

Сложни конструкции за повторение

Повторения с различни стъпки, while



Учителски екип

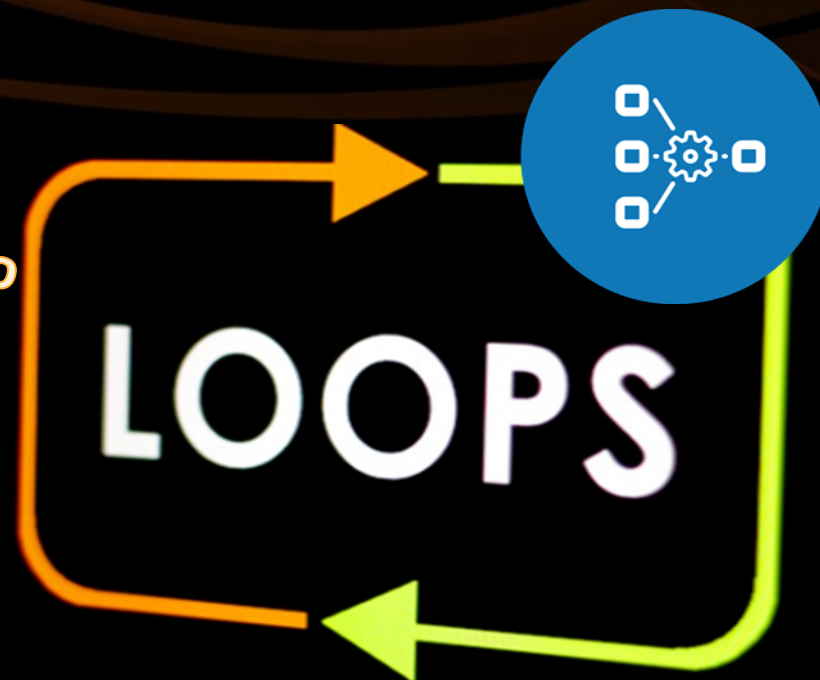
Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>

<https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Basics>



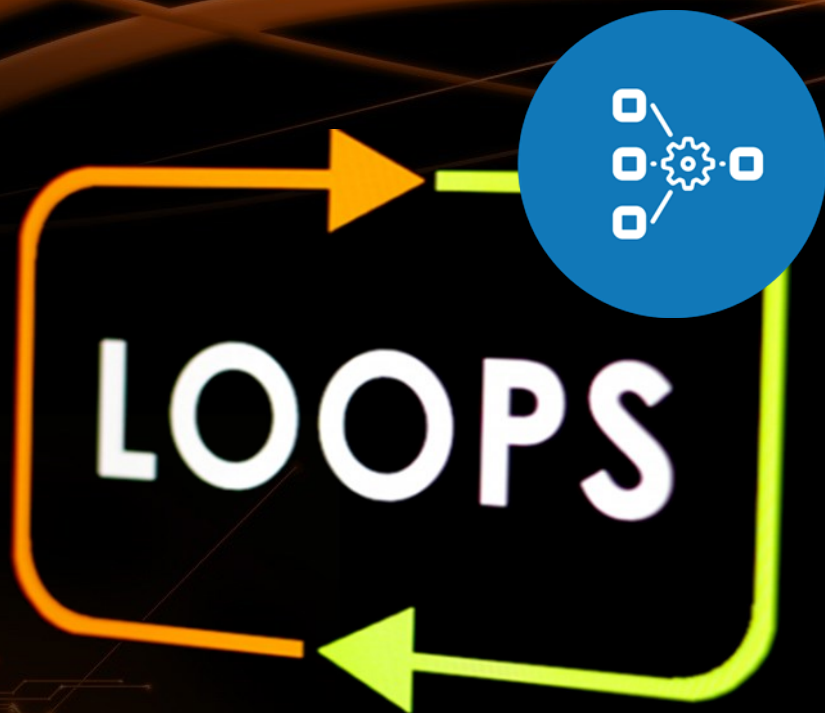
Увод в
програмирането



Съдържание

- Сложни конструкции за повторение:
 - Повторение със стъпка
 - Повторение с намаляваща стъпка
 - Конструкция **while**
- По-сложни задачи с повторения
- Най-голям общ делител (НОД)





Повторения със стъпка
Работа с по-сложни for-цикли

Числата от 1 до N през 3

- Да се отпечатаат числата от **1** до **n** със **стъпка 3**
 - При **n** = 100: **1, 4, 7, 10, ..., 94, 97, 100**

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



Задаване
на стъпка

Тествайте в Judge:

<https://judge.softuni.bg/Contests/2640/Повторения-с-различни-стъпки>

Числата от N до 1 в обратен ред

- Да се отпечатаат числата от **n** до **1** в обратен ред (стъпка -1)
 - При **n** = 100: 100, 99, 98, ..., 3, 2, 1

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

Обърнато
условие:
i >= 1

Отрицателна
стъпка: **-1**



Тествайте в Judge:

<https://judge.softuni.bg/Contests/2640/Повторения-с-различни-стъпки>

Числата от 1 до $2N$ с For-цикъл

- Да се отпечатаат числата от **1** до 2^n
 - При $n = 10$: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024

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

8	2	8	
16	128	4	2
2	4	2	4
2	32	8	

Тествайте в Judge:

<https://judge.softuni.bg/Contests/2640/Повторения-с-различни-стъпки>

Четни степени на 2

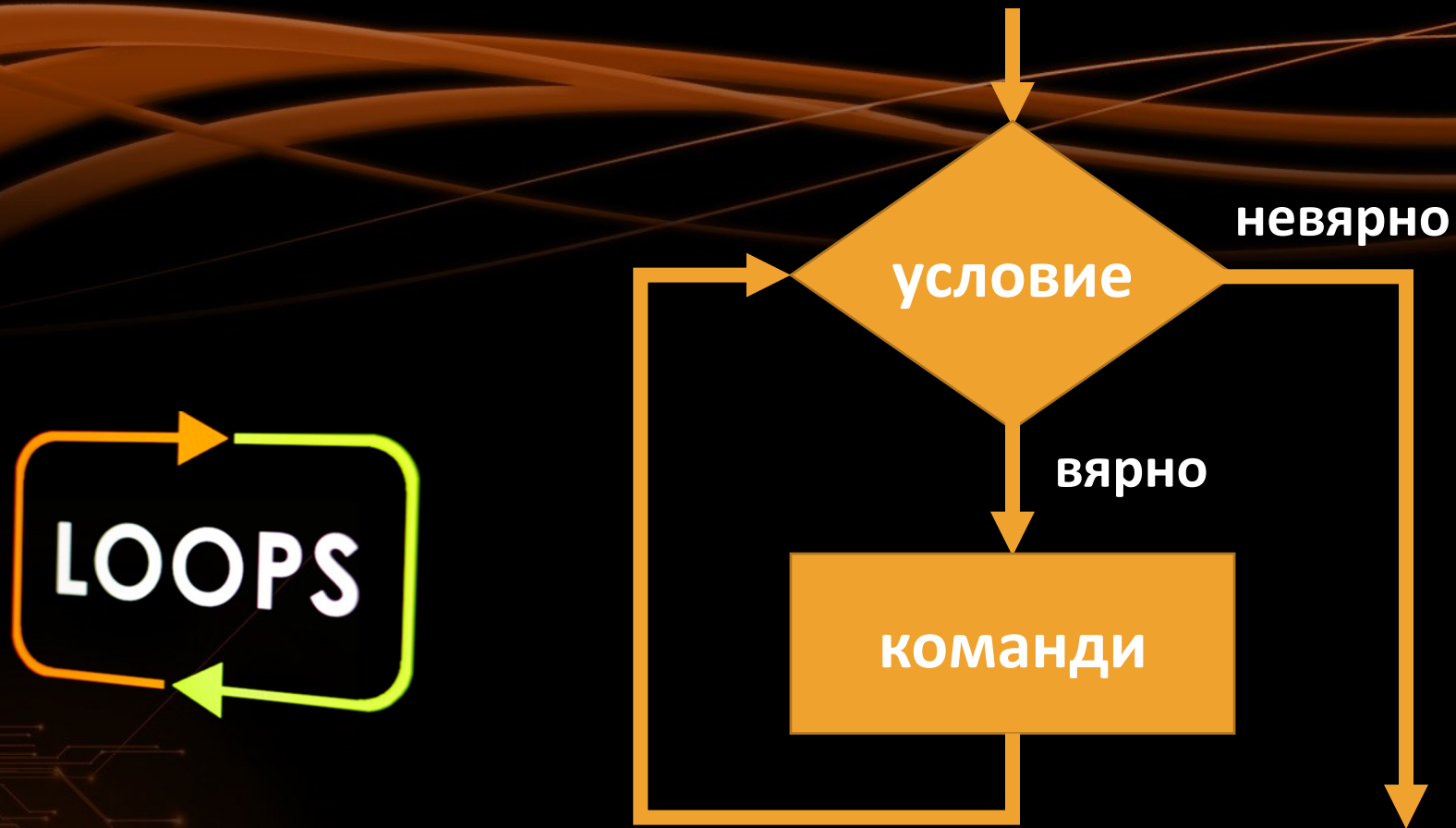
- Да се отпечатаат четните степени на 2 до 2^n : $2^0, 2^2, 2^4, 2^8, \dots, 2^n$
 - При $n = 10$: 1, 4, 16, 64, 256, 1024

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

Ползваме
стъпка 2

Тествайте в Judge:

<https://judge.softuni.bg/Contests/2640/Повторения-с-различни-стъпки>



Конструкция While

Повторение докато е в сила дадено условие

Редица числа $2K+1$

- Да се отпечатаат всички числа $\leq n$ от редицата: 1, 3, 7, 15, 31, ...
 - Всяко следващо число = предишно число * 2 + 1

```
var num = 1;  
while (num <= n)  
{  
    Console.WriteLine(num);  
    num = 2 * num + 1;  
}
```

Повтаряй докато е в
сила условието $num \leq n$

1, 3, 7, 15, 31, 63, ...

Тествайте в Judge:

<https://judge.softuni.bg/Contests/2640/Повторения-с-различни-стъпки>

Число в диапазона [1...100]

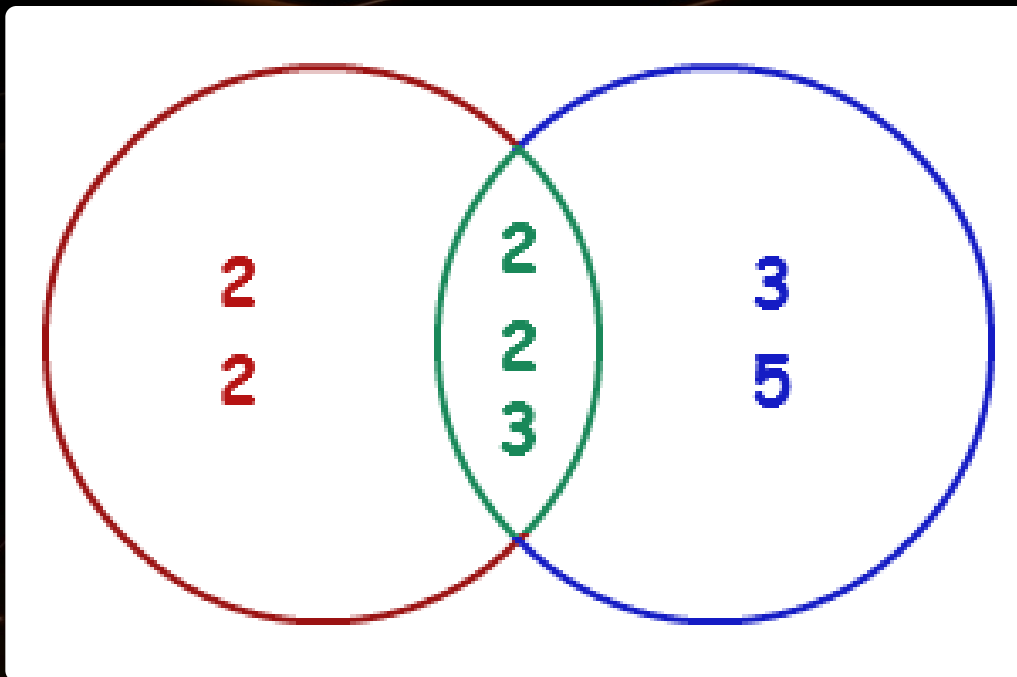
- Да се въведе число в диапазона [1...100]
 - При невалидно число да се въведе отново

```
var num = int.Parse(Console.ReadLine());  
while (num < 1 || num > 100)  
{  
    Console.WriteLine("Invalid number!");  
    num = int.Parse(Console.ReadLine());  
}  
Console.WriteLine("The number is: {0}", num);
```



Тествайте в Judge:

<https://judge.softuni.bg/Contests/2640/Повторения-с-различни-стъпки>



Най-голям общ делител (НОД)

Алгоритъм на Евклид

Най-голям общ делител (НОД)

- Най-голям общ делител (НОД) на две естествени числа **a** и **b** е най-голямото число, което дели едновременно **a** и **b** без остатък
 - $\text{НОД}(24, 16) = 8$
 - $\text{НОД}(12, 24) = 12$
 - $\text{НОД}(10, 10) = 10$
 - $\text{НОД}(67, 18) = 1$
 - $\text{НОД}(15, 9) = 3$
 - $\text{НОД}(100, 88) = 4$
- Алгоритъм на **Евклид** за намиране на НОД:
 - Докато не достигнем остатък 0
 - Делим по-голямото число на по-малкото
 - Вземаме остатъка от делението

```
while b ≠ 0
    var oldB = b;
    b = a % b;
    a = oldB;
print a;
```


Алгоритъм на Евклид за НОД

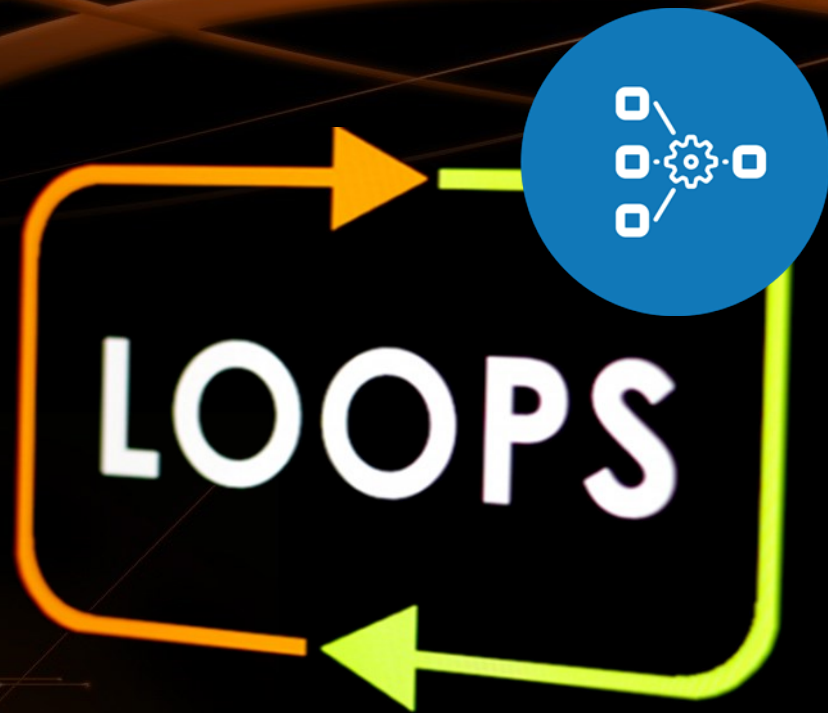
- Да се въведат цели числа **a** и **b** и да се намери **НОД(a, b)**

```
var a = int.Parse(Console.ReadLine());  
var b = int.Parse(Console.ReadLine());  
while (b != 0)  
{  
    var oldB = b;  
    b = a % b;  
    a = oldB;  
}  
Console.WriteLine("GCD = {0}", a);
```



Тествайте в Judge:

<https://judge.softuni.bg/Contests/2640/Повторения-с-различни-стъпки>



Повторения със стъпка и `while`

Работа на живо в клас (лаб)

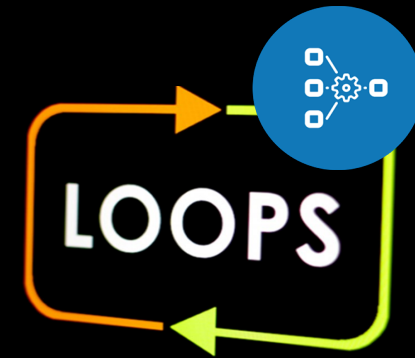
Какво научихме днес?

- Можем да ползваме **for**-цикли със **стъпка**:

```
for (var i = 1; i <= n; i+=3)  
    Console.WriteLine(i);
```

- Конструкция **while** повтаря докато е в сила дадено условие:

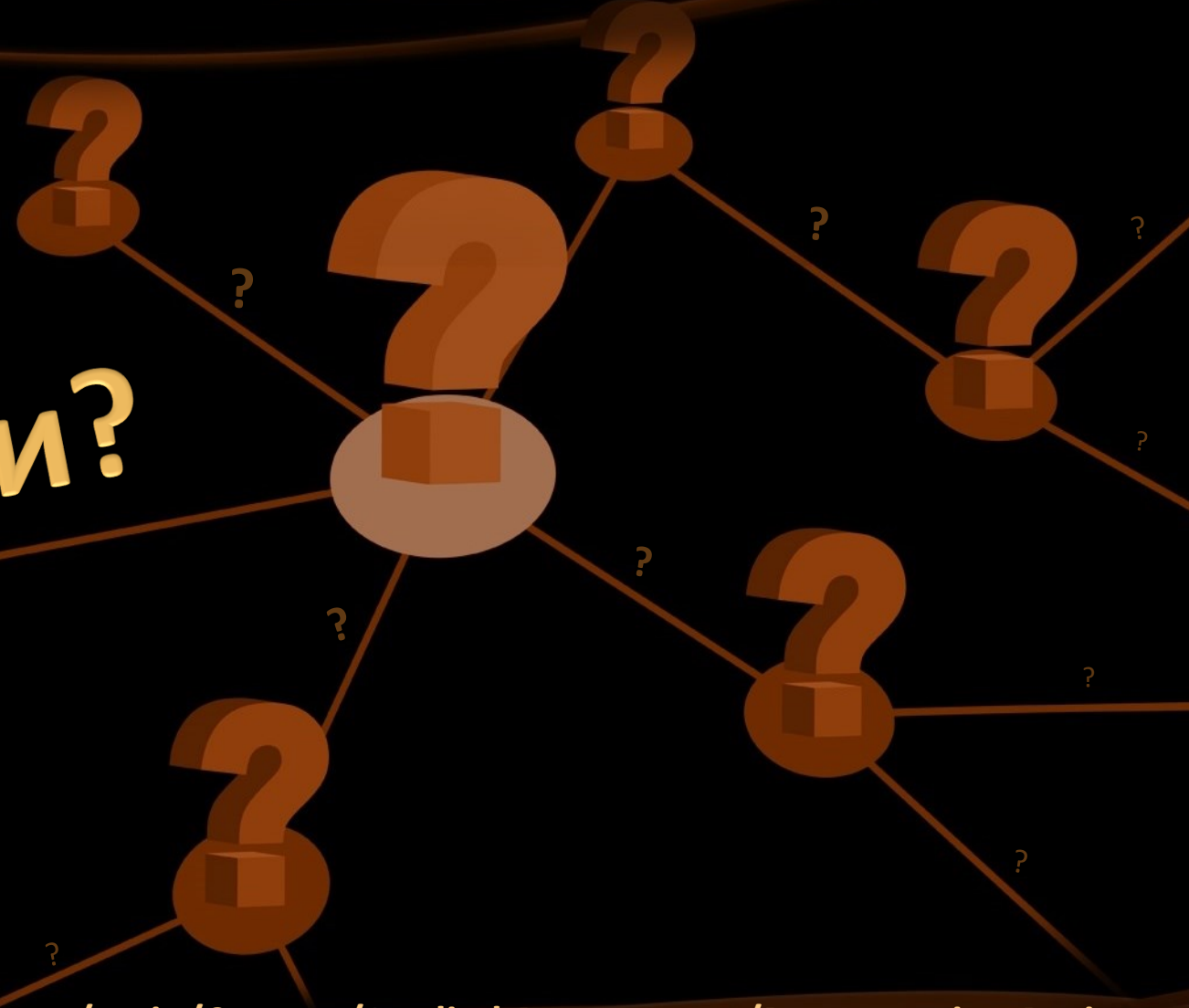
```
int num = 1;  
while (num <= n)  
    Console.WriteLine(num++);
```



Повторения с различни стъпки



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

