

Национална програма  
"Обучение за ИТ умения и кариера"  
<https://it-kariera.mon.bg>

Министерството на  
образованието и науката  
<https://www.mon.bg>



# Методи

Деклариране и извикване на методи

# Съдържание

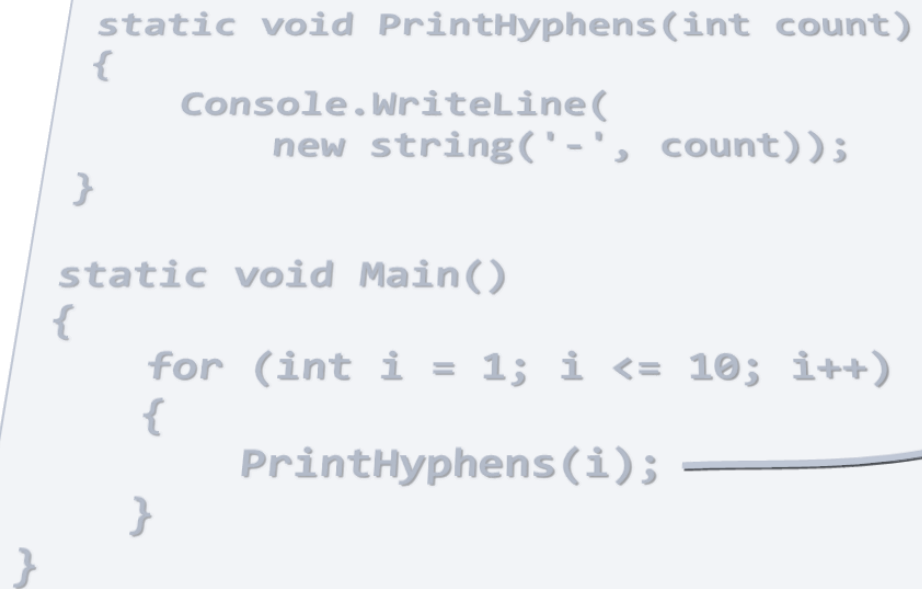
## 1. Използване на методи

- Какво е метод?
- Защо използваме методи?
- Деклариране на методи
- Извикване на методи

## 2. Методи с параметри

- Използване на параметри в методите

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

A diagram showing a call from the Main method to the Printhyphens method. A curved arrow originates from the Printhyphens(i) call inside the Main method's for loop and points back to the Printhyphens method signature at the top of the code block.

# Деклариране и извикване на методи

# Прости Методи

- Методите са именувано парче код, което може да се извика
- Деклариране на прост метод:

```
static void PrintHeader()  
{  
    Console.WriteLine("-----");  
}
```

Име на метода: PrintHeader

Тялото на метода  
винаги е в { }

- Извикване на метода няколко пъти:

```
PrintHeader();  
PrintHeader();
```

# Защо да използваме методи?

- Управляваме процеса на програмиране
  - Разделяме големи програми на малки части
  - По-добра организация на програмата ни
  - Подобрява четимостта на кода
  - Подобрява разбираемостта на кода
- Избягваме повторението на програмен код
  - Подобрява поддръжката на кода
- Преизползваемост на кода
  - Използване на съществуващи методи няколко пъти

# Деклариране на методи

Тип на връщаната стойност

Име на метода

Параметри

```
static double GetSquare(double num)
{
    return num * num;
}
```

Тяло

- В C#, методите се декларират вътре в клас
- Main() също е метод
- Декларираните променливи са локални

```
class MyProgram
{
    static void Main(string[] args)
    {
    }
}
```

# Извикване на метод [1/2]

- Методите могат да бъдат извикани чрез името им

```
static void PrintHeader()  
{  
    Console.WriteLine("-----");  
}
```

Декларация  
на метод

- Извикване на метод:

```
static void Main()  
{  
    PrintHeader();  
}
```

Извикване  
на метод

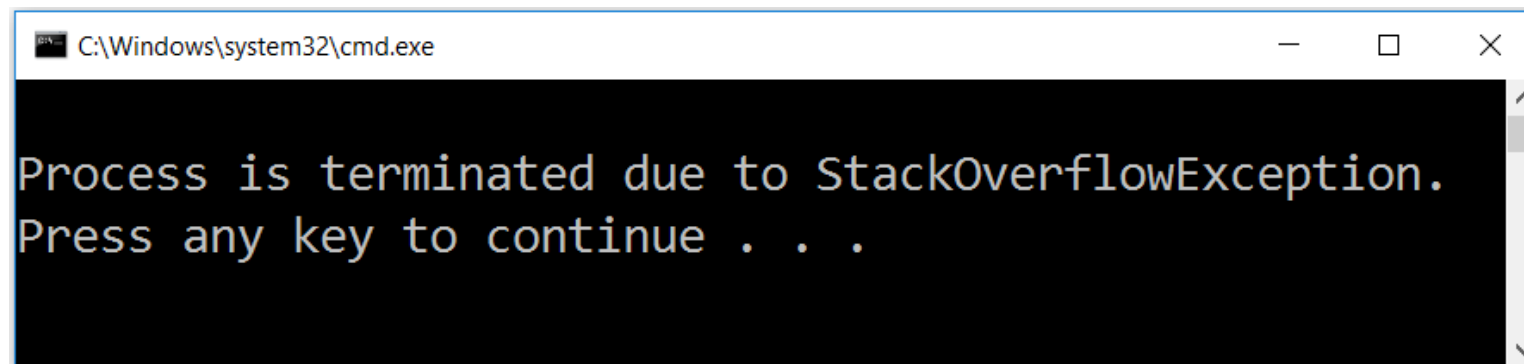
# Извикване на метод [2/2]

- Метод може да бъде извикан от:
  - Главният метод – Main()
  - Някой друг метод

```
static void Main()  
{  
    PrintHeader();  
}
```

```
static void PrintReceipt()  
{  
    PrintHeader();  
    PrintFooter();  
}
```

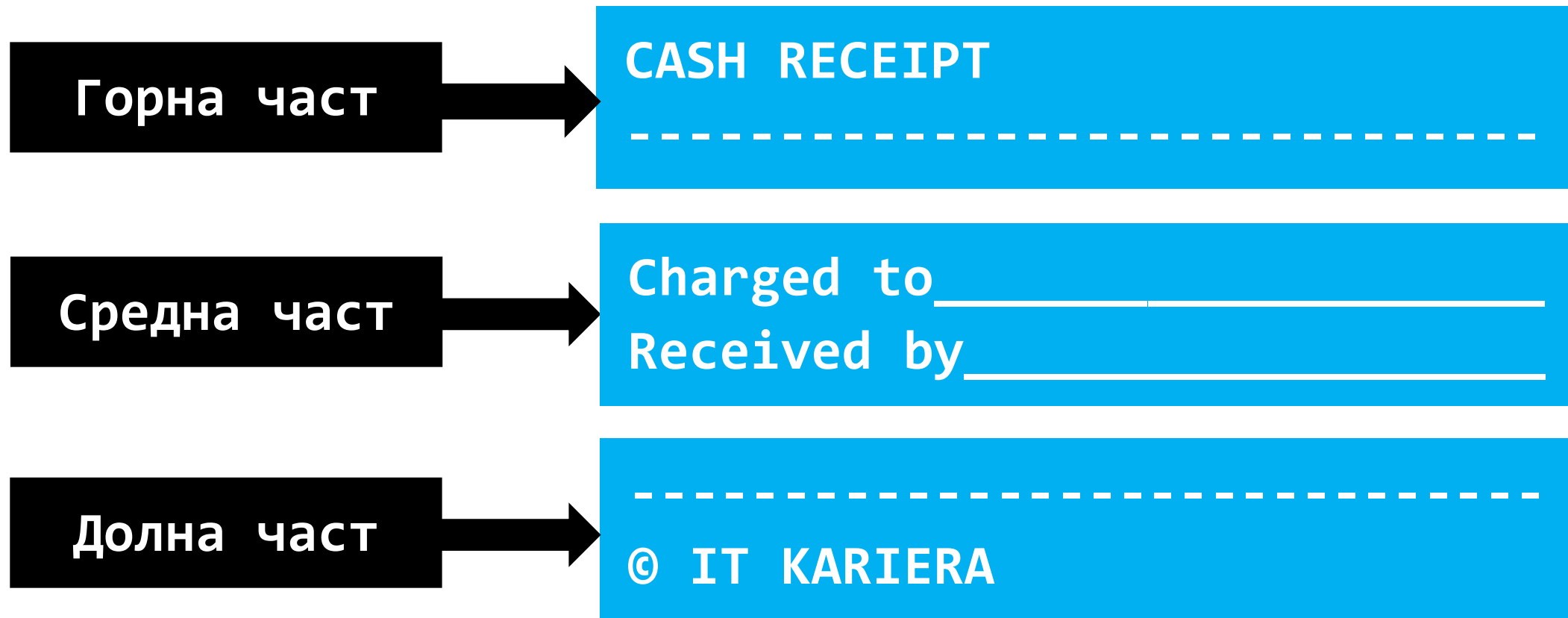
- Собственото си тяло – Рекурсия



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The window contains the text: 'Process is terminated due to StackOverflowException. Press any key to continue . . .'. The text is displayed in a monospaced font on a black background.

# Задача: Празна касова бележка

Да се напише метод, който печата празна касова бележка:



# Решение: Празна касова бележка

- Направете 3 метода които принтират всяка част
  - Копирайте съдържанието от предишния слайд
  - Използвайте знака "\u00A9" за символът ©
- Направете метод PrintReceipt(), който вика трите метода:

```
private static void PrintReceipt()
{
    PrintHeader();
    PrintBody();
    PrintFooter();
}
```



```
C:\Windows\system32\cmd.exe
CASH RECEIPT
-----
Charged to _____
Received by _____
-----
c IT KARIERA
Press any key to continue . . .
```

```
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('-');
}
```

# Memogu c napamempu

# Използване на параметри [1/2]

- Параметрите могат да бъдат всеки тип данни

```
static void PrintNumbers(int start, int end)
{
    for (int i = start; i <= end; i++)
    {
        Console.Write("{0} ", i);
    }
}
```

Деклариране на  
int start и int end

Няколко параметъра,  
разделени със запетайка

- Извикване на метод с конкретни стойности

```
static void Main()
{
    PrintNumbers(5, 10);
}
```

Подаване на конкретни  
стойности

# Използване на параметри [2/2]

- Можем да подаваме нула или няколко параметъра
- Параметрите могат да бъдат от различни типове
- Всеки параметър има име и тип

Няколко параметъра  
от различни типове

Тип на  
параметъра

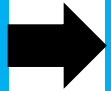
Име на  
параметъра

```
static void PrintStudent(string name, int age, double grade)
{
    Console.WriteLine("Student: {0}; Age: {1}, Grade: {2}",
        name, age, grade);
}
```

# Задача: Знак на цяло число

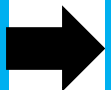
Да се създаде метод, който печата знака на цяло число  $n$ :

2



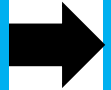
The number 2 is positive.

-5



The number -5 is negative.

0



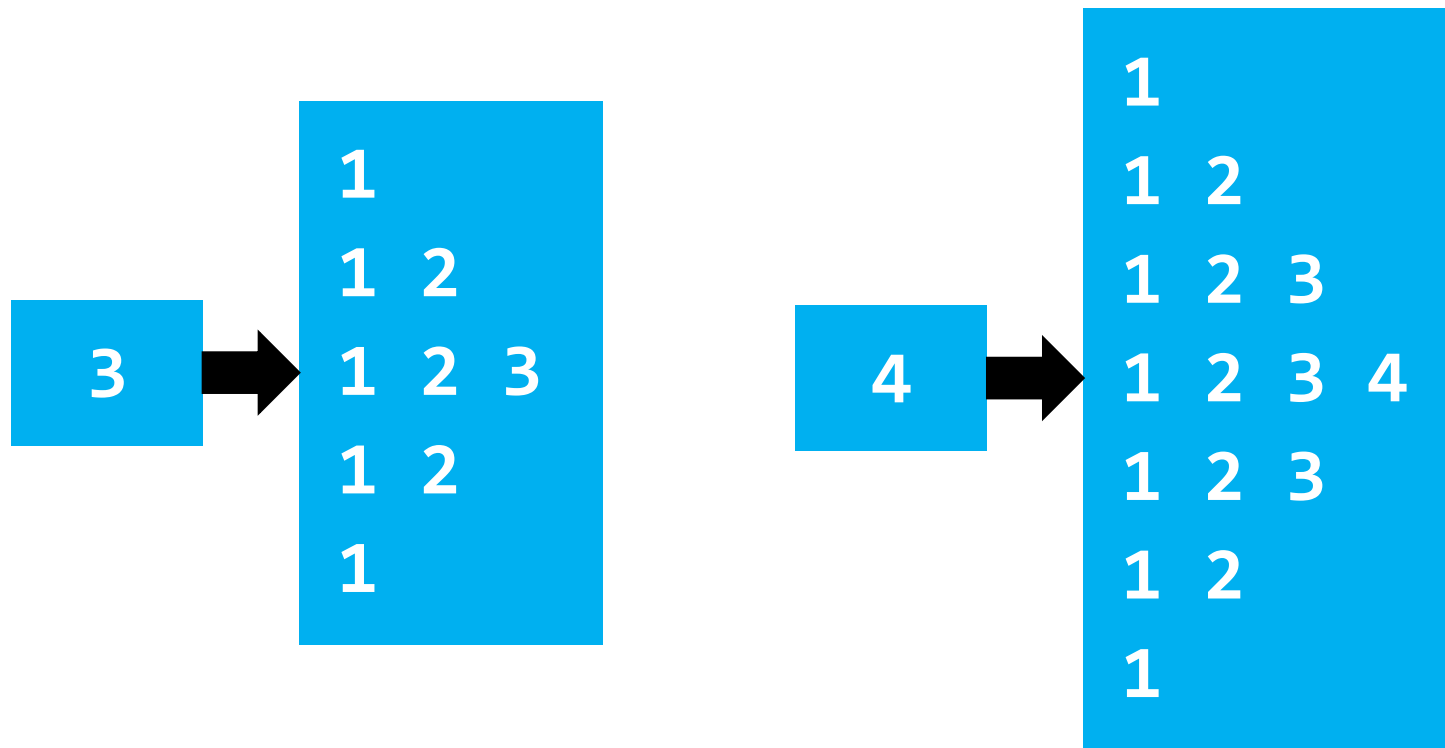
The number 0 is zero.

# Решение: Знак на цяло число

```
static void PrintSign(int number)
{
    if (number > 0)
        Console.WriteLine("The number {0} is positive", number);
    else if (number < 0)
        Console.WriteLine("The number {0} is negative.", number);
    else
        Console.WriteLine("The number {0} is zero.", number);
}
```

# Задача: Принтиране на триъгълник

Да се създаде метод, който принтира триъгълник, както е показано в примерите:



## Решение: Принтиране на триъгълник [1/2]

Създайте метод, който принтира един ред, състоящ се от числа в диапазон от определено начало до определен край:

```
static void PrintLine(int start, int end)
{
    for (int i = start; i <= end; i++)
    {
        Console.Write(i + " ");
    }
    Console.WriteLine();
}
```

## Решение: Принтиране на триъгълник [2/2]

Създайте метод, който принтира първата и после втората половина на триъгълника:

```
static void PrintTriangle(int n)
{
    for (int line = 1; line <= n; line++)
    {
        PrintLine(1, line);
    }

    for (int line = n - 1; line >= 1; line--)
    {
        PrintLine(1, line);
    }
}
```

Метод с параметър n

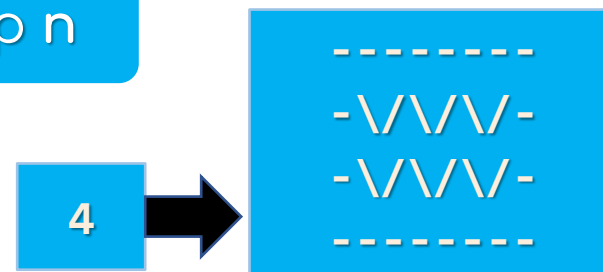
# Задача: Рисуване на запълнен квадрат

Нарисувайте на конзолата запълнен квадрат със страна **n**, както е показано в примера:

Метод с параметър n

```
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('-');
}
```



```
static void Main() {
    int n = // TODO: read n
    PrintHeaderRow(n);
    for (int i = 0; i < n - 2; i++)
        PrintMiddleRow(n);
    PrintHeaderRow(n);
}
```

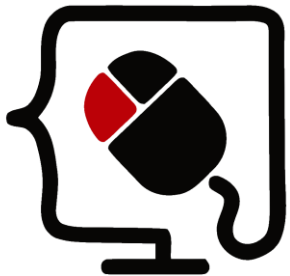
```
static int SumOfDigits(int num)
{
    int sum = 0;
    while (num > 0)
    {
        sum += num % 10;
        num = num / 10;
    }
    return sum;
}
```

# Деклариране и извикване на методи

Работа на живо в клас (лаб)

# Какво научихме днес?

- Можем да разделим голяма програма на прости методи, които решават по-малки проблеми
- Методите имат
  - име, тип, параметри и тяло
- Методите се извикват по тяхното име
- Могат да приемат параметри
  - Параметрите приемат реални стойности, когато методът се извика



Национална програма  
"Обучение за ИТ умения и кариера"  
<https://it-kariera.mon.bg>

Министерството на  
образованието и науката  
<https://www.mon.bg>



**SoftUni  
Foundation**



Документът е разработен за нуждите на Национална програма "Обучение за ИТ умения и кариера" на Министерството на образованието и науката (МОН), базиран е на учебно съдържание и методика, предоставени от фондация "Софтуерен университет" и се разпространява под свободен лиценз CC-BY-NC-SA (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).