



**Dėstytojas**

**Vilmantas Neviera**

# Metodai

**Data**



# Šiandien išmoksite

01

Metodai



# Kas yra metodai?

Metodai yra blokai kodo, kurie yra sukurti atlikti tam tikrą funkciją. Jie gali būti pernaudojami programos kode ir gali priimti parametrus bei grąžinti rezultatą. Metodai C# kalboje yra labai naudingi kodą organizuojant ir modularizuojant, taip padedant palengvinti programos kūrimą ir priežiūrą.



# Kas yra metodai?

Metodų sintaksė yra gana paprasta. Štai bendras metodų struktūros pavyzdys:

```
access_modifier return_type method_name(parameter_list)
{
    // code to be executed
}
```

**access\_modifier** nurodo metodų prieigos lygį, ir tai gali būti **public**, **private**, **protected**, **internal**, **protected internal** ar **private protected**. **public** reiškia, kad metodą galima pasiekti bet kurioje programos dalyje, o **private** reiškia, kad jis yra pasiekiamas tik iš tos pačios klasės.

**return\_type** nurodo grąžinamą rezultato tipą. Jei metodas nieko negrąžina, naudojamas raktažodis **void**.

**method\_name** yra unikalus metodų pavadinimas.

**parameter\_list** yra sąrašas kintamųjų, kuriuos metodas priima kaip parametrus. Kiekvienas parametras turi tipą ir vardą.



# Kas yra metodai?

Pabandykime sukurti paprastą metodą, kuris nieko negrąžina, tokie metodai skirti atlikti kažkokius veiksmus nepranešant apie veiksmų atlikimo rezultatą.

Paanalizuokime:

Metodas yra **public**, reiškia jis bus pasiekiamas iš "išorės", kas **C#** kalboje būtų pasiekiamas iš kitų klasių.

```
public static void PrintMenu()
{
    Console.WriteLine("1. Spausdinti prekes");
    Console.WriteLine("2. Žiūrėti krepšelį");
    Console.WriteLine("3. Išvalyti krepšelį");
    Console.WriteLine("4. Pirkti");
    Console.WriteLine("5. Išjungti programą");
}
```

Metodo grąžinimo tipas yra **void**, tai reiškia jog metodas atlikęs veiksmus negrąžins jokių kintamųjų.

Metodo pavadinimas yra **PrintMenu()**, pagal tai jį reikės kviesti.

Tarp () skliaustų nėra nurodytų jokių parametrų, tai reiškia jog kviečiant **PrintMenu()** metodą nereikės perduoti jokių kintamųjų



# Kas yra metodai?

Dabar pažiūrėkime į metodą, kuris grąžina rezultatą.

Šis metodas taip pat yra public, tik šį kartą vietoj void matome **int** tai reiškia, kad metodo pabaigoje jis privalo grąžinti sveiką skaičių, kaip matome, 27 ir 30 eilutėse.

```
19 public static int GetUserSelection()  
20 {  
21     Console.WriteLine("Įveskite pasirinkimą");  
22     string input = Console.ReadLine();  
23     if (!string.IsNullOrEmpty(input))  
24     {  
25         // geriau būtų naudoti TryParse, bet jį mokysimės vėliau  
26         int numberSelection = int.Parse(input);  
27         return numberSelection;  
28     }  
29     Console.WriteLine("Pasirinkimas neįvestas");  
30     return -1;  
31 }  
32 }
```



# Kas yra metodai?

Iškvietimas abiejų metodų būtų toks:

```
PrintMenu();  
int selection = GetUserSelection();
```

O programa atrodytų taip:

```
1. Spausdinti prekes  
2. Ziureti krepseli  
3. Isvalyti krepseli  
4. Pirkti  
5. Isjungti programa  
Iveskite pasirinkima  
5
```



# Kas yra metodai?

Dabar pažiūrėkime į metodą, kuris priima parametą. Praeitame metode, mes gavome vartotojo pasirinkimą, dabar galime šį pasirinkimą panaudot.

**Parametras** yra aprašomas tarp () po metodo pavadinimo, šiuo atveju parametras yra **int** duomenų tipo(parametrai gali būti visų duomenų tipų) ir jo pavadinimas **selection**, jis šio metodo kontekste veikia kaip bet koks kitas kintamasis, bet mes jį privalome perduoti visada kviečiant šį metodą.

Šis metodas nieko negrąžins, bet galės nukreipti į tolimesnį programos funkcionalumą.

```
public static void ProcessSelection(int selection)
{
    switch (selection)
    {
        case 1:
            Console.WriteLine("Pasirinkite spausdinti prekes");
            break;
        case 2:
            Console.WriteLine("Pasirinkote žiūrėti prekes");
            break;
        case 3:
            Console.WriteLine("Pasirinkote išvalyti krepšelį");
            break;
        case 4:
            Console.WriteLine("Pasirinkote pirkti");
            break;
        case 5:
            Console.WriteLine("Pasirinkote išjungti programą");
            break;
    }
}
```





# Kas yra metodai?

Iškvietimas abiejų metodų būtų toks:

```
private static void Main(string[] args)
{
    PrintMenu();
    int selection = GetUserSelection();
    ProcessSelection(selection);
}
```

O programa atrodytų taip:

```
1. Spausdinti prekes
2. Ziureti krepseli
3. Isvalyti krepseli
4. Pirkti
5. Isjungti programa
Iveskite pasirinkima
4
Pasirinkote pirkti
```



# Metodo kvietimas iš kito metodo

Čia matome kaip galima utilizuoti vieno metodo kvietimą iš kito.

Dažniausiai dėl vienos eilutės kodo mes metodo nekursime, bet pavyzdžio dėlei galime susikurt

```
public static bool IsOfLegalAge(DateTime birthDate)
{
    return CalculateAge(birthDate) >= 18;
}

1 reference
public static int CalculateAge(DateTime birthDate)
{
    return DateTime.Now.Year - birthDate.Year;
}
```



## Metodų pavyzdžiai

Metodas apskaičiuojantis atstumą tarp dviejų taškų dvimatėje erdvėje naudojant euklidinės erdvės formulę.

```
public static double CalculateDistance(double x1, double y1, double x2, double y2)
{
    return Math.Sqrt(Math.Pow((x2 - x1), 2) + Math.Pow((y2 - y1), 2));
}
```

Šis metodas apskaičiuoja trikampio plotą

```
public static double CalculateTriangleArea(double baseLength, double height)
{
    return 0.5 * baseLength * height;
}
```



## Metodų pavyzdžiai

Šis metodas apskaičiuoja ar metai yra keliamieji.

```
public static bool IsLeapYear(int year)
{
    return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0);
}
```

Šis metodas su truputį sudetingesne sintakse, patikrina ar telefono numeris yra tinkamo ilgio ir ar visi simboliai tekste yra skaitmenys

```
public static bool IsPhoneNumberValid(string phoneNumber)
{
    return phoneNumber.Length == 11 && phoneNumber.All(char.IsDigit);
}
```



## Užduotis nr. 1

- Sukurkite metodą pavadinimu **IsValidPassword**, kuris priima vieną **string** tipo parametą (**password**). Metoda turėtų grąžinti **true**, jei **password** yra ilgesnis nei 8 simboliai, ir **false**, jei yra trumpesnis.
- Sukurkite metodą pavadinimu **IsValidEmail**, kuris patikrina ar pateiktas **string** tipo parametras (**email**) yra tinkamas el. pašto adresas. Metoda turėtų grąžinti **true**, jei **email** yra tinkamas (turi "@" ir "."), arba **false**, jei yra netinkamas.
- Sukurkite metodą **ConvertToEuros**, kuris priima vieną **double** tipo parametą (**dollars**). Metoda turėtų konvertuoti dolerius į eurus pagal fiksuotą kursą (pavyzdžiui, 0.85) ir grąžinti eurų sumą.



## Užduotis nr. 2

- Sukurkite metodą **GetInitials**, kuris priima du **string** tipo parametrus (**firstName** ir **lastName**). Metodas turėtų gražinti vardą ir pavardę viename sakinyje.
- Sukurkite metodą **CalculateCylinderVolume**, kuris priima du **double** tipo parametrus (**radius** ir **height**). Metodas turėtų apskaičiuoti ir gražinti cilindro tūrį.
- Sukurkite metodą **IsNumberEven**, kuris priima vieną **int** tipo parametą (**number**). Metodas turėtų gražinti **true**, jei skaičius yra lyginis, ir **false**, jei nelyginis.
- Sukurkite metodą **CalculateRectangleArea**, kuris priima du **double** tipo parametrus (**length** ir **width**). Metodas turėtų apskaičiuoti ir gražinti stačiakampio plotą.



# Rekursija

Rekursija yra kompiuterių mokslo koncepcija, kai funkcija ar metodas save kviečia savo apibrėžime. Paprastai tariant, tai yra kai funkcija sprendžia užduotį naudodama tą pačią užduotį, bet mažesniu mastu, kol pasiekia atvejį, kai atsakymas yra akivaizdus ir gali būti tiesiogiai grąžinamas.



# Rekursija

Čia matome rekursijos pavyzdį, pradžioj tikriname ar paduotas skaičius nėra nulis, jeigu nulis grąžiname nulį, bet jeigu ne nulis, kviečiamas tas pats metodas tik su vienu skaičiu mažesniu parametru.



```
public static void Recursion(int number)
{
    if(number == 0)
    {
        return;
    }
    Console.WriteLine(number);
    Recursion(number - 1);
}
```





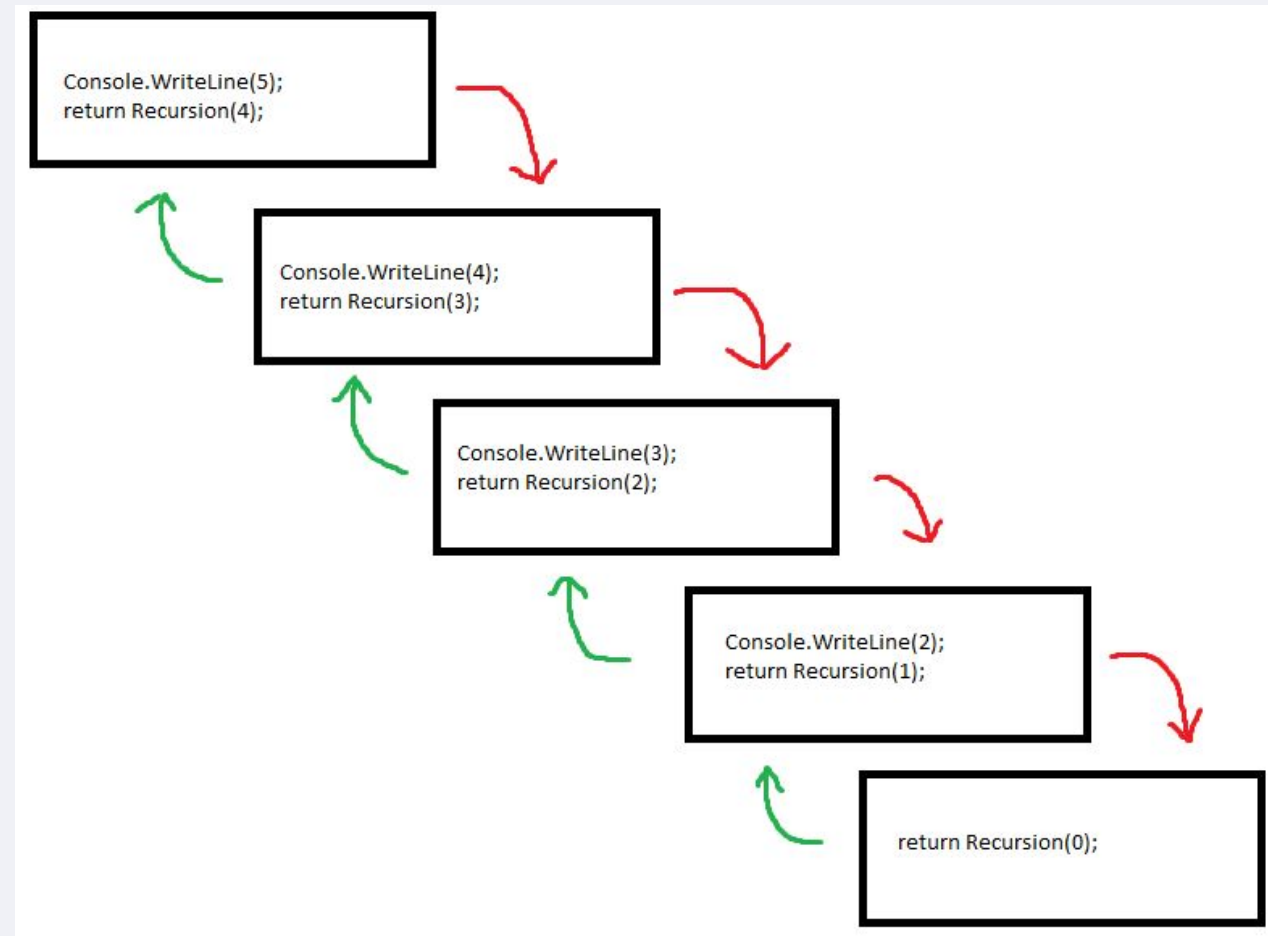
# Rekursija

Iškviečiame šį rekursijos metodą: `Recursion(5);`

Jo veikimas atrodys štai taip:

O išvestis bus tokia

```
5
4
3
2
1
```





### Užduotis nr. 3

- Naudodami rekursiją, sukurkite Faktorialo skaičiavimo programą.
- Naudodami rekursiją, sukurkite Fibonnaci skaičiavimo programą.