

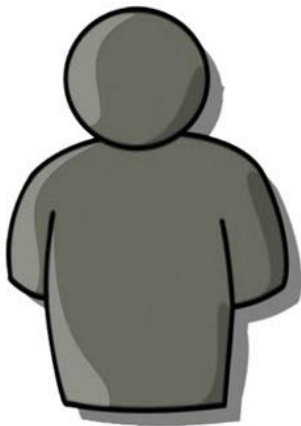
Programozás C# nyelven

4. előadás



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

Nev
SzuletesiEv
Eletkor()

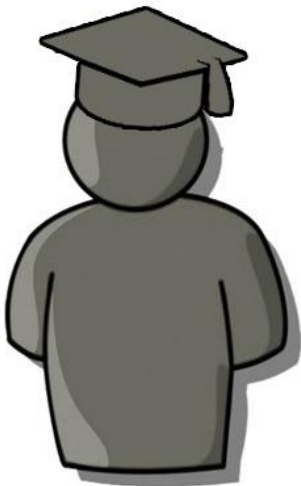


Szemely



ÖRÖKLÉS (INHERITANCE)

Nev
SzuletesiEv
Eletkor()
_jegyek
_jegyekSzama
OsszesJegy()
UjJegy()
Átlag()



Diak



Peti
1996

Éva
1999

Orsi
2001

sz1

sz2

sz3



Feri
2003
1,2,2,1,2

d1



Timi
2002
1,1,3,1

d2

Szemely.cs

```
using System;
```

```
namespace _012_OOP_Orokles
```

```
{
```

```
    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;
```

```
        }
```

```
    }
```

```
}
```

tulajdonságok

konstruktor

metódus

Diak.cs

```
using System;
```

```
namespace _012_OOP_Orokles
```

```
{
```

```
public class Diak : Szemely
```

```
{
```

```
private int[] _jegyek = new int[50];
```

```
private int _jegyekSzama = 0;
```

```
public Diak(string aNev, int aSzuletesiEv, int[] aJegyek) : base(aNev, aSzuletesiEv)
```

```
{
```

```
for (int i = 0; i < aJegyek.Length; i++)
```

```
{
```

```
    _jegyek[i] = aJegyek[i];
```

```
}
```

```
_jegyekSzama = aJegyek.Length;
```

```
}
```

```
public int[] OsszesJegy()
```

```
{
```

```
int[] a = new int[_jegyekSzama];
```

```
for (int i=0; i<_jegyekSzama; i++)
```

```
{
```

```
    a[i] = _jegyek[i];
```

```
}
```

```
return a;
```

```
}
```

privát adatmezők

konstruktor

metódus

```
public void UjJegy(int aJegy)
```

metódus

```
{  
    _jegyek[_jegyekSzama] = aJegy;  
    _jegyekSzama++;  
}
```

metódus

```
public double Atlag()
```

```
{  
    double atlag = 0;  
    for(int i=0; i<_jegyekSzama; i++)  
    {  
        atlag = atlag + _jegyek[i];  
    }  
    atlag = Math.Round(atlag / _jegyekSzama, 2);  
    return atlag;  
}
```

```
}
```

```
}
```

Form1.cs

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

```
namespace _012_OOP_Orokles
```

```
{  
    public partial class Form1  
    {
```

```
        Diak d1, d2;
```

```
        public Form1()  
        {  
            InitializeComponent();  
        }
```

```
        private void button1_Click(object sender, EventArgs e)  
        {
```

```
            int[] d1jegyei = { 1, 2, 2, 1, 2 };  
            d1 = new Diak("Feri", 2003, d1jegyei);  
            int[] d2jegyei = { 1, 1, 3, 1 };  
            d2 = new Diak("Timi", 2002, d2jegyei);  
            textBox1.AppendText("d1, d2 objektumok létrehozva\n");  
        }
```

Diákok

d1, d2 objektumok létrehozása

d1, d2 alapadatainak kiírása

d1, d2 jegyei

1 d1 új jegyet kap

1 d2 új jegyet kap

d1, d2 átlagai

d1, d2 objektumok létrehozása

```
private void button2_Click(object sender, EventArgs e)
```

d1, d2 alapadatainak kiírása

```
{
    textBox1.AppendText(d1.Nev + " születési éve: " + d1.SzuletesiEv
        + ", életkora: " + d1.Eletkor() + "\n");
    textBox1.AppendText(d2.Nev + " születési éve: " + d2.SzuletesiEv
        + ", életkora: " + d2.Eletkor() + "\n");
}
```

```
private void button3_Click(object sender, EventArgs e)
```

d1, d2 jegyei

```
{
    textBox1.AppendText(d1.Nev + " jegyei: ");
    foreach(int j in d1.OsszesJegy())
    {
        textBox1.AppendText(j + ", ");
    }
    textBox1.AppendText("\n");
    textBox1.AppendText(d2.Nev + " jegyei: ");
    foreach (int j in d2.OsszesJegy())
    {
        textBox1.AppendText(j + ", ");
    }
    textBox1.AppendText("\n");
}
```

```
private void button4_Click(object sender, EventArgs e)
```

d1 új jegyet kap

```
{
    int j = Convert.ToInt32(textBox2.Text);
    d1.UjJegy(j);
}
```

```
private void button5_Click(object sender, EventArgs e)
{
    int j = Convert.ToInt32(textBox3.Text);
    d2.UjJegy(j);
}
```

d2 új jegyet kap

```
private void button6_Click(object sender, EventArgs e)
{
    textBox1.AppendText(d1.Nev + " jegyeinek atlaga: " + d1.Atlag() + "\n");
    textBox1.AppendText(d2.Nev + " jegyeinek atlaga: " + d2.Atlag() + "\n");
}
```

d1, d2 átlagai

Láthatósági módosítók:

- **public** – a public láthatósági módosítóval ellátott adatmezők és metódusok bárhol elérhetők.
- **private** – a private láthatósági módosítóval ellátott adatmezők és metódusok kizárólag az adott osztályban érhetők el.
- **protected** – a protected láthatósági módosítóval ellátott adatmezők és metódusok az adott osztályban és az abból leszármaztatott osztályokban érhetők el.

A osztály

```
public int x  
private int y  
protected int z
```

```
public KiirasA()
```



B osztály : A osztály

```
public KiirasB()
```

Láthatósági módosítók

az A osztály adatmezői: x - public, y - private, z - protected
a B osztály az A osztályból volt származtatva

Form1 osztály : Form osztály

```
private A a = new A()  
private B b = new B()
```

```
private KiirasForm1()
```

...

```
a.KiirasA()  
b.KiirasB()  
KiirasForm1()
```

A.cs

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

```
namespace _013_OOP_Lathatosagi_modositok
```

```
{
```

```
    public class A
```

```
    {
```

```
        public int x = 1;
```

```
        private int y = 2;
```

```
        protected int z = 3;
```

```
        public void KiirasA(TextBox tbox)
```

```
        {
```

```
            tbox.AppendText("x = " + x + ", ");
```

```
            tbox.AppendText("y = " + y + ", ");
```

```
            tbox.AppendText("z = " + z + "\n");
```

```
        }
```

```
    }
```

```
}
```

B.cs

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

```
namespace _013_OOP_Lathatosagi_modositok
{
    public class B : A
    {
        public void KiirasB(TextBox tbox)
        {
            tbox.AppendText("x = " + x + ", ");
            tbox.AppendText("y nem látható, ");
            tbox.AppendText("z = " + z + "\n");
        }
    }
}
```

Form1.cs

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

namespace _013_OOP_Lathatosagi_modositok
{
    public partial class Form1 : Form
    {
        private A a = new A();
        private B b = new B();

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            a.KiirasA(textBox1);
        }
    }
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    b.KiirasB(textBox1);
}
```

```
private void KiirasForm1(TextBox tbox)
{
    tbox.AppendText("a.x = " + a.x + ", ");
    tbox.AppendText("a.y nem látható, ");
    tbox.AppendText("a.z nem látható\n");
    tbox.AppendText("b.x = " + b.x + ", ");
    tbox.AppendText("b.y nem látható, ");
    tbox.AppendText("b.z nem látható\n");
}
```

```
private void button3_Click(object sender, EventArgs e)
{
    KiirasForm1(textBox1);
}
```

```
}
```

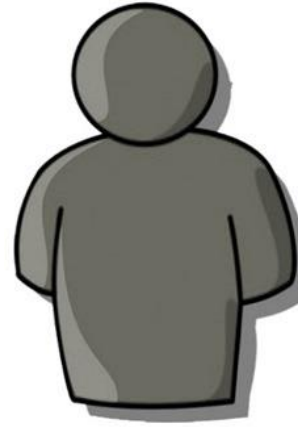
```
}
```

Feladat:

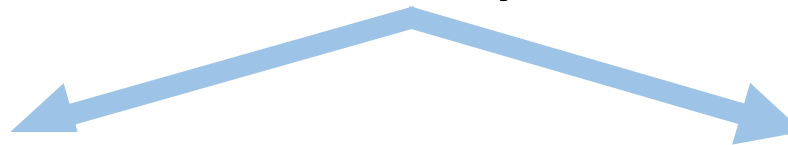
Hozzunk létre egy **Munkas** osztályt, amelynek őse a **Szemely** osztály legyen.

- A Munkas osztálynak legyen egy **_fizetes** privát adatmezője, melynek kezdeti értékét a konstruktor egyik paraméterében lehessen megadni.
- Készítsünk egy **GetFizetes()** publikus metódust, amely visszatérési értéke a munkás fizetése.
- A munkás fizetését az alábbi két publikus metódus segítségével lehessen módosítani, melyek növelik ill. csökkentik a fizetést a paraméterükben megadott százalékkal:
 - **FizetesEmeles(int szazalek)**
 - **FizetesCsokkentes(int szazalek)**
- A létrehozott Munkas osztály segítségével hozzunk létre két objektumot: **m1**, **m2**, melyek használatát próbáljuk ki.

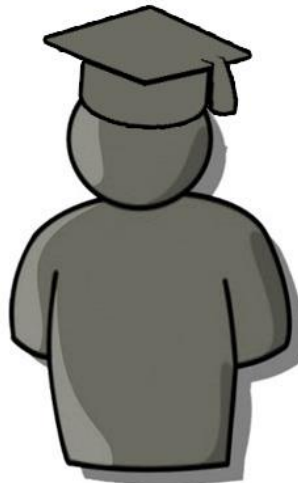
Nev
SzuletesiEv
Eletkor()



Szemely

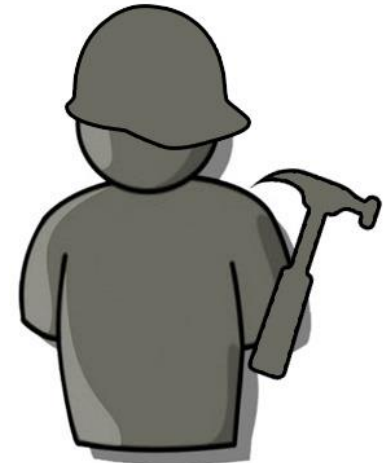


Nev
SzuletesiEv
Eletkor()
_jegyek
_jegyekSzama
OsszesJegy()
UjJegy()
Átlag()



Diak

Nev
SzuletesiEv
Eletkor()
_fizetes
GetFizetes()
FizetesEmeles()
FizetesCsokkentes()



Munkas

Nev
SzuletesiEv
Eletkor()
_fizetes
GetFizetes()
FizetesEmeles()
FizetesCsokkentes()



Munkas

Zoli
1995
590,- EUR



m1

Tomi
1997
560,- EUR



m2

Összefoglalás:

- Öröklés (Szemely -> Diak)
- Láthatósági módosítók: public, private, protected

Feladat:

- Öröklés (Szemely -> Munkas)