }
}
Console.WriteLine("Vowels sum = " + sum);
```



While Loop

While Loop - Construction

- As we already mentioned, in programming, we often need to execute a block of commands **multiple** times
 - We use **loops** for that purpose – **while**, **for** etc.



While Loop – Example

Break condition

```
int a = 5;  
while (true)  
{  
    if (a > 10) {  
        break;  
    }  
    Console.WriteLine("a = " + a);  
    a++;  
}
```



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

While Loop – Example



```
string input = Console.ReadLine();
```

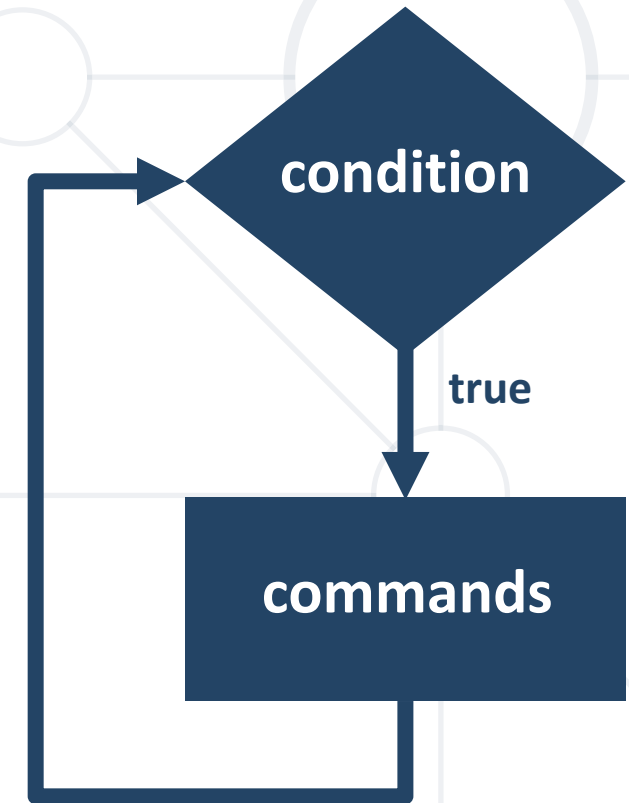
Break condition

```
while (input != "Stop") {  
    Console.WriteLine("Invalid input");  
    input = Console.ReadLine();  
}
```

- **Infinite Loop** – repeating a block of code an infinite number of times:

The condition is
always true

```
while (true)
{
    Console.WriteLine("Infinite loop");
}
```





Interrupting the Loop Using `break`

Infinite Loops

- Operator **break** – discontinues the loop
- It cannot exist on its own outside of a loop

```
while (true)
{
    Console.WriteLine("Infinite loop");
    if (...)
    {
        break;
    }
}
```

Break condition

Reading a Text - Problem

- Write a console program, that:
 - Reads a text input from the console (string)
 - Reading is discontinued when the command "Stop" is received
- Example input and output:



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

```
Nakov  
SoftUni  
Sofia  
Bulgaria  
SomeText
```

```
while (true)
{
    string input = Console.ReadLine();
    if (input == "Stop")
    {
        break;
    }
    Console.WriteLine(input);
}
```

- Write a console program, that:
 - Takes a user's username and password as a string input from the console
 - Reads an access password and verifies if it is correct
 - If:
 - An invalid password is entered, the program reads a new one
 - A correct password is entered, the program breaks the execution

```
string username = Console.ReadLine();  
string password = Console.ReadLine();  
  
string input = Console.ReadLine();  
while (input != password)  
{  
    input = Console.ReadLine();  
}  
  
Console.WriteLine($"Welcome: {username}!");
```

Sum of Numbers - Problem

- Write a console program that:
 - Reads an integer value from the console, this will be the target sum
 - Starts reading and sums integer values from the console
 - Ends reading when the target sum is reached
 - Print on the console the calculated sum
- Example input and output:



100
10
20
30
45

105

20
1
2
3
4
5
6

21

Sum of Numbers - Solution

```
int n = int.Parse(Console.ReadLine());  
int sum = 0;  
  
while (sum < n)  
{  
    int currentNum =int.Parse(Console.ReadLine());  
    sum += currentNum;  
}  
Console.WriteLine(sum);
```

Sequence of numbers $2k + 1$ Problem

- Write a console program, that:
 - Reads an integer number - **n**
 - Prints all numbers $\leq n$ from the sequence: 1, 3, 7, 15, 31, ...
 - Each subsequent number is equal to the **previous** * **2 + 1**

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

Read input

$k = 1$

$k \leq n$

false

End loop

true

Print k

$k = 2 * k + 1$



Sequence of Numbers $2k + 1$ - Solution

```
int n = int.Parse(Console.ReadLine());  
int k = 1;  
while (k <= n)  
{  
    Console.WriteLine(k);  
    k = 2 * k + 1;  
}
```

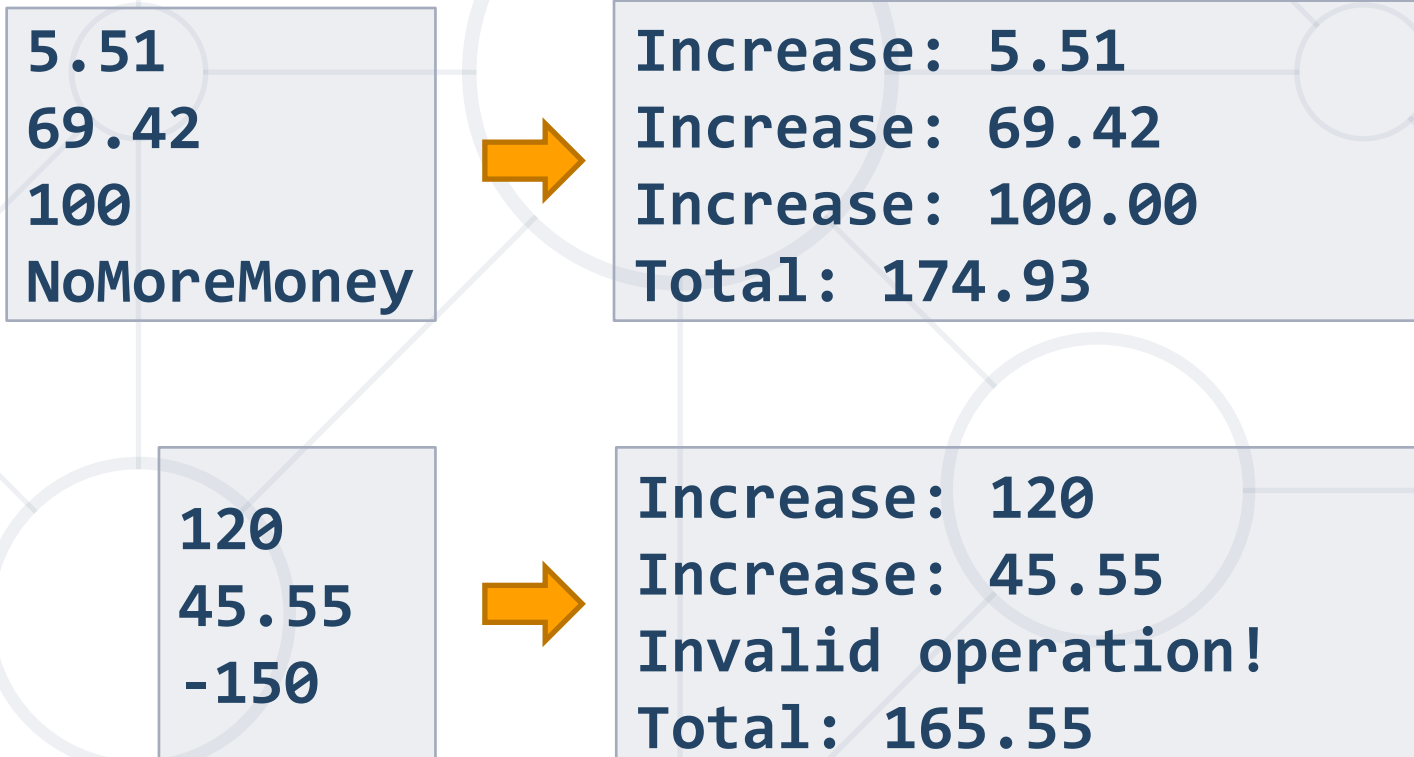
Repeat until $k \leq n$ is a valid condition

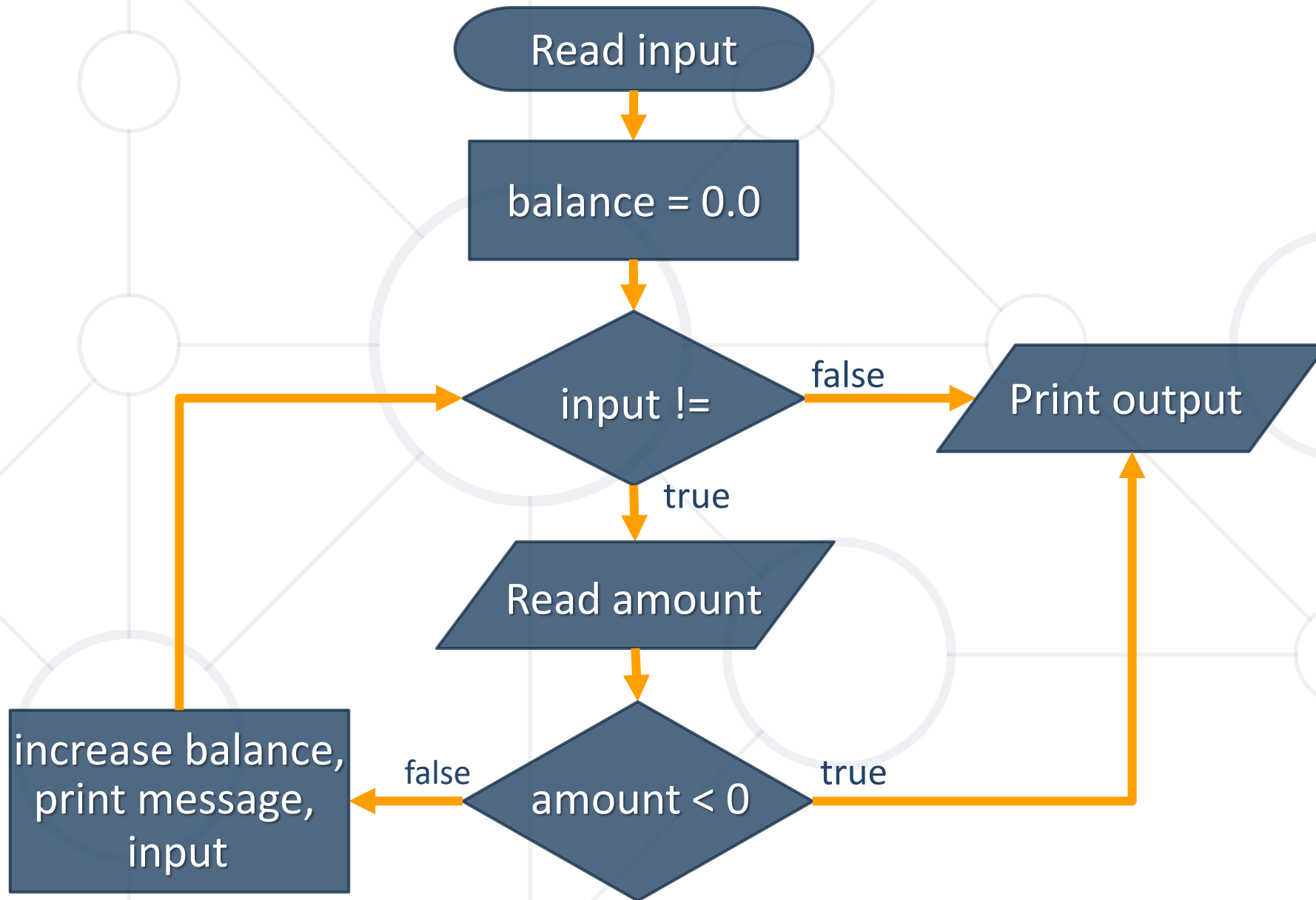
Account Balance - Problem

- Write a console program, that:
 - Reads **n** - count numbers that represent payments to a bank account until the following command is received: **"NoMoreMoney"**
 - Upon every payment is displayed the following message:
"Increase: {sum}"
 - If a negative value is entered, the following message is displayed:
"Invalid operation!" The program stops thereafter
 - In the end the following message should be displayed on the console:
"Total: {accountBalance}"

Account Balance - Description

- Example input and output:





Account Balance - Solution

```
string input = Console.ReadLine();
double balance = 0.0;
while (input != "NoMoreMoney")
{
    double amount = double.Parse(input);
    if (amount < 0) { //TODO: Print output and exit the loop }
    balance += amount;
    Console.WriteLine($"Increase: {amount:F2}");
    input = Console.ReadLine();
}
Console.WriteLine($"Total: {balance:F2}");
```

The Greatest Number - Problem

- Write a console program, that:
 - Reads **n** count consecutive numbers until the following command is received: **"Stop"**
 - Finds the greatest number among all numbers
- Example input and output:

```
100
99
80
70
Stop
```



```
100
```

```
-10
20
-30
Stop
```



```
20
```

```
45
-20
7
99
Stop
```



```
99
```

5

3

The Greatest Number - Solution

```
string input = Console.ReadLine();  
int max = int.MinValue;  
while (input != "Stop")  
{  
    int num = int.Parse(input);  
  
    if (num > max)  
    {  
        max = num;  
    }  
    input = Console.ReadLine();  
}  
Console.WriteLine(max);
```


The Smallest Number - Problem

- Write a console program that:
 - Reads **n** count consecutive numbers until the following command is received: **"Stop"**
 - Finds the smallest number among all numbers
- Example input and output:

```
100  
99  
80  
70  
Stop
```



```
70
```

```
-10  
20  
-30  
Stop
```



```
-30
```

```
45  
-20  
7  
99  
Stop
```



```
-20
```



The Smallest Number - Solution

```
string input = Console.ReadLine();  
int min = int.MaxValue;  
while (input != "Stop")  
{  
    //TODO: Use logic similar  
    to the previous problem  
}
```

- Operator **continue** – switches to the next loop iteration

```
int i = 0;
while (i < 10)
{
    if (i % 2 == 0)
    {
        i++;
        continue;
    }
    Console.WriteLine(i);
    i++;
}
```



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

- Write a console program, that:
 - Calculates the **average grade** of a student for the whole session
 - If the annual grade of the student is:
 - ≥ 4.00 , the student continues to the next grade
 - < 4.00 , the student stays in the same grade for another year
 - If the student fails to continue in the next grade, more than once, he will be excluded, and the program stops. Print on the console the name of the student and the grade that he managed to reach
"{studentName} has been excluded at {gradeReached} grade"
 - Is the student managed to **graduate successfully**:
"{studentName} graduated. Average grade: {averageGrade}"

Graduation - Description

Gosho

5
5.5
6
5.43
5.5
6
5.55
5
6
6
5.43
5



Gosho graduated.
Average grade: 5.53

Mimi

5
6
5
6
5
6
6
2
3



Mimi has been
excluded at 8 grade

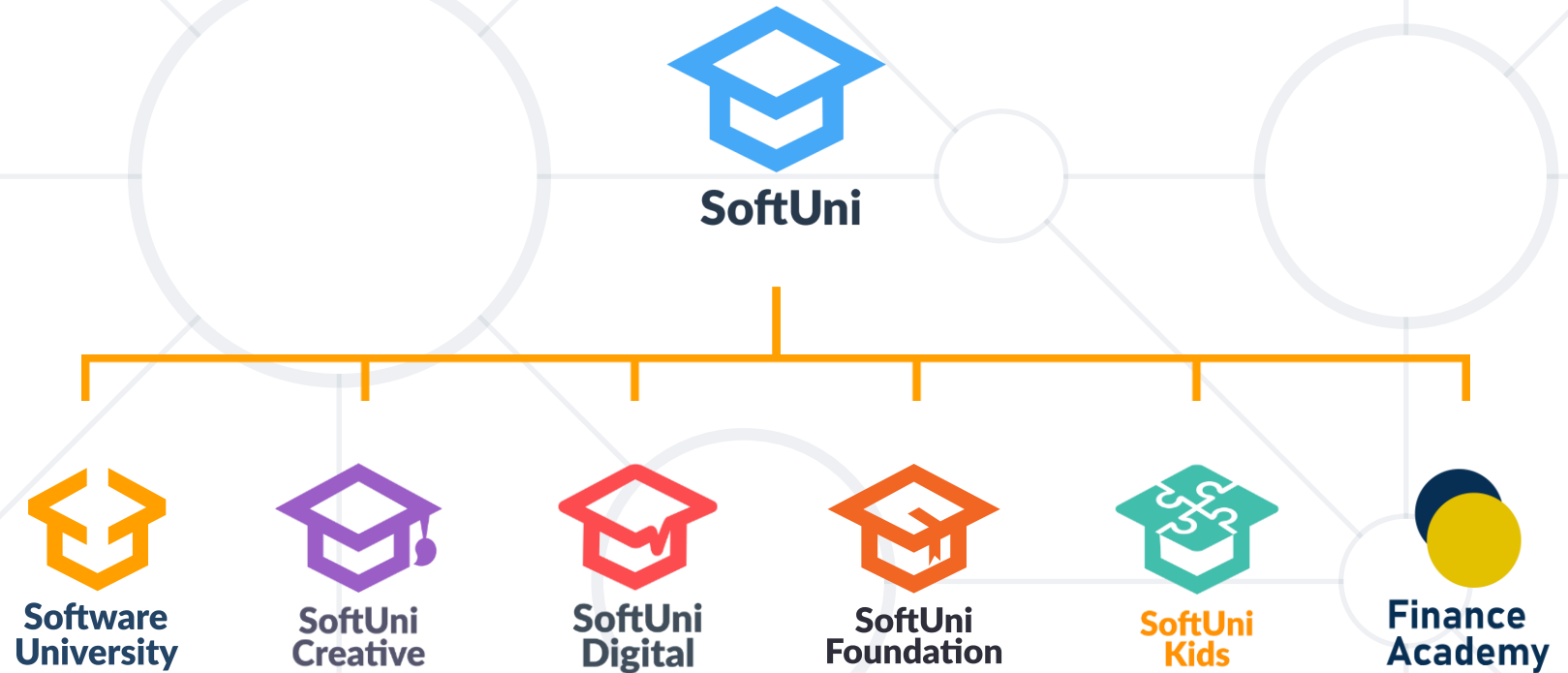
```
string name = Console.ReadLine();
int grades = 1;
double sum = 0;
int excluded = 0;
while (grades <= 12)
{
    double grade = double.Parse(Console.ReadLine());
    if (grade < 4.00) {
        // TODO: increase excluded count and break if is more than 1
        continue;
    }
    // TODO: add grade to sum and increase grades count
}
double average = sum / 12; //TODO: print the output options
```



- **Incrementing** and **decrementing** a variable
- What is a **Loop**?
- **For** Loop
- Working with Text
- **While** Loop
- **Breaking** a Loop



Questions?



SoftUni Diamond Partners

**SUPER
HOSTING
.BG**



**Coca-Cola HBC
Bulgaria**

 **Flutter**TM
International

INDEAVR
Serving the high achievers



AMBITIONED

 **DRAFT
KINGS**



BOSCH

 **Postbank**
Решения за твоето утре

 **PHAR
VISION**

 **SmartIT**

DXC
TECHNOLOGY

createX

- Software University – High-Quality Education, Profession and Job for Software Developers

- softuni.bg, about.softuni.bg

- Software University Foundation

- softuni.foundation

- Software University @ Facebook

- facebook.com/SoftwareUniversity

- Software University Forums

- forum.softuni.bg



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>

