

Методи

*Връщане на резултат и
варианти на метод*

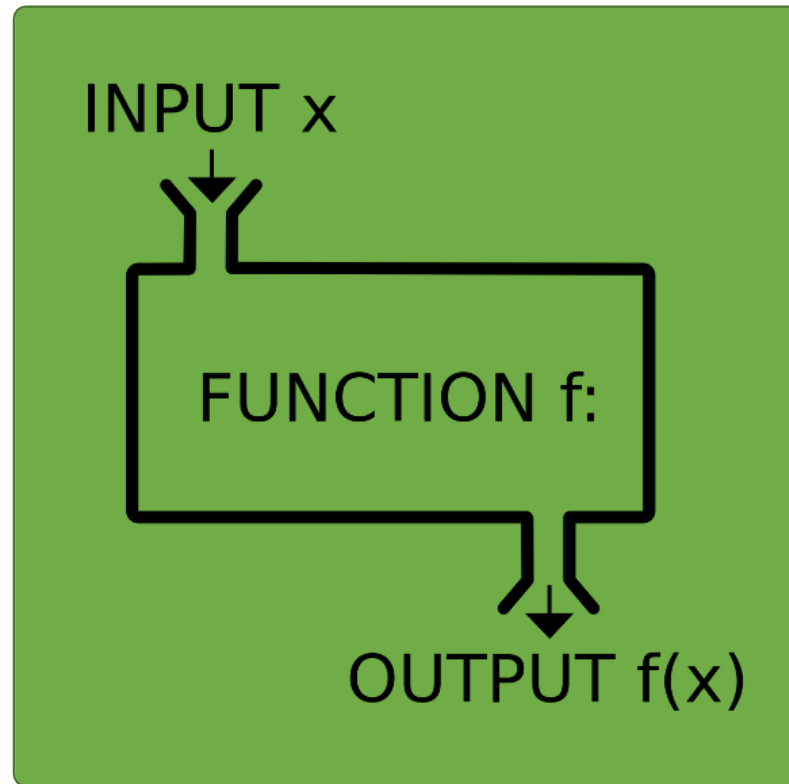
Увод в
програмирането



Съдържание

- Връщане на резултат
- Варианти на метод
- Процес на изпълнение на програма





Връщана стойност от метод

Типове връщана стойност

- Тип **void** – не връща стойност (само изпълнява код)

```
static void AddOne(int n)
{
    n += 1;
    Console.WriteLine(n);
}
```

Липсва
команда **return**

- Друг тип – връщат стойност от тип, съвместим с **типа** на метода

```
static int PlusOne(int n)
{
    return n + 1;
}
```

връща стойност
от тип **int**

Команда `return`

- **`return`** веднага прекъсва изпълнението на метода
- Връща указаната стойност

```
static string ReadFullName()  
{  
    string firstName = Console.ReadLine();  
    string lastName = Console.ReadLine();  
    return firstName + " " + lastName;  
}
```

Връща **string**

- Void методите могат да бъдат **завършени** чрез команда **`return`**

```
return;
```

Употреба на връщаната стойност

- Връщаната стойност може да бъде:

- Присвоена на променлива:

```
int max = GetMax(5, 10);
```

- Включена в израз:

```
decimal total = GetPrice() * quantity * 1.20m;
```

- Подадена на друг метод:

```
int age = int.Parse(Console.ReadLine());
```

Конвертор на температури – пример

- Конвертира температури от Фаренхайт към Целзий:

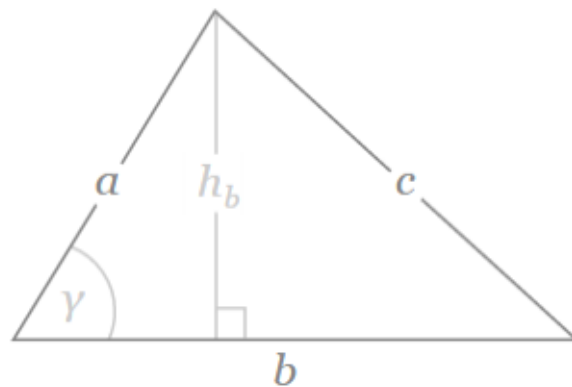
```
static double FahrenheitToCelsius(double degrees)
{
    double celsius = (degrees - 32) * 5 / 9;
    return celsius;
}
```

```
static void Main()
{
    Console.Write("Temperature in Fahrenheit: ");
    double fahrenheit = double.Parse(Console.ReadLine());
    double celsius = FahrenheitToCelsius(fahrenheit);
    Console.Write("Temperature in Celsius: {0:F2}", celsius);
}
```

Задача: Пресмятане на лице на триъгълник

- Създайте метод който пресмята и връща **лицето на триъгълник** по дадени **основа** и **височина**

$$A = \frac{h_b b}{2}$$



$$\begin{aligned} b &= 3 \\ h_b &= 4 \end{aligned}$$



$$A = 6$$

Решение: Пресмятане на лице на триъгълник

- Създайте метод с два параметъра от тип **double**, който връща резултат от тип **double**:

```
static double CalcTriangleArea(double width, double height)
{
    return width * height / 2;
}
```

```
static void Main()
{
    double width = double.Parse(Console.ReadLine());
    double height = double.Parse(Console.ReadLine());
    Console.WriteLine(CalcTriangleArea(width, height));
}
```

Задача: Метод за повдигане на степен

- Създайте метод, който изчислява и връща стойността на **число**, **повдигнато на степен**:

$$2^8 \rightarrow 256$$

$$3^4 \rightarrow 81$$

```
static double RaiseToPower(double number, int power)
{
    double result = 1;
    for (int i = 0; i < power; i++)
        result *= number;
    return result;
}
```


Сигнатура на метод

- Комбинацията от **името** и **параметрите** на метод се нарича негов **сигнатура**

```
static void Print(string text)  
{  
    Console.WriteLine(text);  
}
```

Сигнатура
на метода

- Сигнатурата ни помага да **различим** методи с еднакви имена
- Когато два метода с **едно и също име** имат **различна сигнатура**, това се нарича „**предефиниране**“ на метод


Предефиниране на методи

- Използване на едно и също име за множество методи с различни **сигнатури** (име и параметри на метода)

```
static void Print(string text)
{
    Console.WriteLine(text);
}
```

```
static void Print(int number)
{
    Console.WriteLine(number);
}
```

```
static void Print(string text, int number)
{
    Console.WriteLine(text + ' ' + number);
}
```



Методи с различни
сигнатури

Сигнатура и връщан тип данни

- Типът данни, връщани от метода **не е част** от сигнатурата му
- Разгледайте следния пример – в кода извикваме **Print("test")**:

```
static void Print(string text)
{
    Console.WriteLine(text);
}
```

```
static string Print(string text)
{
    return text;
}
```

Грешка по време
на компилиране!

- Как компилаторът да прецени **кой от двата метода да извика?**

Задача: По-голямото от две числа

- Създайте метод **GetMax()**, който връща по-голямата от две стойности (те могат да са от тип **int**, **char** or **string**)

int
2
16



16

string
aaa
bbb



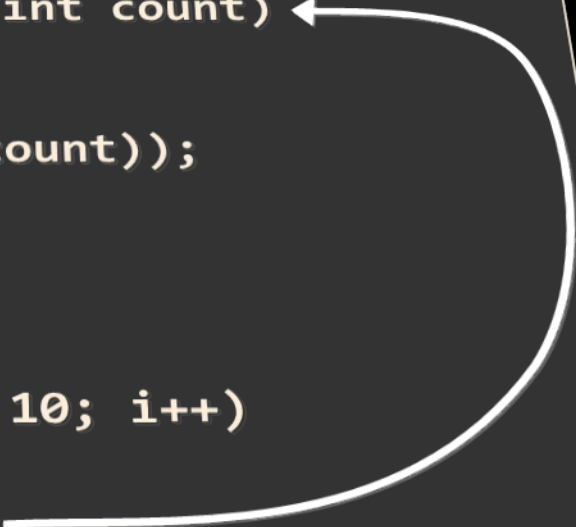
bbb

char
a
z



z

```
static void PrintHyphens(int count) ←  
{  
    Console.WriteLine(  
        new string('-', count));  
}  
  
static void Main()  
{  
    for (int i = 1; i <= 10; i++)  
    {  
        PrintHyphens(i);  
    }  
}
```



Процес на изпълнение
на програма

Изпълнение на програма

- Изпълнението се продължава след извикване на метод:

```
static void Main()
```

```
{
```

```
    Console.WriteLine("before method executes");
```

```
    PrintLogo();
```

```
    Console.WriteLine("after method executes");
```

```
}
```

Първо изпълнение

Извикване на метод

Следващо изпълнение

```
static void PrintLogo()
```

```
{
```

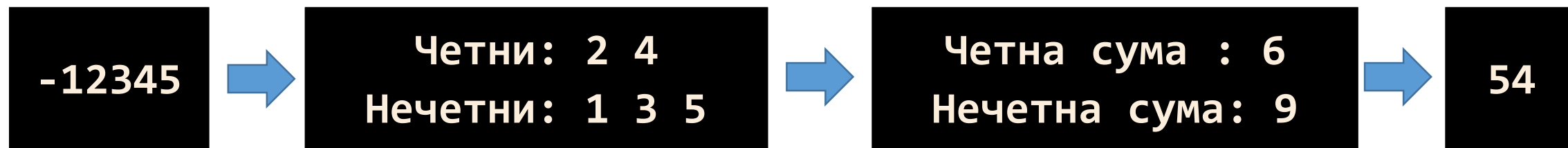
```
    Console.WriteLine("Company Logo");
```

```
    Console.WriteLine("http://www.companywebsite.com");
```

```
}
```

Задача: Умножение на четна и нечетна сума

- Да се напише програма, която **умножава сумата от всички четни цифри** на число **и сумата на всички нечетни цифри** на същото число:
 - Направете метод `GetMultipleOfEvensAndOdds()`
 - Направете методи `GetSumOfEvenDigits()` и `GetSumOfOddDigits()`
 - Използвайте `Math.Abs()` за негативните числа



Какво научихме днес?

- Методите могат да връщат стойност...
- ...или не (тип `void`)
- Методите могат да бъдат предефинирани в различни варианти с едно име
- Какъв е процесът на изпълнение на програма

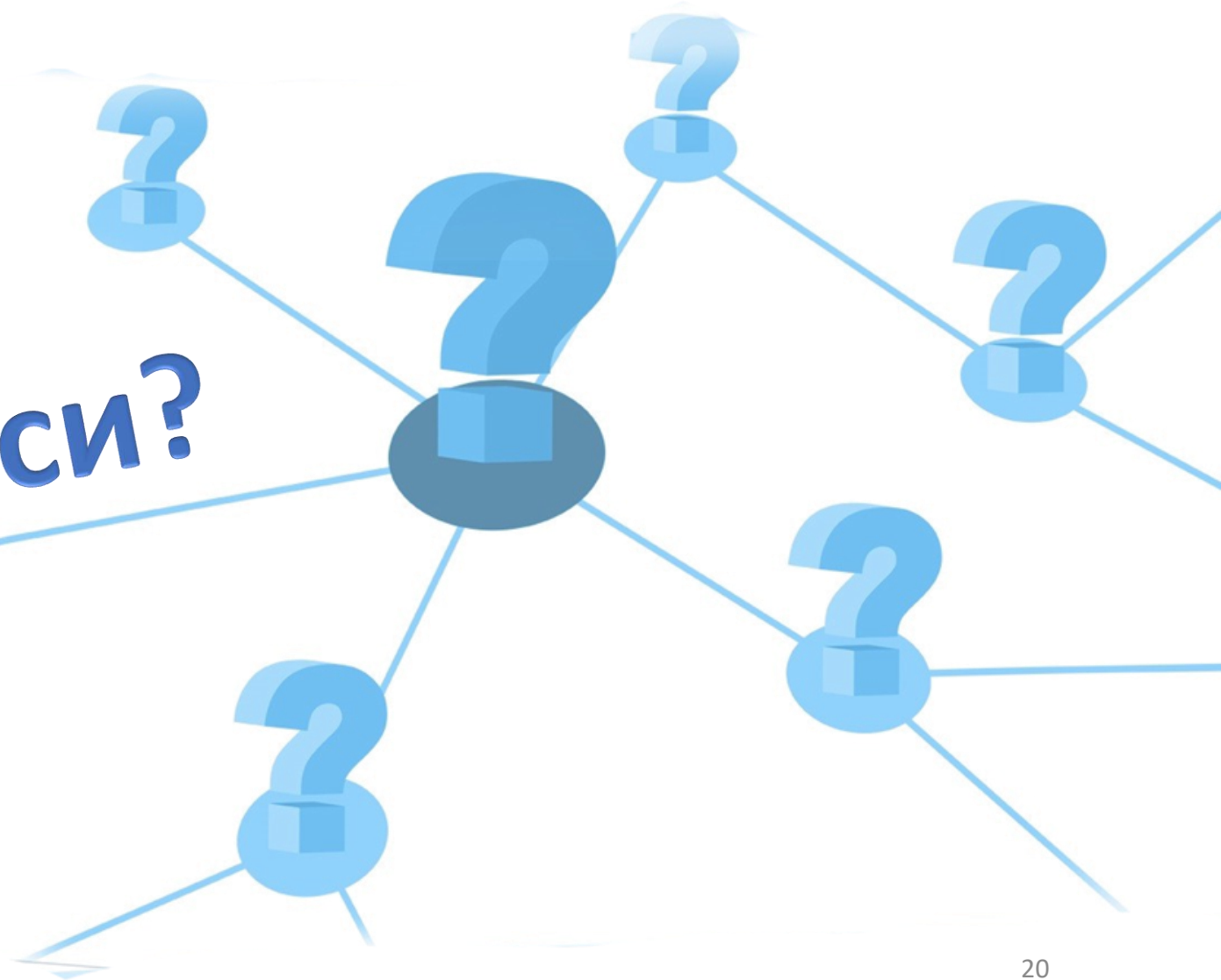


```
static int SumOfDigits(int num)
{
    int sum = 0;
    while (num > 0)
    {
        sum += num % 10;
        num = num / 10;
    }
    return sum;
}
```

Връщане на резултат и варианти на метод



Въпроси?



Договор за ползване

Този курс (слайдове, примери, задачи и др.) се разпространяват под свободен лиценз "[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International](https://creativecommons.org/licenses/by-nc-sa/4.0/)"



Базиран е на учебните материали на [НП „Обучение за ИТ Кариера“](#).

Може да съдържа части от следните източници:

- Книга "[Основи на програмирането със C#](#)" от Светлин Наков и колектив с лиценз [CC-BY-SA](#)