

Упражнения: Методи

I. Деклариране и извикване на методи

1. Празна касова бележка

Създайте метод, който отпечатва празна касова бележка. Методът трябва да извиква три други метода: един за отпечатване на хедъра, един за основната част на бележката и един за футъра.

| | |
|---|---------------------------------------|
| Хедърът трябва да съдържа следния текст: | CASH RECEIPT ----- |
| Основната част на бележката съдържа текста: | Charged to _____ Received by _____ |
| Ето го и текста на футъра: | ----- © SoftUni |

Пример

| Изход |
|--|
| CASH RECEIPT ----- Charged to _____ Received by _____ ----- © SoftUni |

Подсказки

1. Първо създайте метод без параметри за отпечатването на хедъра. Дефиницията му започва със **static void**. Дайте му **смислено име**, например "PrintReceiptHeader" и напишете кода, който ще изпълнява този метод:

```
static void PrintReceiptHeader()
{
    Console.WriteLine("CASH RECEIPT");
    Console.WriteLine("-----");
}
```

2. Направете същото и за същинската част и футъра на касовата бележка.
3. Създайте **метод, който ще извиква тези три метода** в правилния ред. И на него дайте **смислено и описателно име**, примерно "PrintReceipt" и напишете неговия програмен код:

```
static void PrintReceipt()
{
    PrintReceiptHeader();
    PrintReceiptBody();
    PrintReceiptFooter();
}
```

4. За отпечатване на "©" използвайте Уникод-а "\u00A9"
5. **Извиквайте** методът PrintReceipt от main метода.

```
static void Main(string[] args)
{
    PrintReceipt();
}
```

2. Знак на цяло число

Създайте метод, отпечатващ знака на цяло число n.

Пример

| Вход | Изход |
|------|----------------------------|
| 2 | The number 2 is positive. |
| -5 | The number -5 is negative. |
| 0 | The number 0 is zero. |

Подсказки

1. Създайте метод с **описателно** име като "PrintSign". Методът трябва да получава **един параметър** от тип **int**.

```
static void PrintSign(int number)
{
}
```

2. Изградете и тялото на метода, като обработите трите случая:

- a. Ако числото е по-голямо от нула
- b. Ако числото е по-малко от нула
- c. И ако числото е равно на нула

3. Извикайте новосъздадения метод от метода main.

```
static void Main(string[] args)
{
    int n = int.Parse(Console.ReadLine());
    PrintSign(n);
}
```

3. Отпечатване на триъгълник

Създайте метод за отпечатване на триъгълници както е показано по-долу:

Примери

| Вход | Изход |
|------|---------------------------------------|
| 3 | 1 1 2 1 2 3 1 2 1 |
| 4 | 1 1 2 1 2 3 1 2 3 4 1 2 3 |

| | |
|--|----------|
| | 1 2 1 |
|--|----------|

Подсказки

1. След като прочетете входните данни
2. Започнете със създаването на метод за **отпечатване на един ред** от **дадено число start** до **дадено число end**. Изберете му **смислено име**, което описва неговото предназначение:

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

3. Помислете как може да го използвате, за да намерите решение на задачата
4. След като анализирате заданието, би трябвало да стигнете до извода, че ви трябват два цикъла
5. В първия цикъл може да отпечтатте горната половина от триъгълника без средния ред:

```
for (int i = 0; i < n; i++)
{
    PrintLine(1, i);
}
```

6. След това отпечатваме средния ред:

```
PrintLine(1, n);
```

7. Накрая отпечатваме и останалото от триъгълника:

```
for (int i = n - 1; i >= 0 ; i--)
{
    PrintLine(1, i);
}
```

4. Изчертаване на запълнен квадрат

Изчертайте на конзолата запълнен квадрат с дължина на страната n като в примера:

Пример

| Вход | Изход |
|------|--|
| 4 | <pre> ----- -\\//\\/- -\\//\\/- -\\//\\/- ----- </pre> |

Подсказки

1. Прочетете входните данни
2. Създайте метод, който ще печати най-горния и най-долния ред (тъй като те са еднакви). Не забравяйте да му дадете описателно име и като параметър да му подадете някаква дължина.

- а. Вместо цикъл може да използвате командата "new string", която създава нов текст, съставен от символ, повторен определен брой пъти:

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

3. Сега създайте метод, който ще отпечата средните редове. Е, ясно е, ще го наречете предполагаем "PrintMiddleRow" 😊

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

    Console.WriteLine('-');
}
```

4. Използвайте методите, които току-що създадохте, за изчертаването на квадрата:

```
static void Main()
{
    int n = int.Parse(Console.ReadLine());
    PrintHeaderRow(n);
    // TODO: Draw the rest of the square
}
```

II. Връщане на стойности и предефиниране

5. Конвертор за температури

Създайте метод, който конвертира температура от **Фаренхайт** в **Целзий**. Форматирайте резултата до втория десетичен знак.

Използвайте формулата: $(\text{fahrenheit} - 32) * 5 / 9$.

Примери

| Вход | Изход |
|------|--------|
| 95 | 35.00 |
| 33.8 | 1.00 |
| -40 | -40.00 |

Подсказки

1. Прочетете входните данни

2. Създайте метод, който **върща стойност от тип double**:

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

3. Извикайте метода в main и запишете върнатата стойност в нова променлива:

```
static void Main(string[] args)
{
    var fahrenheit = double.Parse(Console.ReadLine());
    var celsius = FahrenheitToCelsius(fahrenheit);
    Console.WriteLine("{0:F2}", celsius);
}
```

6. Пресмятане на лице на триъгълник

Създайте метод, който изчислява и **върща** лицето на триъгълник по дадени основа и височина:

Пример

| Вход | Изход |
|--------|-------|
| 3 4 | 6 |

Подсказки

- Първо прочетете входните данни
- После създайте метод, но този път **вместо** да пишете "**static void**" преди името му, напишете "**static double**", така че да го накараме да **върне стойност от тип double**:

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

3. Извикайте метода в main и **съхранете върната стойност в нова променлива**:

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

7. Повдигане на степен

Създайте метод, който пресмята и връща стойността на число, повдигнато на указаната степен:

Примери

| Вход | Изход |
|--------|-------|
| 2 8 | 256 |
| 3 4 | 81 |

Подсказки

1. Както обикновено, прочетете входните данни
2. Създайте метод, който ще има два параметъра - числото и степента, и ще връща резултат от тип double:

```
static double RaiseToPower(double number, int power)
{
    double result = 0d;

    // TODO: Calculate result (use a loop, or Math.Pow())

    return result;
}
```

3. Отпечатайте резултата

8. По-голямата от две стойности

Имате подадени като входни данни две стойности от един и същи тип. Стойностите може да са от тип int, char или string. Създайте метод GetMax() който връща по-голямата от двете стойности:

Примери

| Вход | Изход |
|-------------------------|-------|
| int 2 16 | 16 |
| char a z | z |
| string Ivan Todor | Todor |

Подсказки

1. За тази задача ще трябва да създадете три метода с едно и също име и с различни сигнатури
2. Създайте метод, който ще сравнява цели числа:

```
static int GetMax(int first, int second)
{
    if (first >= second)
    {
        // TODO: return value
    }

    // TODO: handle other cases
}
```

3. Създайте втори метод със същото име, който ще сравнява символи. Следвайте логиката на предния метод:

```
static char GetMax(char first, char second)
{
    // TODO: create logic
}
```

- И накрая създайте метод за сравняване на низове. Той ще е малко по-различен, тъй като низовете не може да бъдат сравнявани с операторите > и <

```
static string GetMax(string first, string second)
{
    if (first.CompareTo(second) >= 0)
    {
        // TODO: return value
    }
    // TODO: return value
}
```

Трябва да използвате метода "CompareTo()", който връща целочислена стойност (положителна ако сравняваният обект е по-голям, отрицателна, ако е по-малък и нула, ако двата са равни).

- Последната стъпка е да прочетете входните данни, да използвате променливи от подходящ тип и да извикате GetMax() от вашия Main():

```
var type = Console.ReadLine();
if (type == "int")
{
    int first = int.Parse(Console.ReadLine());
    int second = int.Parse(Console.ReadLine());
    int max = GetMax(first, second);
    Console.WriteLine(max);
}
else if (type == "char")
{
    // TODO: call GetMax with char arguments
}
else if (type == "string")
{
    // TODO: call GetMax with string arguments
}
```