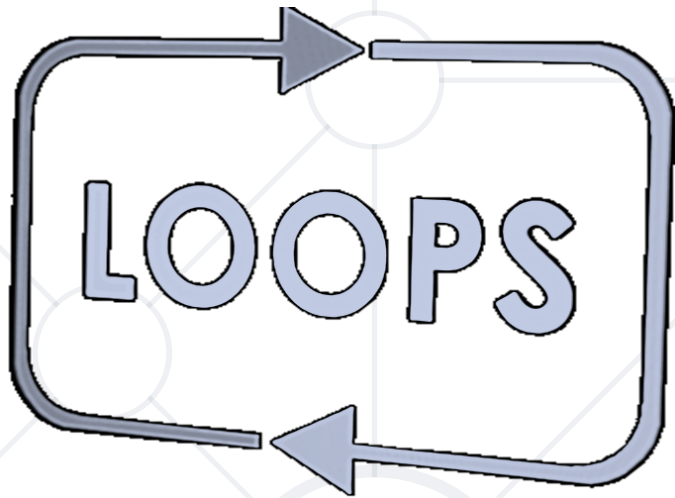


Повторения (цикли)

Прости повторения с For-цикъл



СофтУни

Преподавателски екип



SoftUni



Софтуерен университет
<https://softuni.bg>

1. Преговор
2. Увеличаване и намаляване на стойността на променлива
3. Повторения на блокове код
4. Работа с по-сложни for-цикли
5. Работа с текст
6. Техники за използване на for-цикли





Преговор

1. Каква ще е стойността на променливата **a** след изпълнението на следната програма:

```
int a = 5;  
switch (a)  
{  
    case 5:  
    case 6:  
        a = a + 1;  
        break;  
    default:  
        a = a + 2;  
        break;  
}
```

0

5

7

Отговора 6

2. Какво ще се отпечата на конзолата, ако изпълним следната

КОМАНДА:

```
Console.WriteLine(!(5 == 5) && (4 + 1 == 5));
```

// и двата израза трябва да са true за да върне true при „&&“ /и/ проверка

True

Отговора

False

Runtime
error

Compile time
error

3. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
Console.WriteLine(!(3 == 3) || (3 == 5));
```

True

Runtime
error

Отговора

False

Compile time
error

4. Какво ще се отпечата на конзолата, ако изпълним следната проверка:

```
Console.WriteLine(!(3 > 5) || (1 == 1));
```

```
// при едно true цялото става true
```

Когато VisualStudio ни почертава грешки, – грешка при компилиране/писане на кода, липва нещо, грешен синтаксис/ C# е език които се компилира докато се

пише/ **Compile time
error**

Грешка която възниква по време на изпълнението т.е. Код се компилира без проблем/няма грешен синтаксис/, но възникват грешки при изпълнението/например идват грешки данни/

Runtime error

False

True

5. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

```
int number = 101;  
if (number >= 1) {  
    Console.WriteLine("Larger than 1");  
}  
if (number <= 101) {  
    Console.WriteLine("Less than 101");  
    Console.WriteLine("Equal to 101");  
}  
} //тъй като няма else ще се изпишат и трите Console.WriteLine, те ще се  
  проверят отделно
```

Larger than 1

Less than 101

Equal to 101

Нямаме грешка, ще се изпишат
и 3те отговора

Compile
time error

6. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

```
string role = "Administrator";  
string password = "SoftUni";  
if(role == "SoftUni")  
{  
    if(password == "SoftUni")  
    {  
        Console.WriteLine("Welcome!");  
    }  
}  
// те са вложени проверки, но щом като още първата ни  
// дава false, няма да влезе да проверява и няма да има изход  
// т.е. щом външна проверка връща false - вътрешната не се проверява
```

"Welcome!"

Runtime error

Compile time
error

Отговора

No output



**Увеличаване и намаляване на стойността
на променливи**

- **Инкрементиране** - увеличаването на стойността на дадена променлива
 - Извършва се чрез оператори за инкрементиране: **префиксни** и **постфиксни**
 - Извършва се само върху променливи, които имат числена стойност

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

- **Пре**-инкрементация

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

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

- **Пост**-инкрементация

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

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

- **Декрементиране** – намаляването на стойността на дадена променлива
 - Извършва се чрез оператори за декрементиране: **префиксни** и **постфиксни**
 - Извършва се само върху променливи, които имат числена стойност

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

- **Пре-**декрементация

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

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

- **Пост-**декрементация

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

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



Повторения на блокове код

Конструкция за For-цикъл

Какво е цикъл?

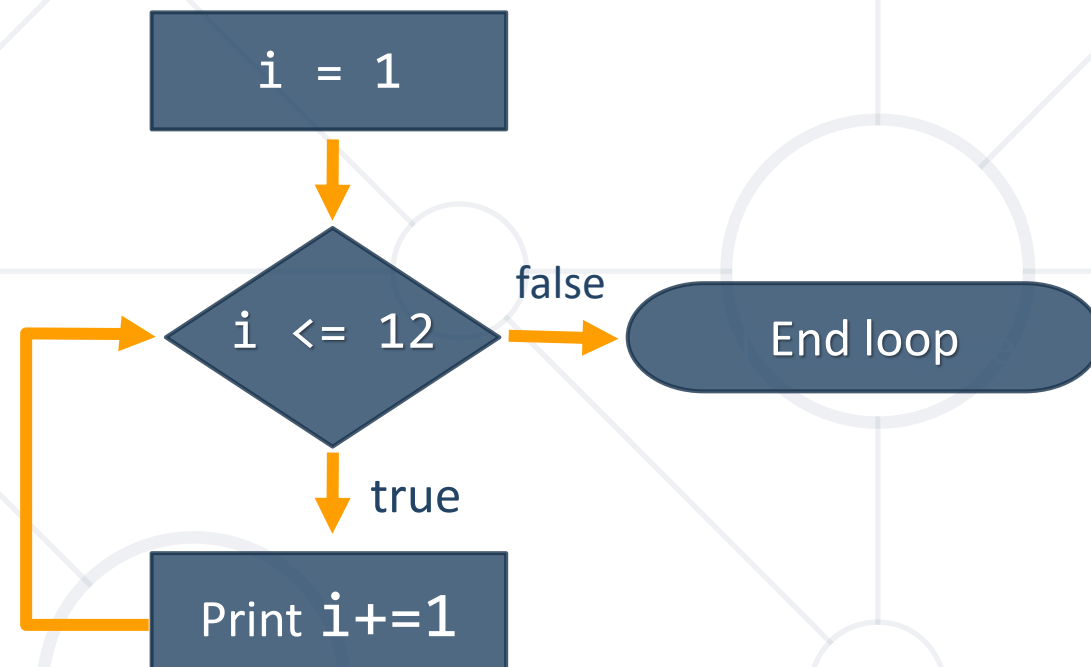
- Често ни се налага да **повтаряме** едно и също действие **многократно**
- Когато сме абитуриенти броим до 12



Какво е цикъл? (2)

- Циклите в програмирането ни позволяват да повтаряме **едни и същи действия** определен брой пъти:

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



- Можем да повтаряме действия до определен момент чрез **for**-цикли

Ключова дума за
конструкцията

Начална
стойност

Крайна
стойност

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

Стъпка

Тяло на цикъла: блок от код за
повторение

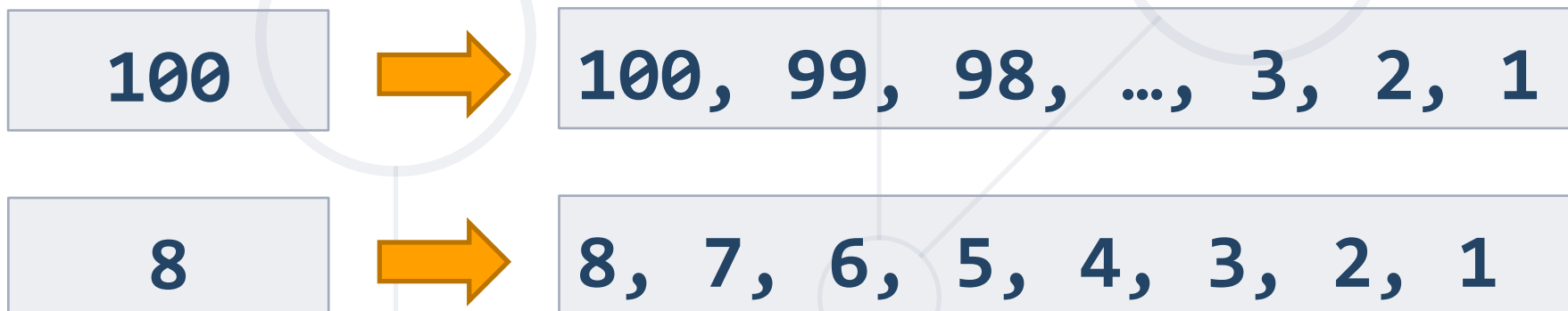


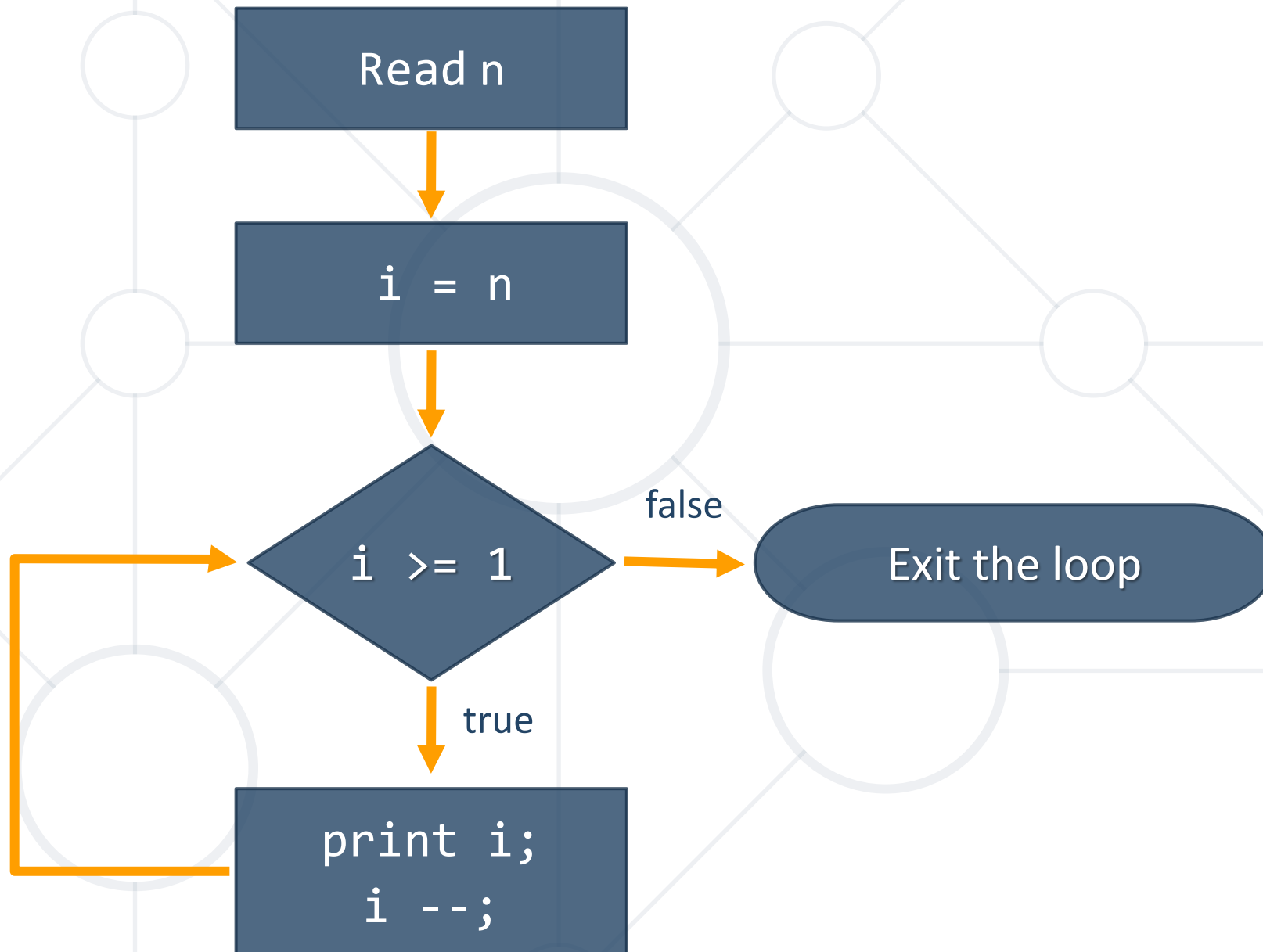
Работа с по-сложни For-цикли

Цикли със стъпка

Числата от N до 1 в обратен ред – условие

- Напишете програма, която:
 - Прочита цяло число **n**
 - Отпечатва числата от **n** до **1** в обратен ред (стъпка -1)
- Примерен вход и изход:





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

Обърнато условие: $i \geq 1$

Намаляваща стъпка: -1

Числата от 1 до N през 3 – условие

- Напишете програма, която:
 - Прочита цяло число **n**
 - Отпечатва числата от **1** до **n** със стъпка **3**
- Примерен вход и изход:



Read n

$i = 1$

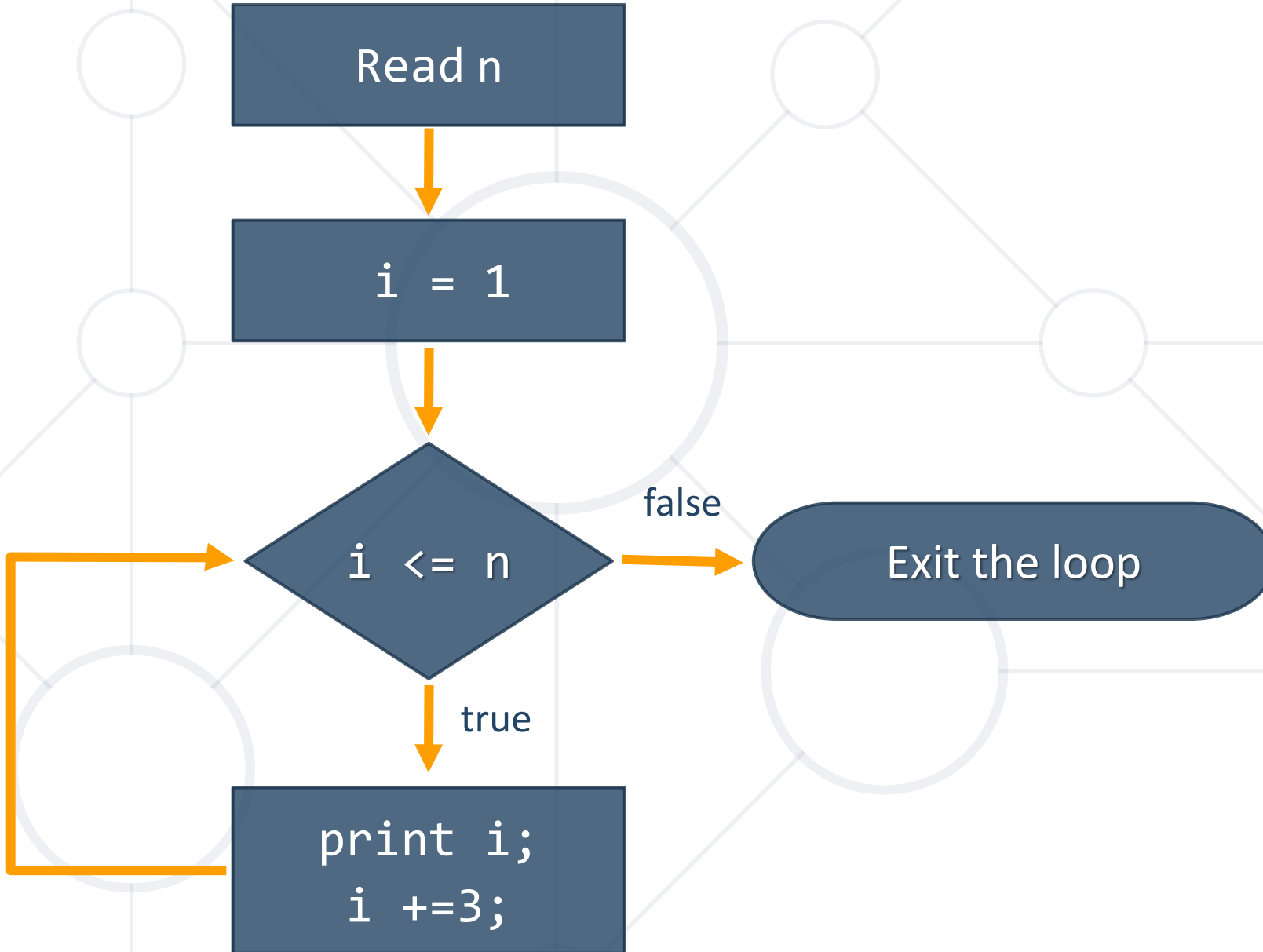
$i \leq n$

false

Exit the loop

true

print i;
 $i += 3;$



Числата от 1 до N през 3 – решение

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

Задаване на
стъпка 3

Четни степени на 2 – условие

- Напишете програма, която:
 - Прочита цяло число **n**
 - Отпечатва четните степени на 2 до **2ⁿ**: $2^0, 2^2, 2^4, 2^8, \dots, 2^n$
- Примерен вход и изход:

10 → 1, 4, 16 , ..., 1024

7 → 1, 4, 16 , ..., 64

Четни степени на 2 – решение

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

Ползваме
стъпка 2



Работа с текст

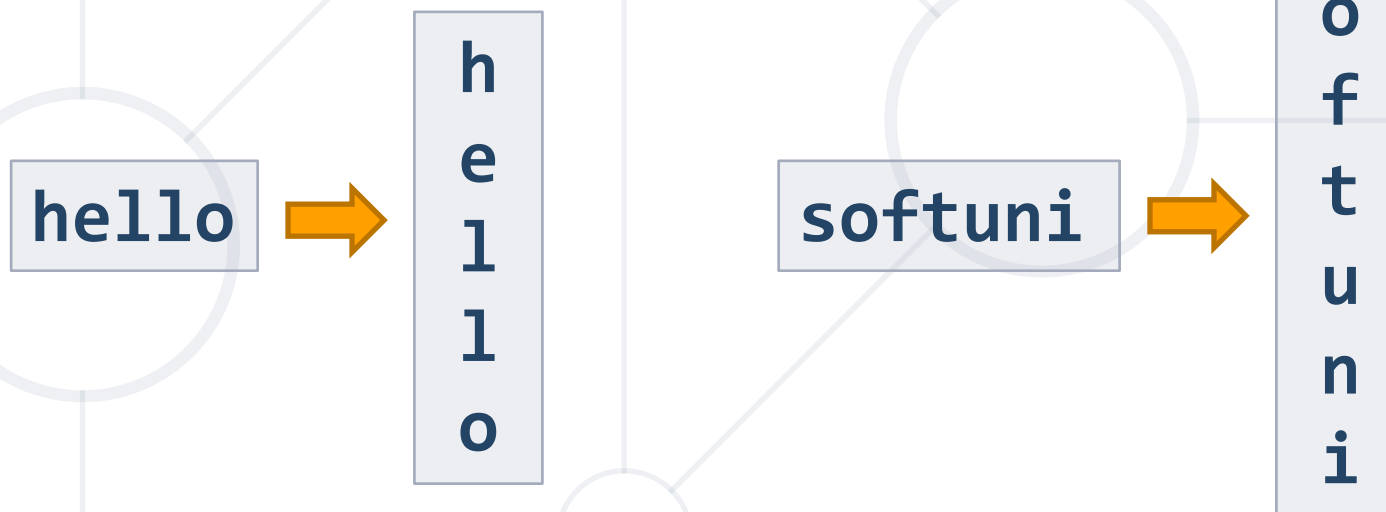
- Можем да вземем дължината на текст

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

- Можем да вземем символ от текст по индекс

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

- Напишете програма, която
 - чете текст (**стринг**)
 - печата всеки **СИМВОЛ** от текста на отделен ред
- Примерен вход и изход:



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

Взимаме дължината
на текста

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

```
{
```

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

Взимаме всеки
символ по индекс *i*

```
}
```

- Напишете програма, която:
 - Прочита от потребителя текст
 - Извежда сумата на гласните букви според таблицата по-долу:

а	е	и	о	и
1	2	3	4	5

- Примерен вход и изход:

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)

Сумиране на гласни букви – решение

```
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);
```



Техники за използване на For-цикли

Задачи с цикли

- Напишете програма, която:
 - Прочита цяло число **n** от потребителя
 - Прочита **n** последователни пъти числа и ги сумира
 - Извежда пресметнатата сума
- Примерен вход и изход:

2
10
20



30

3
-10
-20
-30



-60

4
45
-20
7
11



43

Read input

$i = 1$
 $sum = 0$

$i \leq n$

false

Print the sum

true

Read a number;
Add it to the sum

- Напишете програма, която:
 - Чете **n** на брой цели числа
 - Принтира най-голямото и най-малкото число
- Примерен вход и изход:

```
5
10
20
304
0
50
```

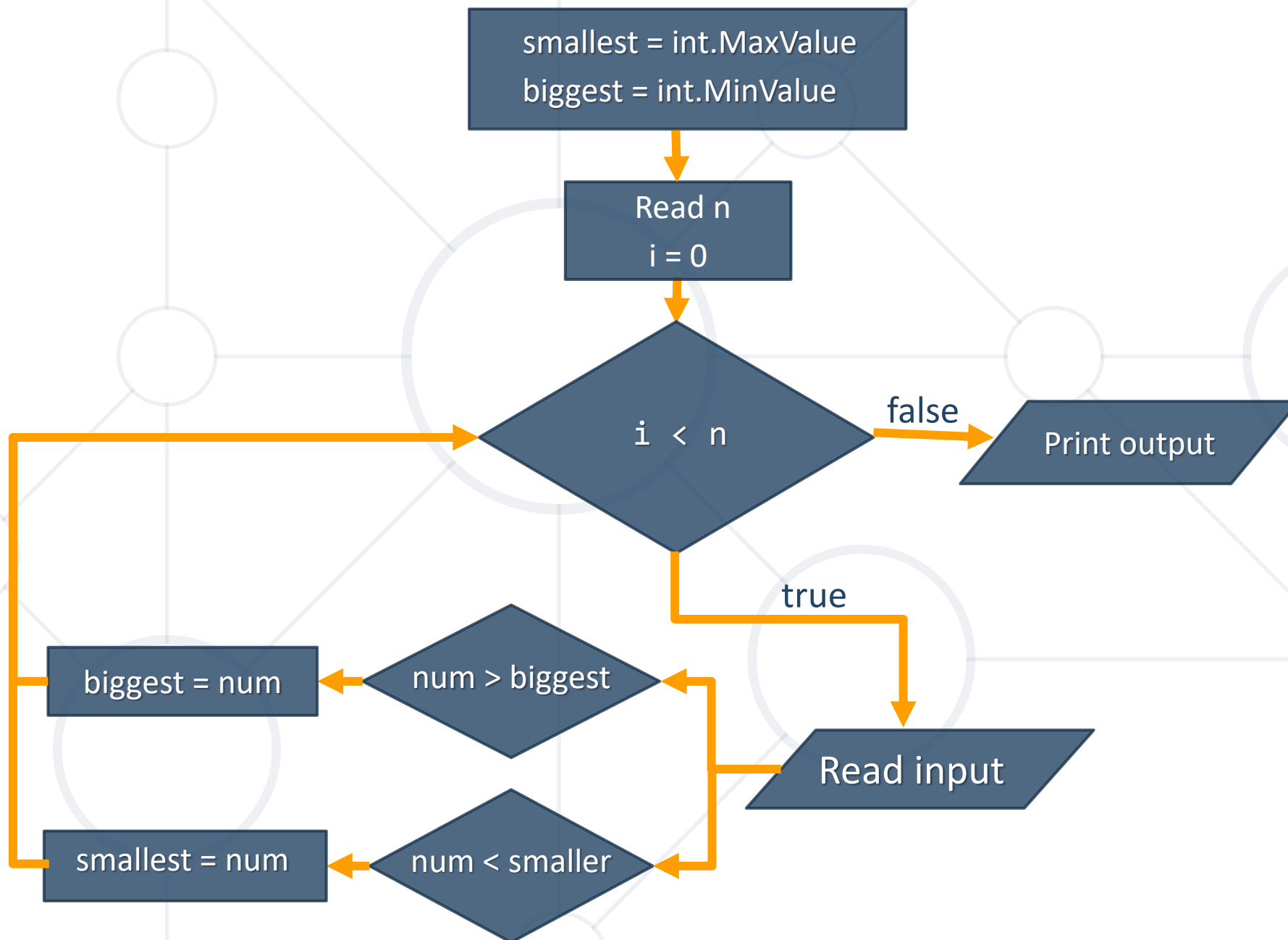


```
Max number: 304
Min number: 0
```

```
5
15
20
350
20
1
```



```
Max number: 350
Min number: 1
```



```
int smallest = int.MaxValue;
int biggest = int.MinValue;
int n = int.Parse(Console.ReadLine());

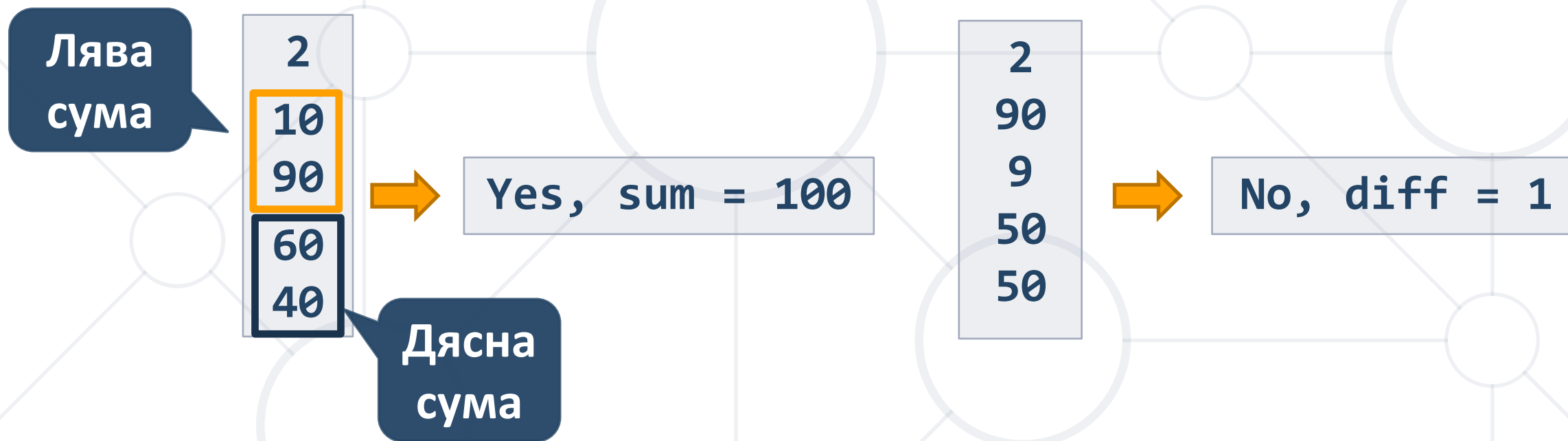
for (int i = 0; i < n; i++)
{
    int num = int.Parse(Console.ReadLine());
    if (num < smallest) smallest = num;
    if (num > biggest) biggest = num;
}

Console.WriteLine($"Max number: {biggest}");
Console.WriteLine($"Min number: {smallest}");
```

1 3 7

- Напишете програма, която:
 - Прочита цяло число n от потребителя
 - Прочита последователно $2*n$ числа
 - Проверява дали сумите на **левите** n и **десните** n числа са равни
 - При равенство извежда "**Yes**" и сумата, в противен случай - "**No**" и разликата (изчислена като положително число)

- Примерен вход и изход:



Решение: лява и дясна сума

```
int n = int.Parse(Console.ReadLine());
int leftSum = 0;
for (int i = 1; i <= n; i++)
{
    leftSum = leftSum + int.Parse(Console.ReadLine());
}
// TODO: read and calculate the rightSum
if (leftSum == rightSum)
    Console.WriteLine("Yes, sum = " + leftSum);
else
    int diff = Math.Abs(rightSum - leftSum);
    Console.WriteLine("No, diff = " + diff);
```

- Напишете програма, която:
 - Прочита цяло число(**n**) от потребителя
 - Прочита последователно **n** на брой числа
 - Проверява дали сумата на числата на **четни** позиции е равна на сумата на числата на **нечетни** позиции
 - При равенство печата "**Yes**" и сумата; иначе печата "**No**" и разликата (положително число).

Четна / нечетна сума – условие

- Примерен вход и изход:

4
10
50
60
20



Yes
Sum = 70

4
3
5
1
-2



No
Diff = 1

3
5
8
1



No
Diff = 2

Решение: четна / нечетна сума

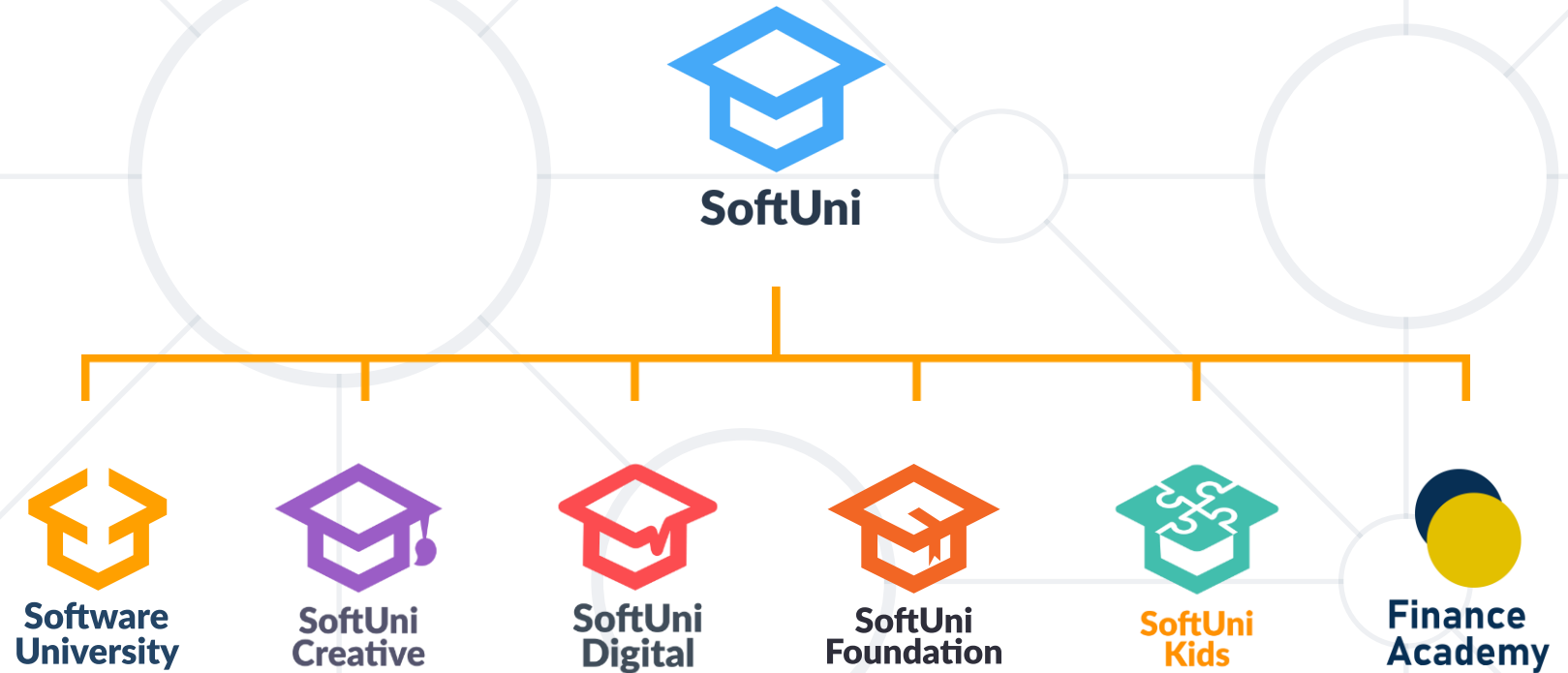
```
int n = int.Parse(Console.ReadLine());
int oddSum = 0;
int evenSum = 0;
for (int i = 1; i <= n; i++)
{
    int element = int.Parse(Console.ReadLine());
    if (i % 2 == 0) evenSum += element;
    else oddSum += element;
}

// TODO: print the sum / difference
```

- Повторение на блок код с **for**-цикъл
- Цикли със стъпка
 - Цикли с увеличаваща стъпка
 - Цикли с намаляваща стъпка
- Вземане на символ по индекс от текст



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
- Фондация "Софтуерен университет"
 - softuni.foundation
- Софтуерен университет @ Facebook
 - facebook.com/SoftwareUniversity

