

Programozás C# nyelven

3. előadás



<http://e-learning.ujs.sk/>

Az Objektum Orientált Programozás (OOP) alapelvei:

- **Egységbezárás (encapsulation)**
Az adatokat és a hozzájuk tartozó függvényeket egyetlen egységben (osztályban) kezeljük.
- **Öröklés (inheritance)**
Az osztály továbbfejlesztése. Ennek során a származtatott osztály öröklí az ősoosztálytól az összes adatmezőt (adatot) és metódust (függvényt).
- **Sokalakúság (polymorphism)**
Ugyanarra a metódusra különböző objektumok különbözőképpen reagálnak.

Osztályok és objektumok



sz1



sz2



sz3

Objektum Orientált Programozás (OOP) alapfogalmak:

- Az **OSZTÁLY**-t úgy képzelhetünk el, mint bizonyos objektumok sablonját. Pl.: **Szemely**
- Az **OBJEKTUM** az osztály egy konkrét előfordulása.
Pl.: **sz1, sz2, sz3**
- Az **adatmező** az osztály egy változója.
Pl.: **_nev, _szuletesiEv**
- A **metódus** az osztály egy függvénye, mely az adott osztály adatmezőivel végez valamilyen műveletet.
Pl.: **SetNev(), GetNev()**
- Osztályból objektumokat **példányosítással** hozhatunk létre.
Pl.: **sz1 = new Szemely();**

```
class Szemely
{
    private string _nev;
    public Szemely()
    {
        _nev = "Ismeretlen";
    }
    public string GetNev()
    {
        return _nev;
    }
    public void SetNev(string nev)
    {
        _nev = nev;
    }
}
```

adatmező

konstruktor

metódusok

```
Szemely sz1;


private void button1_Click(object se
{
    sz1 = new Szemely();
}

private void button2_Click(object se
{
    if (sz1 != null)
    {
        sz1.SetNev("Peti");
    }
}

private void button3_Click(object se
{
    if (sz1 != null)
    {
        label1.Text = sz1.GetNev();
    }
}
```

GetNev, SetNev metódusok helyett Nev tulajdonság (property) használata:

```
class Szemely
{
    private string _nev;

    public string Nev  tulajdonság
    {
        get { return _nev; }
        set { _nev = value; }
    }

    public Szemely()
    {
        _nev = "Ismeretlen";
    }
}
```

~~sz1.SetNev("Peti");~~

sz1.Nev = "Peti";

~~label1.Text = sz1.GetNev();~~


label1.Text = sz1.Nev;

Automatikusan implementált Nev tulajdonság (auto-implemented property) használata:

```
class Szemely
{
    public string Nev { get; set; }

    public Szemely()
    {
        Nev = "Ismeretlen";
    }
}
```

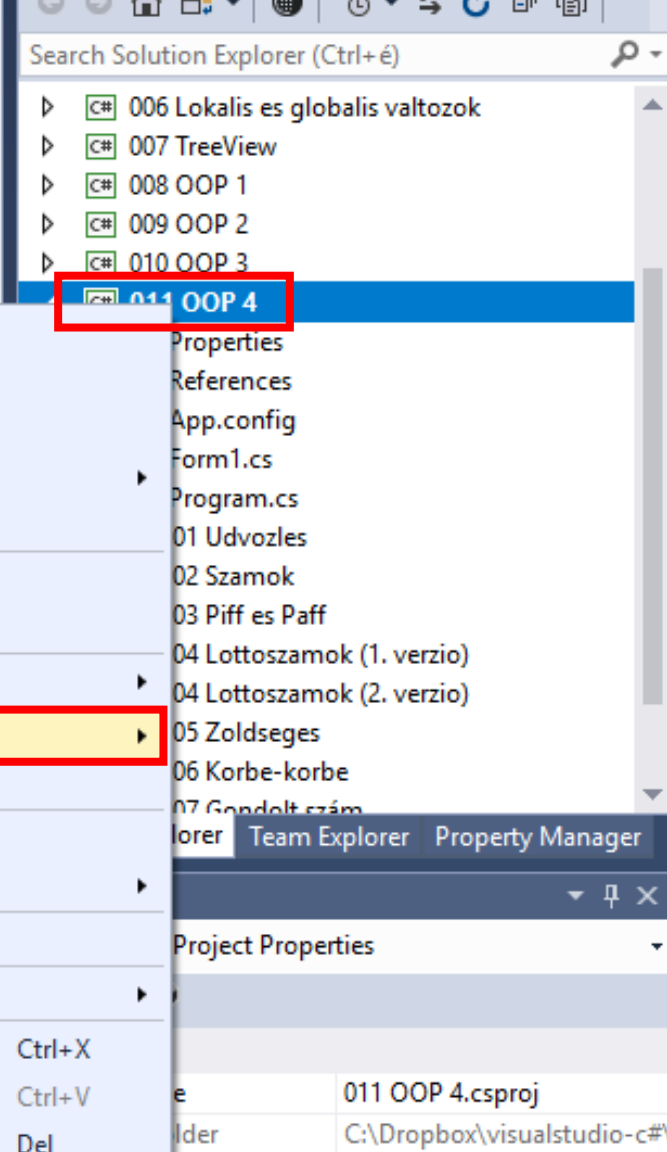
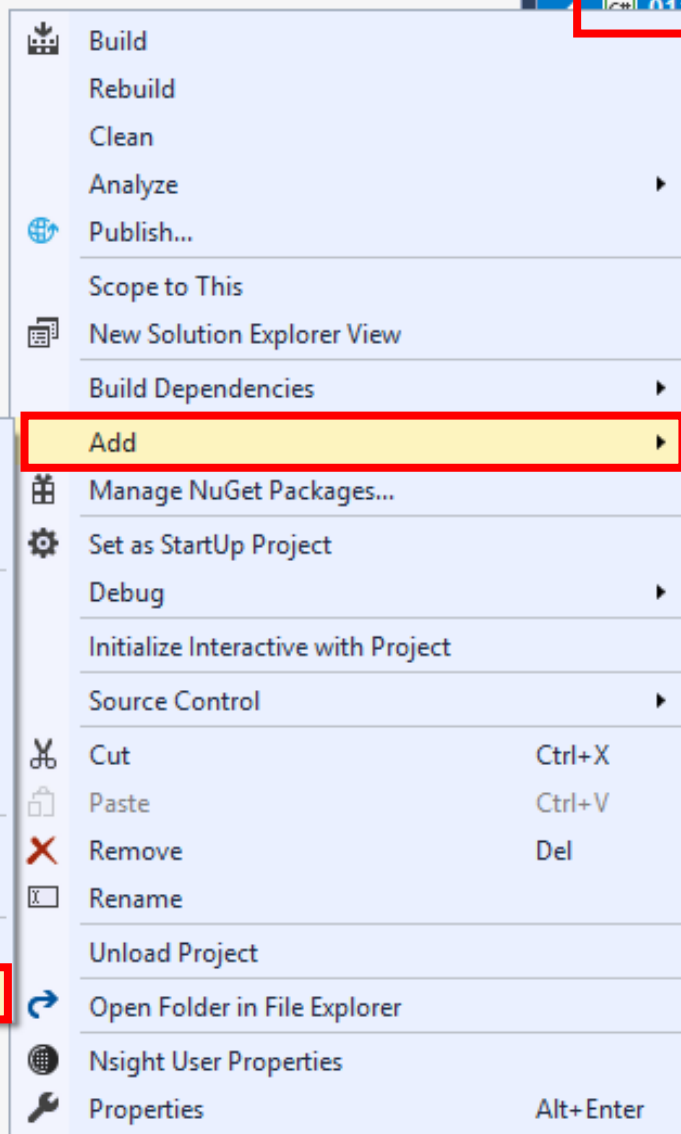
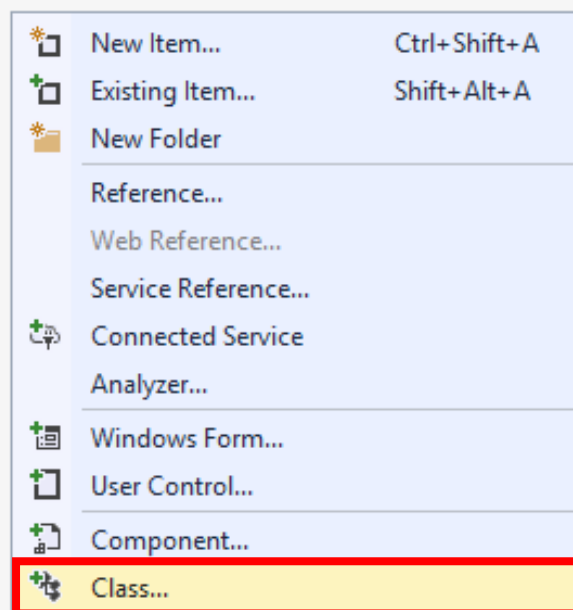
automatikusan
implementált
tulajdonság



```
sz1.Nev = "Peti";
```

```
label1.Text = sz1.Nev;
```

Osztály hozzáadása a projekthez (külön állományba):



Személy osztály (Szemely.cs állomány):

011 OOP 4

```
using System;
```

```
namespace _011_OOP_4
```

```
{
```

```
    public class Szemely
```

```
    {
```

```
        public string Nev { get; }
```

```
        public int SzuletesiEv { get; }
```

```
        public Szemely(string aNev, int aSzuletesiEv)
```

```
        {
```

```
            Nev = aNev;
```

```
            SzuletesiEv = aSzuletesiEv;
```

```
        }
```

```
        public int Eletkor()
```

```
        {
```

```
            return DateTime.Now.Year - SzuletesiEv;
```

```
        }
```

```
    }
```

```
}
```

csak olvasható (read-only)
automatikusan implementált
tulajdonságok

konstruktor

metódus

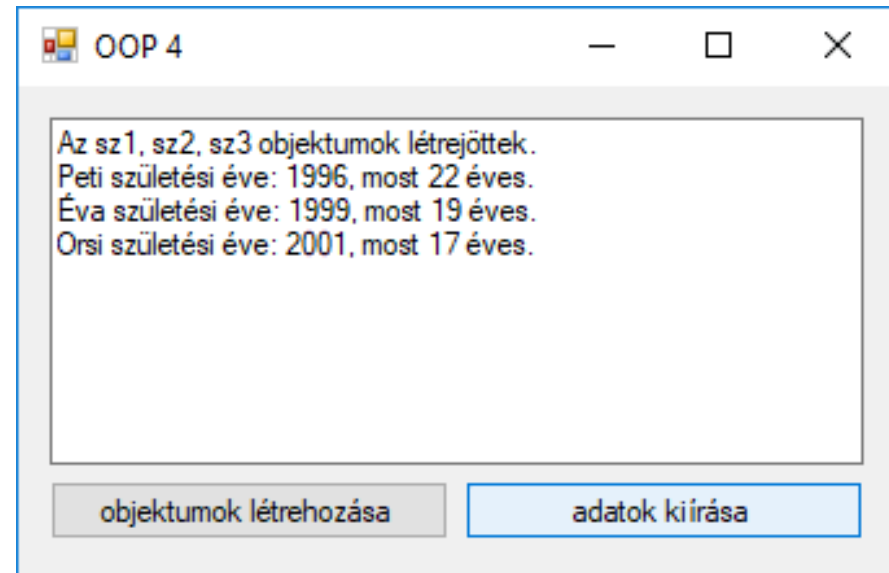
A Személy osztály használata (Form1.cs állományban):

```
using System;
using System.Windows.Forms;

namespace _011_OOP_4
{
    public partial class Form1 : Form
    {
        Szemely sz1, sz2, sz3;

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            sz1 = new Szemely("Peti", 1996);
            sz2 = new Szemely("Éva", 1999);
            sz3 = new Szemely("Orsi", 2001);
            textBox1.AppendText("Az sz1, sz2, sz3 objektumok létrejöttek.\n");
        }
    }
}
```



```
private void button2_Click(object sender, EventArgs e)
{
    textBox1.AppendText(sz1.Nev + " születési éve: " + sz1.SzuletesiEv
        + ", most " + sz1.Eletkor() + " éves.\n");
    textBox1.AppendText(sz2.Nev + " születési éve: " + sz2.SzuletesiEv
        + ", most " + sz2.Eletkor() + " éves.\n");
    textBox1.AppendText(sz3.Nev + " születési éve: " + sz3.SzuletesiEv
        + ", most " + sz3.Eletkor() + " éves.\n");
}
```

Összefoglalás:

- OOP alapelvei: egységbezárás, öröklés, sokalakúság
- OOP: osztály, objektum, adatmező, metódus, példányosítás
- Adatmezők használata („_nev”, „SetNev”, „GetNev”)
- Tulajdonságok („_nev”, „Nev”)
- Automatikusan implementált tulajdonságok („Nev”)
- „Szemely” osztály létrehozása külön állományban