



**Code
Academy**



Dėstytojas

Vilmantas Neviera

Delegatai ir anoniminiai metodai

Data



Šiandien išmoksime

01

Kas yra delegatai?

02

Kas yra anoniminės funkcijos?



Delegatas

- Taip kaip sukurtas **klasės** objektas yra reference į duomenis atmintyje, taip ir delegatas yra reference tipo kintamasis į **metodą**.
- Delegato kūrimas yra susideda iš scope'o nurodymo (private/public etc.), nurodymo jog tai yra delegatas ("delegate"), nurodymo kokio duomenų tipo grąžinimo metodas tai bus(string, int etc.), nurodymo pavadinimo ir parametrų (string s).

```
private delegate int NumberChanger(int number);
```

- Toks delegatas galės reference'inti bet kokį metodą, kuris grąžina int ir parametruose turi vieną int kintamąjį.
- **Reference į kokį metodą delegatas rodo gali kisti programos veikimo metu(during runtime)**

Delegato sukūrimas



```
private delegate int NumberChanger(int n);
```

```
public static int AddNumber(int number)
{
    return GlobalNumber + number;
}

public static int SubtractNumber(int number)
{
    return GlobalNumber - number;
}
```

```
var numberChanger1 = new NumberChanger(AddNumber);
var numberChanger2 = new NumberChanger(SubtractNumber);
```

```
Console.WriteLine(numberChanger1(5));
Console.WriteLine(numberChanger2(6));
```

- Pirma deklaruojame delegatą
- Antra sukuriame funkcijas kurios tinka mūsų delegatui
- Trečia inicializuojame delegatą
- Ketvirta kviečiame metodus kuriuos referencina delegatai

Delegato sukūrimas



```
class Program
{
    static readonly int GlobalNumber = 10;
    private delegate int NumberChanger(int n);

    static void Main(string[] args)
    {
        var numberChanger1 = new NumberChanger(AddNumber);
        var numberChanger2 = new NumberChanger(SubtractNumber);

        Console.WriteLine(numberChanger1(5));
        Console.WriteLine(numberChanger2(6));
    }

    public static int AddNumber(int number)
    {
        return GlobalNumber + number;
    }

    public static int SubtractNumber(int number)
    {
        return GlobalNumber - number;
    }
}
```

- Paleidus šią programą bus sukurtas delegatas, metodai AddNumber, SubtractNumber.
- Tada inicializuoti delegatai numberChanger1 ir numberChanger2 ir jiems priskirti sukurti metodai.
- Inicializavus delegatus jie tampa kintamaisiais turinčiais rodyklę į metodą.
- Norint iškviesti tą metodą į kurį rodo delegatas tereikia nurodyti delegato objektą su atitinkamais parametrais.



Užduotis nr. 1

- Sukurkite delegatą, kuris rodys į metodą su `string` grąžinamuoju tipu ir trimis parametrais `firstName`, `lastName` ir `age`. Iškvieskite delegato metodą.
- Sukurkite delegatą, kuris rodys į metodą su `int` grąžinamuoju tipu ir dviem parametrais `number1` ir `number2`. Iškvieskite delegato metodą.
- Sukurkite delegatą, kuris rodys į metodą su `List<int>` grąžinamuoju tipu ir dviem parametrais `List<int>` ir `int step`, metodo esmė bus grąžinti kas kažkelintą elementą (kas kažkelintą nurodo parametras `"step"`). Iškvieskite delegato metodą.
- Sukurkite delegatą, kuris rodys į `GetType<T>` metodą su grąžinamuoju tipu `string` ir parametru `T element`, metodas grąžins element kintamojo duomenų tipą. Iškvieskite delegato metodą.



Anoniminiai metodai

- Anoniminiai metodai kaip ir implikuoja pats pavadinimas yra metodai be pavadinimo
- Anoniminiai metodai sukuriama naudojant delegate raktažodį ir juos priskirti galima delegate tipo kintamiesiems



Anoniminiai metodai

- Pirma sukuriame delegatą
- Antra inicializuojame sukurtą delegatą ir reikšmę jam priskiriame iškart aprašydami funkciją.

```
private delegate int NumberChanger(int n);

static void Main(string[] args)
{
    NumberChanger changer = delegate (int number)
    {
        return number + 5;
    };

    Console.WriteLine(changer(5));
}
```




Anoniminiai metodai

- Anoniminiai metodai taip pat mato išorinio scope kintamuosius

```
private delegate int NumberChanger(int n);

static void Main(string[] args)
{
    int x = 5;

    NumberChanger changer = delegate (int number)
    {
        return number + x;
    };

    Console.WriteLine(changer(5));
}
```



Anoniminiai metodai

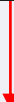
- Anoniminis metodas taip pat gali būti perduodamas kaip **parametras** į kitą metodą

```
public delegate void NumberChanger(int n);

static void Main(string[] args)
{
    int x = 5;

    ExecuteNumberChangerWithValue(x, delegate(int value)
    {
        Console.WriteLine(value);
    });
}

public static void ExecuteNumberChangerWithValue(int val, NumberChanger numberChanger)
{
    val += 10;
    numberChanger(val);
}
```



```
public delegate void NumberChanger(int n);

static void Main(string[] args)
{
    int x = 5;

    ExecuteNumberChangerWithValue(x, delegate(int value)
    {
        Console.WriteLine(value);
    });
}

public static void ExecuteNumberChangerWithValue(int val, NumberChanger numberChanger)
{
    val += 10;
    numberChanger(val);
}
```



Points to Remember :

- 1) Anonymous method can be defined using the delegate keyword
- 2) Anonymous method must be assigned to a delegate.
- 3) Anonymous method can access outer variables or functions.
- 4) Anonymous method can be passed as a parameter.
- 5) Anonymous method can be used as event handlers.



Užduotis nr. 2

- Perdarykite "Užduotis nr. 1" užduotis, kad delegatų inicializavimui naudotų anonimines funkcijas, o ne reguliarias.



Užduotis nr. 3

- Sukurkite klasę `Person` su `string name` ir `int age`
- Main metode sukurkite sąrašą žmonių skirtingais vardais ir amžiais
- Sukurkite delegatą `Filter`, kuris grąžins `bool` o per parametą pasiims `Person` objektą.
- Sukurkite tris metodus kurie grąžins `bool` reikšmes ir priiminės `Person` per parametą, vienas metodas tikrins ar žmogus yra vaikas `< 18` metų, kitas tikrins ar suaugęs `>= 18` metų ir trečias tikrins ar senjoras `>= 65` metai.
- Sukurkite metodą `DisplayPeople`, su parametrais `title`, `List<Person>` ir delegatu `Filter`. Metodo esmė bus eiti ciklu per asmenys ir paleisdinėti perduotą per parametrus filtrą patikrinti ar žmogus pvz. yra vaikas.

Metodo kvietimas atrodys maždaug taip: `DisplayPeople("Children:", people, IsChild);`



Delegates ir Anoniminiai metodai

https://www.tutorialspoint.com/csharp/csharp_delegates.htm

<https://www.tutorialsteacher.com/csharp/csharp-anonymous-method>

**Naudinga
informacija**