Programozás C# nyelven

6. előadás

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

osztály
(class)

objektum
(object)

adatmező
(field)

metódus
(method)

példányosítás
(instantiation)

tulajdonság
(property)

konstruktor
(constructor)

egységbezárás
(encapsulation)

automatikusan
implementált
tulajdonság
(auto-implemented
property)

öröklés
(inheritance)

sokalakúság
(polymorphism)

private public protected

statikus típus
(static type)

dinamikus típus
(dynamic type)

statikus kötés,
fordítási idejű kötés,
korai kötés
(early binding)

dinamikus kötés,
futási idejű kötés,
késői kötés
(late binding)

new virtual override

osztályok hierarchiája
(class hierarchy)

„Object” osztály

GetClass() ToString()

partial class

metódus túlterhelés
(method overloading)

konstruktor túlterhelés
(constructor overloading)

Upcast: Ősre konvertálás.

Downcast: Leszármazottra konvertálás.

021 OOP Upcast, downcast

```
Diak d = new Diak("Feri", 2003, new int[] { 1, 2, 2, 1, 2 });
textBox1.AppendText("----- d objektum: -----\n");
textBox1.AppendText("Nev: " + d.Nev + "\n");
textBox1.AppendText("Eletkor: " + d.Eletkor() + "\n");
textBox1.AppendText("Jegyek atlaga: " + d.Atlag() + "\n");

// --- UPCAST --- ősre konvertálás
Szemely sz = d;
textBox1.AppendText("----- sz objektum: -----\n");
textBox1.AppendText("Nev: " + sz.Nev + "\n");
textBox1.AppendText("Eletkor: " + sz.Eletkor() + "\n");
// textBox1.AppendText("Jegyek atlaga: " + sz.Atlag() + "\n");

// --- DOWNCAST --- gyermekre konvertalas
Diak d2 = (Diak)sz;
textBox1.AppendText("----- d2 objektum: -----\n");
textBox1.AppendText("Nev: " + d2.Nev + "\n");
textBox1.AppendText("Eletkor: " + d2.Eletkor() + "\n");
textBox1.AppendText("Jegyek atlaga: " + d2.Atlag() + "\n");
```

„this” kulcsszó: Hivatkozás az osztályból létrehozott aktuális példányra.

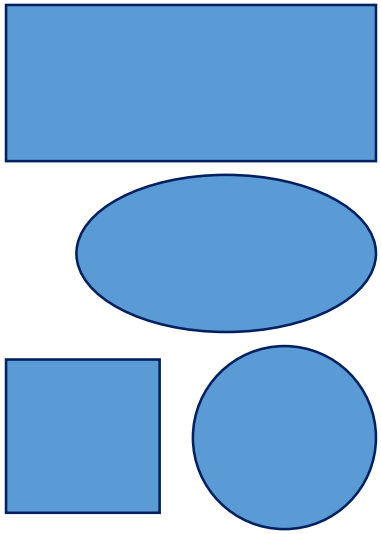
022 OOP This kucsszo

```
public class Alakzat
{
    private int szelesseg, magassag;

    public Alakzat(int szelesseg, int magassag)
    {
        this.szelesseg = szelesseg;
        this.magassag = magassag;
    }

    public string Info()
    {
        string s = "Alakzat objektum - ";
        s += "szelessege: " + szelesseg + ", ";
        s += "magassaga: " + magassag + ", ";
        return s;
    }

}
```



Alakzat al = new Alakzat(15, 7);

Alakzat

private int **szelesseg**, **magassag**

public **Alakzat(int szelesseg, int magassag)**

public string **Info()**

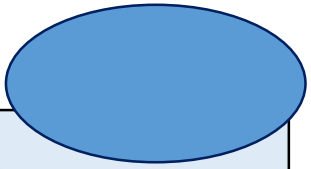
public **abstract** double **Terulet()**

Hova tegyük a ***public double Terulet()*** *metódust, ha az* ***Info()*** *segítségével az alakzat területét is vissza akarjuk adni?*

Teglalap : Alakzat

public **Teglalap(int szelesseg, int magassag)** :
base(szelesseg, magassag)

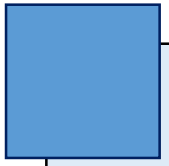
public **override** double Terulet()



Ellipszis : Alakzat

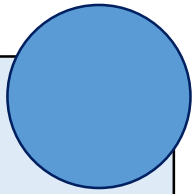
public **Ellipszis(int szelesseg, int magassag)** :
base(szelesseg, magassag)

public **override** double **Terulet()**



Negyzet : Teglalap

public **Negyzet(int szelesseg)** :
base(szelesseg, szelesseg)



Kor : Ellipszis

public **Kor(int szelesseg)** :
base(szelesseg, szelesseg)

Absztrakt metódus: olyan metódus, melynek törzsét nem írjuk meg.
Absztrakt osztály: absztrakt metódust tartalmazó osztály. Ilyen osztályból nem hozhatunk létre objektumot, csupán arra szolgál, hogy abból további osztályokat származtassunk.

023 OOP Absztrakt osztaly es metodus

```
public abstract class Alakzat
{
    protected int szelesseg, magassag;

    public Alakzat(int szelesseg, int magassag)
    {
        this.szelesseg = szelesseg;
        this.magassag = magassag;
    }

    public string Info()
    {
        string s = GetType().Name + " objektum - ";
        s += "szelessege: " + szelesseg + ", ";
        s += "magassaga: " + magassag + ", ";
        s += "terulete: " + Math.Round(Terulet(), 2);
        return s;
    }

    public abstract double Terulet();

}
```

```
public class Teglalap : Alakzat
{
    public Teglalap(int szelesseg, int magassag) : base(szelesseg, magassag)
    {
    }

    public override double Terulet()
    {
        return szelesseg * magassag;
    }
}
```

```
public class Ellipszis : Alakzat
{
    public Ellipszis(int szelesseg, int magassag) : base(szelesseg, magassag)
    {
    }

    public override double Terulet()
    {
        return (szelesseg / 2.0) * (magassag / 2.0) * Math.PI;
    }
}
```

```
public class Negyzet : Teglalap
{
    public Negyzet(int szelesseg) : base(szelesseg, szelesseg)
    {
    }
}
```

```
public class Kor : Ellipszis
{
    public Kor(int szelesseg) : base(szelesseg, szelesseg)
    {
    }
}
```

„sealed” módosító: az ilyen módosítóval ellátott osztályt nem lehet továbbszármaztatni. Metódus esetén: „sealed“ módosítóval ellátott metódus nem írható felül a származtatott osztályban.

024 OOP Sealed modosito

```
public sealed class Negyzet : Teglalap
{
    public Negyzet(int szelesseg) : base(szelesseg, szelesseg)
    {
    }
}
```

```
public sealed class Kor : Ellipszis
{
    public Kor(int szelesseg) : base(szelesseg, szelesseg)
    {
    }
}
```

Statikus metódus: olyan metódus, amely nem a példányhoz, hanem az osztályhoz kapcsolódik. Ilyen metódust nem a példány, hanem az osztály nevével hívunk meg.

Statikus osztály: olyan osztály, amely csak statikus metódusokat tartalmaz.

025 OOP Statikus metodus, statikus osztaly

```
public static class Muveletek
{
    public static int Osszeadas(int a, int b)
    {
        return a + b;
    }

    public static int Kivonas(int a, int b)
    {
        return a - b;
    }

    public static int Szorzas(int a, int b)
    {
        return a * b;
    }
```

```
    public static double Osztas(int a, int b)
    {
        return (double)a / b;
    }

}
```

```
...

private void button1_Click(object sender, EventArgs e)
{
    textBox1.Clear();
    textBox1.AppendText("8 + 7 = " + Muveletek.Osszeadas(8,7) + "\n");
    textBox1.AppendText("8 - 7 = " + Muveletek.Kivonas(8, 7) + "\n");
    textBox1.AppendText("8 * 7 = " + Muveletek.Szorzas(8, 7) + "\n");
    textBox1.AppendText("8 / 7 = " + Muveletek.Osztas(8, 7) + "\n");
}

...
```

Az alábbiak közül melyek statikus metódusok?

Convert.ToInt32()

Convert.ToDouble()

Random.Next()

Random.NextDouble()

Int32.ToString()

Int32.Parse()

Double.ToString()

Double.GetType()

Math.Round();

Math.Sqrt();

Az alábbiak közül melyek statikus osztályok?

Random

Convert

Int32

Double

Math

Statikus adatmező: **olyan adatmező, amely nem a példányhoz, hanem az osztályhoz kapcsolódik.**

026 OOP Statikus adatmezo

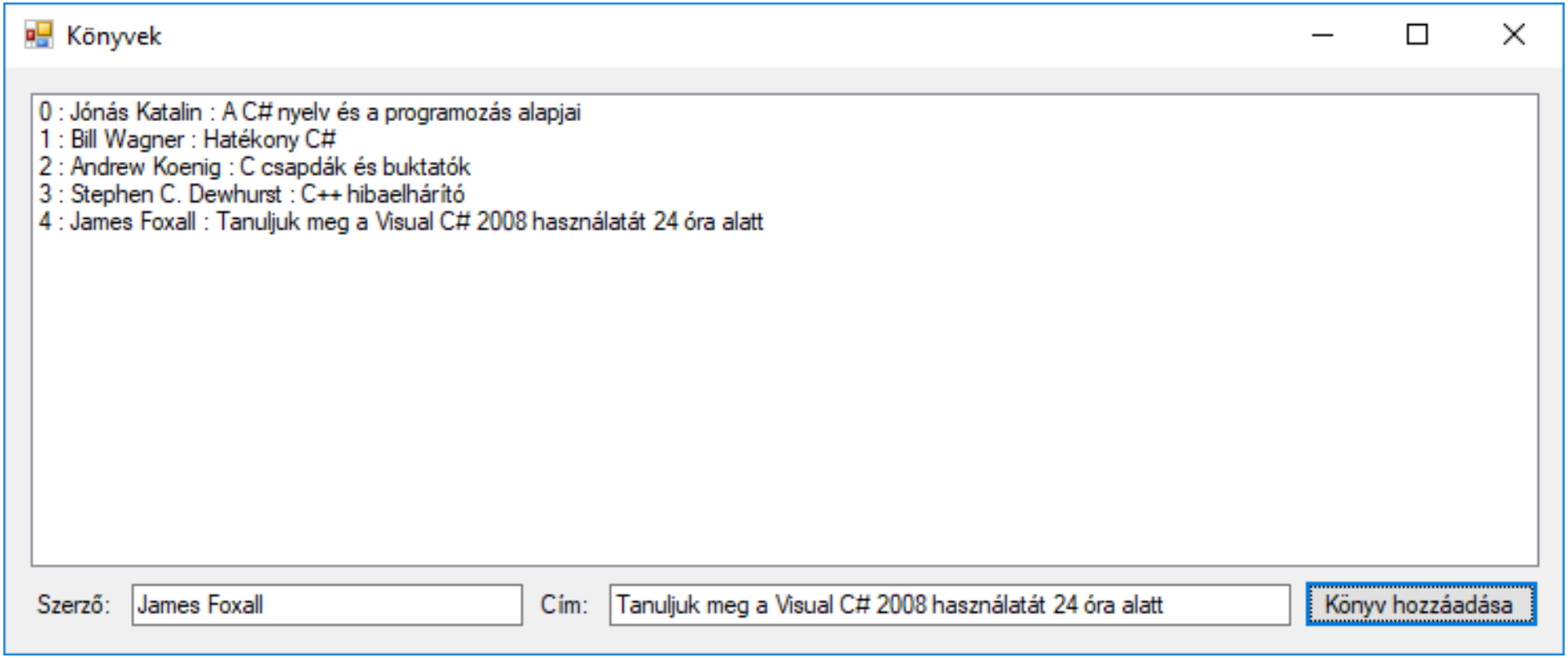
```
public class Konyv
{
    private string szerzo;
    private string cim;
    private int id;

    private static int szamlalo = 0;

    public Konyv(string szerzo, string cim)
    {
        this.szerzo = szerzo;
        this.cim = cim;
        id = szamlalo;
        szamlalo++;
    }

    public override string ToString()
    {
        return id + " : " + szerzo + " : " + cim;
    }

}
```




```
private void button1_Click(object sender, EventArgs e)
{
    Konyv k = new Konyv(textBox1.Text, textBox2.Text);
    listBox1.Items.Add(k);
}
```

Összefoglalás:

- Upcast, downcast
- “this” kulcsszó
- Absztrakt metódus, absztrakt osztály (“abstract”)
- “sealed” módosító
- Statikus metódus, statikus osztály, statikus adatmező (“static”)