

# Методи

Връщане на резултат и  
варианти на метод



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

1. Връщане на резултат
2. Варианти на метод
3. Процес на изпълнение на програма





**Връщане на резултат  
от метод**

# Типове на връщаната от метода стойност

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

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

Няма оператор  
**return**

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

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

Оператор **return**  
със стойност **int**



# Оператор return

- Веднага спира изпълнението на метода
- Връща определената стойност

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

Връща стойност  
string

- Void методите могат да бъдат **спрени** с използване на **return;**

```
return;
```

# Употреба на връщаната стойност

- Стойностите могат да се:
  - Присвояват на променлива:

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

- Използват в изрази:

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

- Предават директно на друг метод:

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

# Пример: превръщане на температура

- Превърнете температурата от Фаренхайт в Целзий:

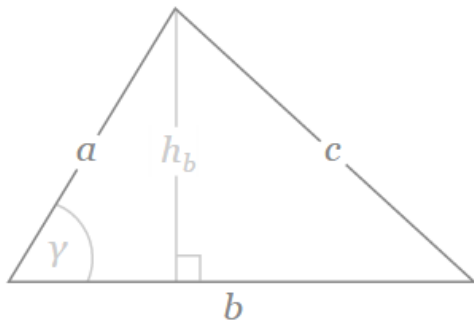
```
static double FahrenheitToCelsius(double degrees)
{
    double celsius = (degrees - 32) * 5 / 9;
    return celsius;
}
```

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

# Задача: Лице на триъгълник

- Да се напише метод, който изчислява лицето на триъгълник по дадени основа и височина и връща стойността му.

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



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



$$A = 6$$



# Решение: Лице на триъгълник

- Направете метод с два **double** параметъра и **double** връщана стойност:

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

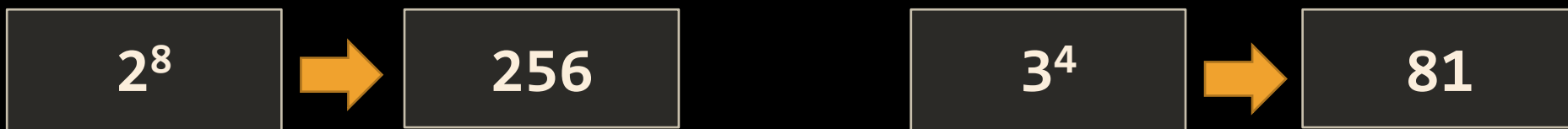
```
static void Main()
{
    double width = double.Parse(Console.ReadLine());
    double height = double.Parse(Console.ReadLine());
    Console.WriteLine(GetTriangleArea(width, height));
}
```

Тествайте в Judge:

<https://judge.softuni.bg/Contests/2644/Връщане-на-резултат-и-варианти-на-метод>

# Задача: Степен на число

- Да се напише метод, който изчислява и връща резултата от повдигането на число на дадена степен



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

Тествайте в Judge:

<https://judge.softuni.bg/Contests/2644/Връщане-на-резултат-и-варианти-на-метод>



# Варианти на методи

# Сигнатура на метода

- Комбинацията от **име** и **параметри** се нарича **сигнатура на метод**

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

Сигнатура  
на метод

- Сигнатурата се използва за **разграничаване** между методи с едно и също име
- Методи с **едно и също име**, но **различна сигнатура** се наричат **варианти на метод**



# Варианти на методи

- Можем да използваме едно име на няколко метода с различни **сигнатури** (име и параметри на метод)

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

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

Различни  
сигнатури

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



# Сигнатура и тип на връщаната стойност

- Типът на връщаната стойност **не е част** от сигнатурата
- Ето един пример:

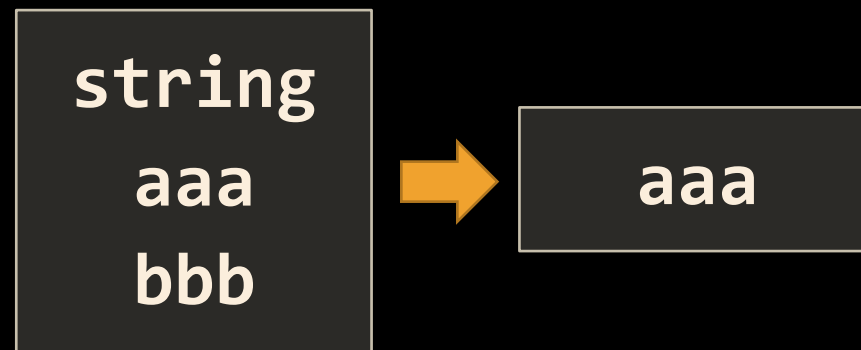
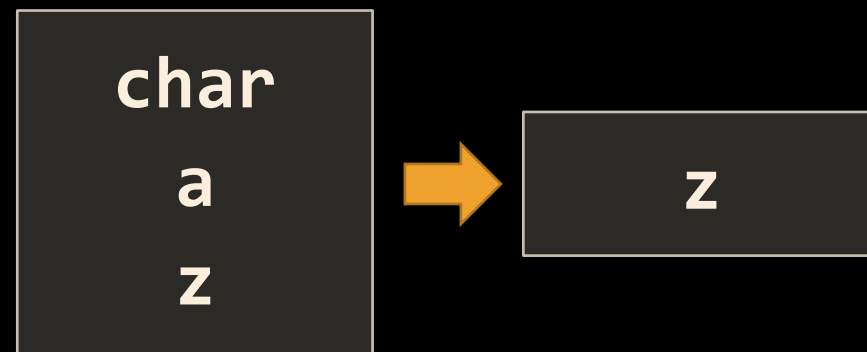
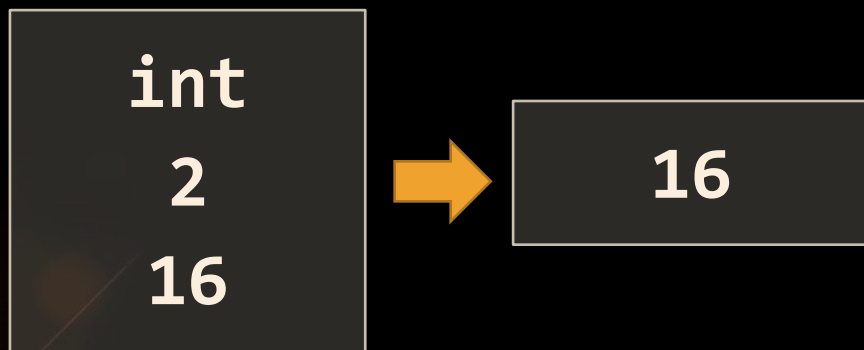
```
static void Print(string text)
{
    Console.WriteLine(text);
}
```

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

- Компиляторът не би могъл да прецени **кой от двата метода да изпълни**

# Задача: По-голямата от две стойности

- Да се създаде метод `GetMax()`, който връща като резултат по-голямата от двете стойности.
- Стойностите могат да бъдат `int`, `char` или `string`.



Тествайте в Judge:

<https://judge.softuni.bg/Contests/2644/Връщане-на-резултат-и-варианти-на-метод>

```
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



Тествайте в Judge:

<https://judge.softuni.bg/Contests/2644/Връщане-на-резултат-и-варианти-на-метод>



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



```
static void PrintHeaderRow(int n)
{
    Console.WriteLine(new
        string('-', 2 * n));
}

static void PrintMiddleRow(int n)
{
    Console.Write('-');
    for (int i = 1; i < n; i++)
        Console.Write("\\\\");
    Console.WriteLine('-');
}
```

# Връщане на резултат и варианти на метод

## Работа на живо в клас (лаб)

# Какво научихме днес?

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

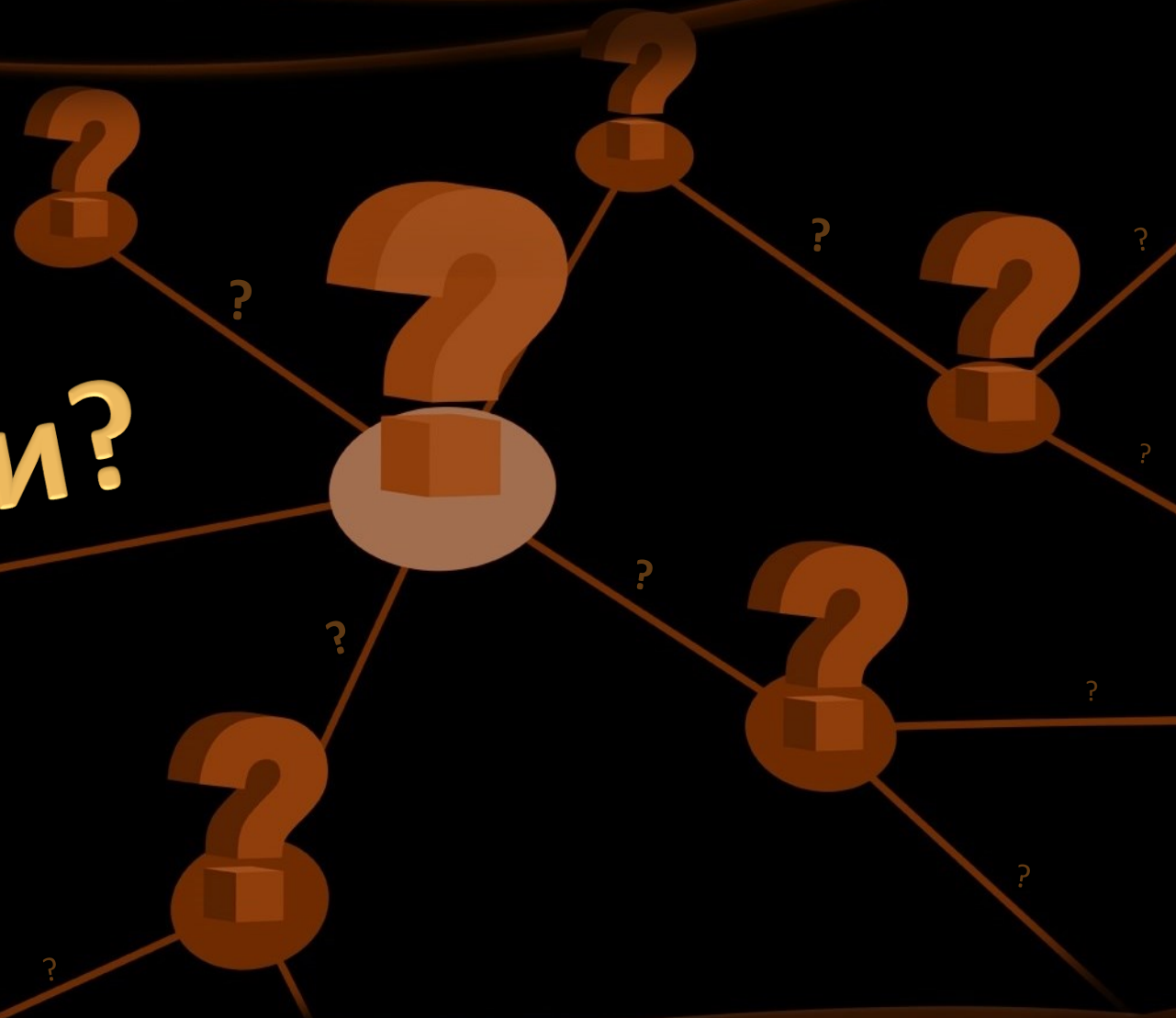


```
static int SumOfDigits(int num)
{
    int sum = 0;
    while (num > 0)
    {
        sum += num % 10;
        num = num / 10;
    }
    return sum;
}
```

# Връщане на резултат и варианти на метод



Въпроси?





# 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

