Programozás C# nyelven

11. előadás

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

Generikus típusok (Generics): Segítségükkel nem kell adatokat konvertálnunk, egyszerűbb, gyorsabb a munkánk. Fő tulajdonságuk, hogy típusparamétereket fogadnak, melyek arra szolgálnak, hogy meghatározzák a műveleteknél használt típusokat.

Pl. „Verem” megvalósítása generikus osztályként:

041 Generikus osztaly

```
public class Verem<T> : IEnumerable
{
    private T[] elemek;
    private int n = 0;

    public Verem(int meret)
    {
        elemek = new T[meret];
    }
```

```
    public void ElemBe(T s)
    {
        if (n < elemek.Length)
        {
            elemek[n] = s;
            n++;
        }
    }

    public T ElemKi()
    {
        if (n > 0)
        {
            n--;
            return elemek[n];
        }
        return default(T);
    }

    public IEnumerator GetEnumerator()
    {
        T[] tomb = new T[n];
        Array.Copy(elemek, 0, tomb, 0, n);
        return new VeremEnumerator<T>(tomb);
    }
}
```

```
public class VeremEnumerator<T> : IEnumerator
{
    private T[] tomb;
    private int index;

    public object Current
    {
        get
        {
            return tomb[index];
        }
    }

    public VeremEnumerator(T[] tomb)
    {
        this.tomb = tomb;
        index = tomb.Length;
    }

    public bool MoveNext()
    {
        index--;
        return index >= 0;
    }
```

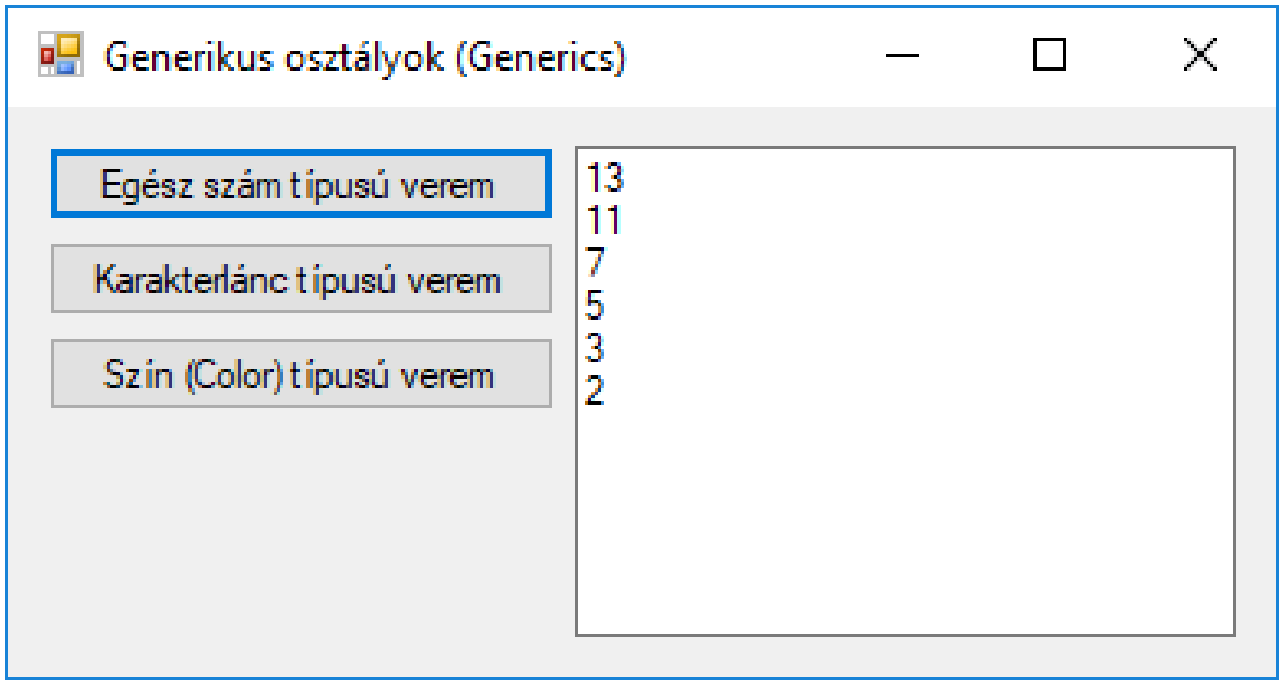
```
    public void Reset()
    {
        index = tomb.Length;
    }
}
```

T

int

String

Color

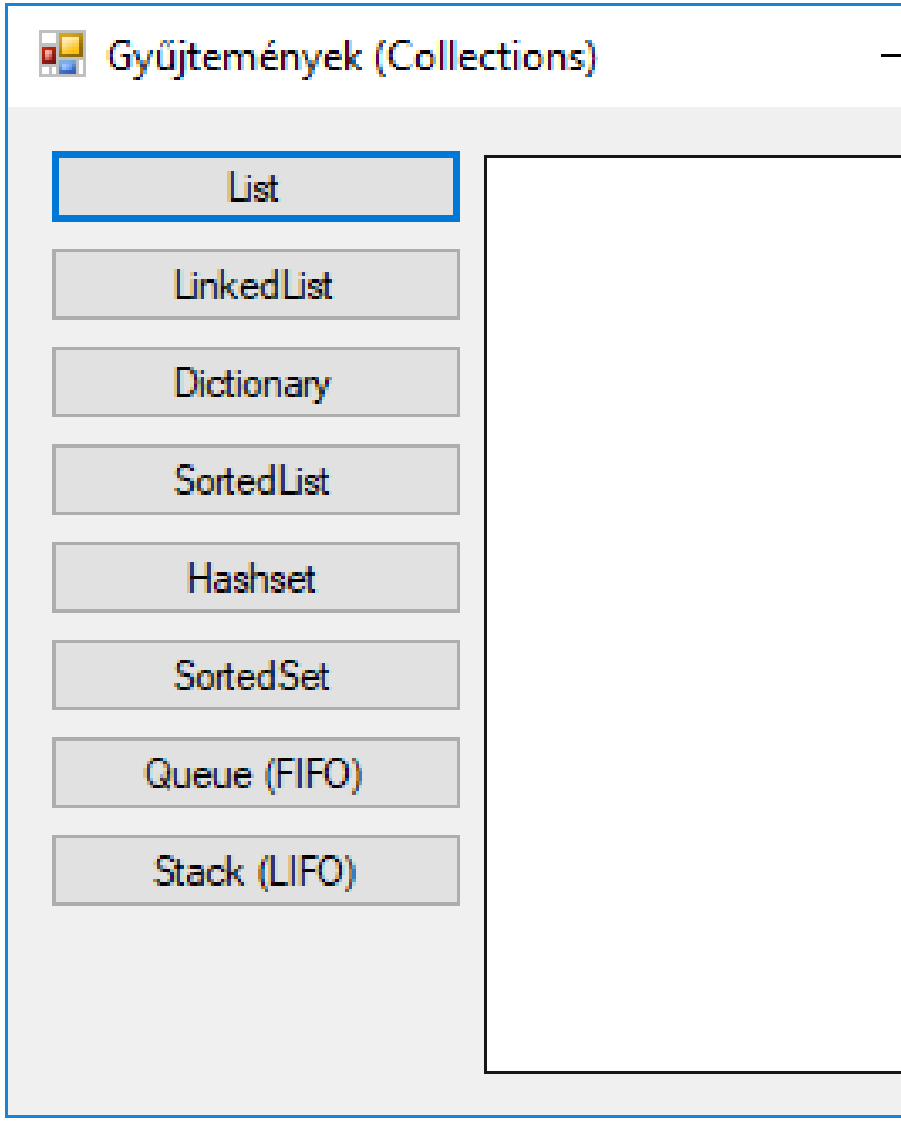


```
private void button1_Click(object sender, EventArgs e)
{
    Verem<int> v1 = new Verem<int>(10);
    v1.ElemBe(2);
    v1.ElemBe(3);
    v1.ElemBe(5);
    v1.ElemBe(7);
    v1.ElemBe(11);
    v1.ElemBe(13);
    textBox1.Clear();
    foreach (int elem in v1)
    {
        textBox1.AppendText(elem + "\r\n");
    }
}
private void button2_Click(object sender, EventArgs e)
{
    Verem<String> v2 = new Verem<String>(10);
    v2.ElemBe("Alma");
    v2.ElemBe("Barack");
    v2.ElemBe("Citrom");
    v2.ElemBe("Dió");
    v2.ElemBe("Egres");
    v2.ElemBe("Földimogyoró");
    v2.ElemBe("Gesztenye");
    textBox1.Clear();
    foreach (String elem in v2)
    {
        textBox1.AppendText(elem + "\r\n");
    }
}
```

```
private void button3_Click(object sender, EventArgs e)
{
    Verem<Color> v3 = new Verem<Color>(10);
    v3.ElemBe(Color.Red);
    v3.ElemBe(Color.Green);
    v3.ElemBe(Color.Blue);
    v3.ElemBe(Color.Yellow);
    v3.ElemBe(Color.Orange);
    v3.ElemBe(Color.Purple);
    v3.ElemBe(Color.Black);
    textBox1.Clear();
    foreach (Color elem in v3)
    {
        textBox1.AppendText(elem + "\r\n");
    }
}
```

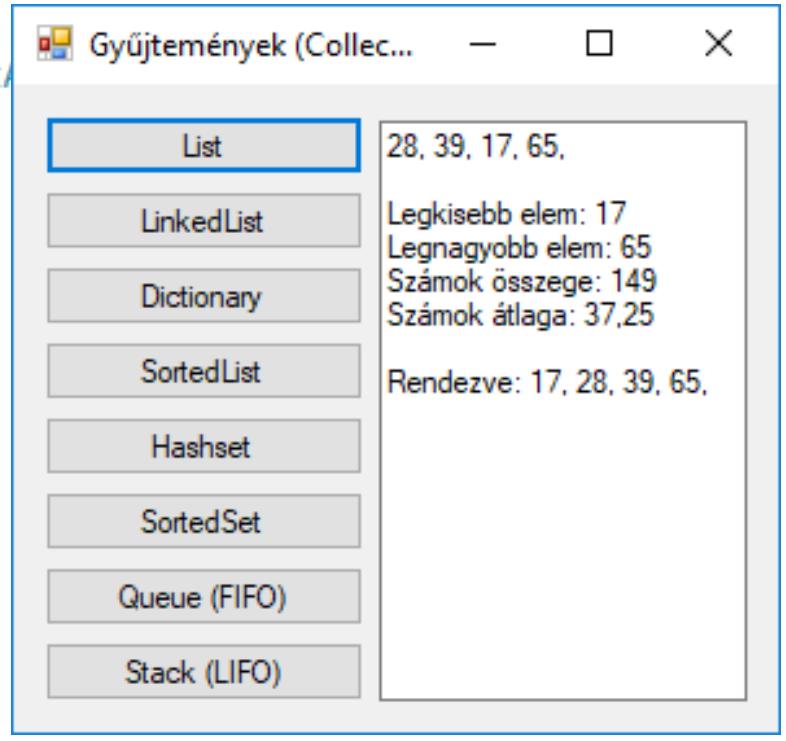
Gyűjtemények (Collections): a tömbökhöz hasonlóan adatok, objektumokat tárolására használhatók, több hasznos metódussal is rendelkeznek.

Pl.: listák, szótárak, halmazok.

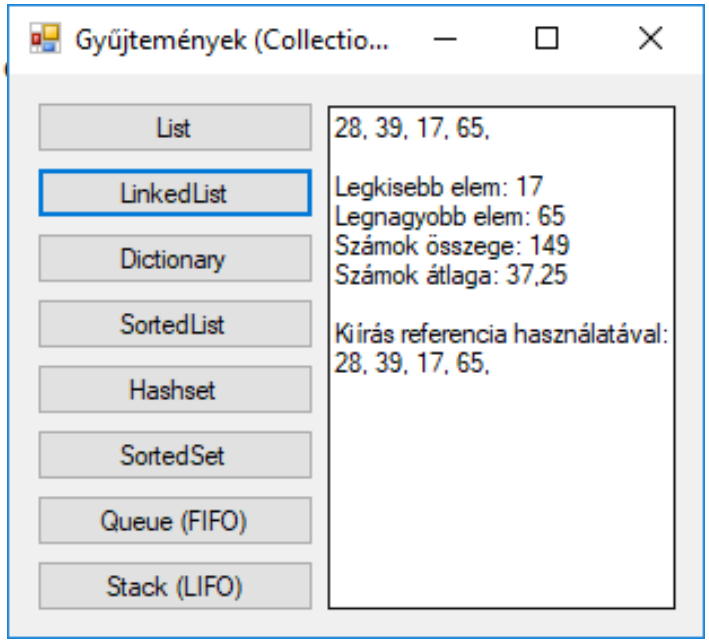


042 Gyujtemenyek

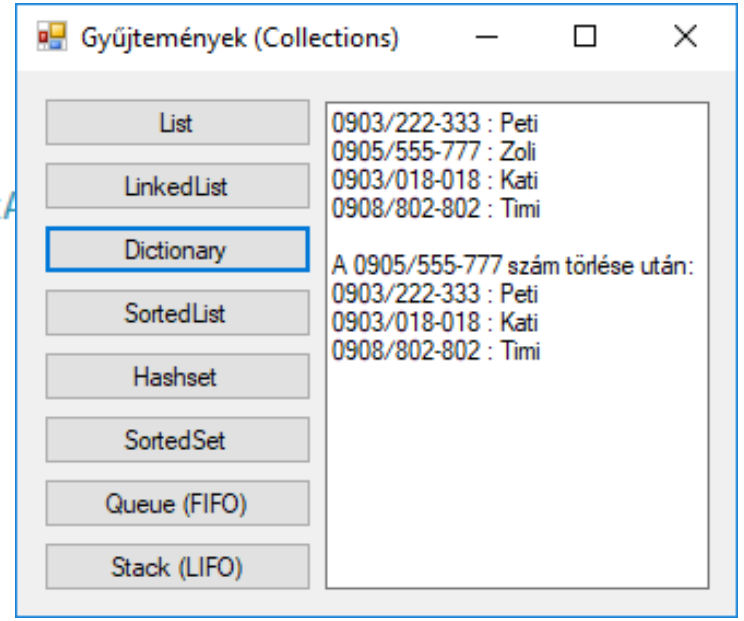
```
private void button1_Click(object sender, EventA
{
    List<int> szamok = new List<int>();
    szamok.Add(28);
    szamok.Add(39);
    szamok.Add(17);
    szamok.Add(65);
    textBox1.Clear();
    for (int i = 0; i < szamok.Count; i++)
    {
        textBox1.AppendText(szamok[i] + ", ");
    }
    textBox1.AppendText("\r\n\r\n");
    textBox1.AppendText("Legkisebb elem: " + szamok.Min() + "\r\n");
    textBox1.AppendText("Legnagyobb elem: " + szamok.Max() + "\r\n");
    textBox1.AppendText("Számok összege: " + szamok.Sum() + "\r\n");
    textBox1.AppendText("Számok átlaga: " + szamok.Average() + "\r\n\r\n");
    szamok.Sort();
    textBox1.AppendText("Rendezve: ");
    foreach (int szam in szamok)
    {
        textBox1.AppendText(szam + ", ");
    }
}
```



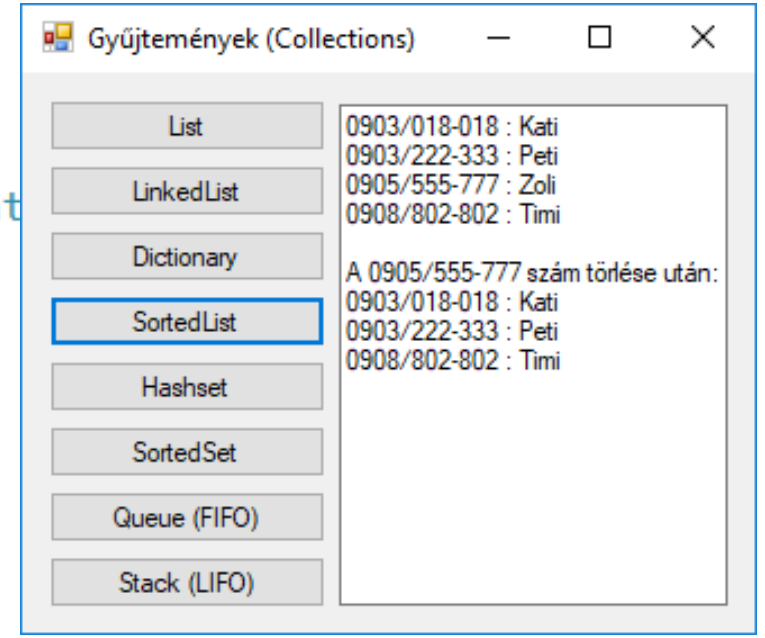
```
private void button2_Click(object sender, EventArgs
{
    LinkedList<int> szamok = new LinkedList<int>();
    szamok.AddLast(28);
    szamok.AddLast(39);
    szamok.AddLast(17);
    szamok.AddLast(65);
    textBox1.Clear();
    foreach (int szam in szamok)
    {
        textBox1.AppendText(szam + ", ");
    }
    textBox1.AppendText("\r\n\r\n");
    textBox1.AppendText("Legkisebb elem: " + szamok.Min() + "\r\n");
    textBox1.AppendText("Legnagyobb elem: " + szamok.Max() + "\r\n");
    textBox1.AppendText("Számok összege: " + szamok.Sum() + "\r\n");
    textBox1.AppendText("Számok átlaga: " + szamok.Average() + "\r\n\r\n");
    textBox1.AppendText("Kiírás referencia használatával:\r\n");
    LinkedListNode<int> akt = szamok.First;
    while (akt != null)
    {
        textBox1.AppendText(akt.Value + ", ");
        akt = akt.Next;
    }
}
```



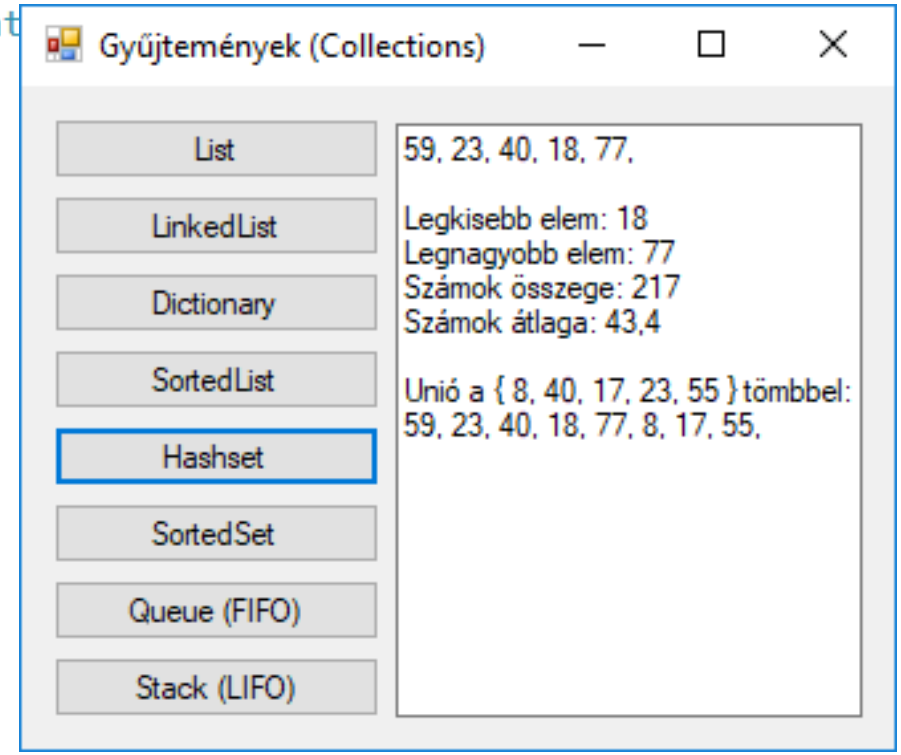
```
private void button3_Click(object sender, EventA
{
    Dictionary<string, string> telefonkonyv =
        new Dictionary<string, string>();
    telefonkonyv.Add("0903/222-333", "Peti");
    telefonkonyv.Add("0905/555-777", "Zoli");
    telefonkonyv.Add("0903/018-018", "Kati");
    telefonkonyv.Add("0908/802-802", "Timi");
    textBox1.Clear();
    foreach (string tszam in telefonkonyv.Keys)
    {
        textBox1.AppendText(tszam + " : " + telefonkonyv[tszam] + "\r\n");
    }
    textBox1.AppendText("\r\n");
    telefonkonyv.Remove("0905/555-777");
    textBox1.AppendText("A 0905/555-777 szám törlése után:\r\n");
    foreach (string tszam in telefonkonyv.Keys)
    {
        textBox1.AppendText(tszam + " : " + telefonkonyv[tszam] + "\r\n");
    }
}
```



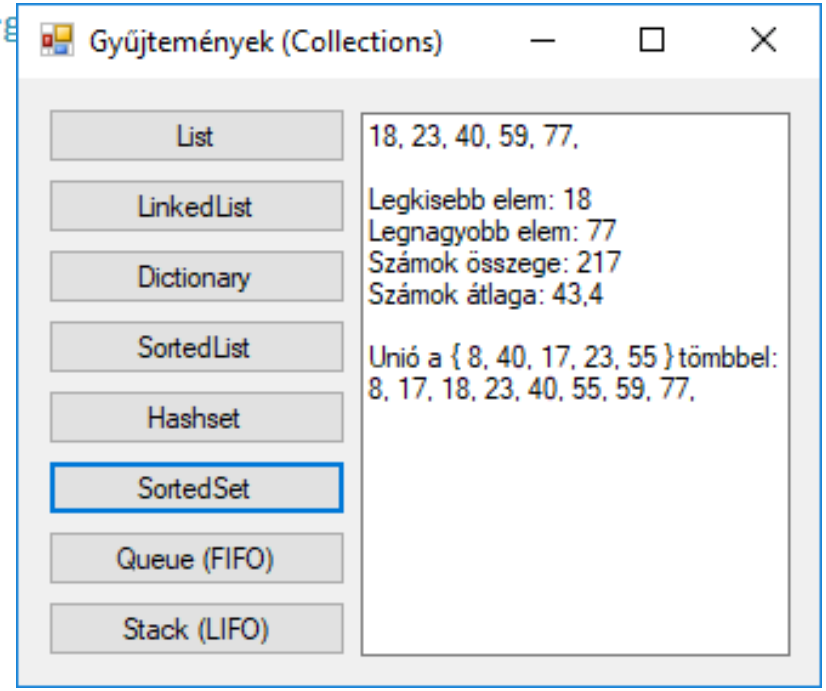
```
private void button4_Click(object sender, Event
{
    SortedList<string, string> telefonkonyv =
        new SortedList<string, string>();
    telefonkonyv.Add("0903/222-333", "Peti");
    telefonkonyv.Add("0905/555-777", "Zoli");
    telefonkonyv.Add("0903/018-018", "Kati");
    telefonkonyv.Add("0908/802-802", "Timi");
    textBox1.Clear();
    foreach (string tszam in telefonkonyv.Keys)
    {
        textBox1.AppendText(tszam + " : " + telefonkonyv[tszam] + "\r\n");
    }
    textBox1.AppendText("\r\n");
    telefonkonyv.Remove("0905/555-777");
    textBox1.AppendText("A 0905/555-777 szám törlése után:\r\n");
    foreach (string tszam in telefonkonyv.Keys)
    {
        textBox1.AppendText(tszam + " : " + telefonkonyv[tszam] + "\r\n");
    }
}
```



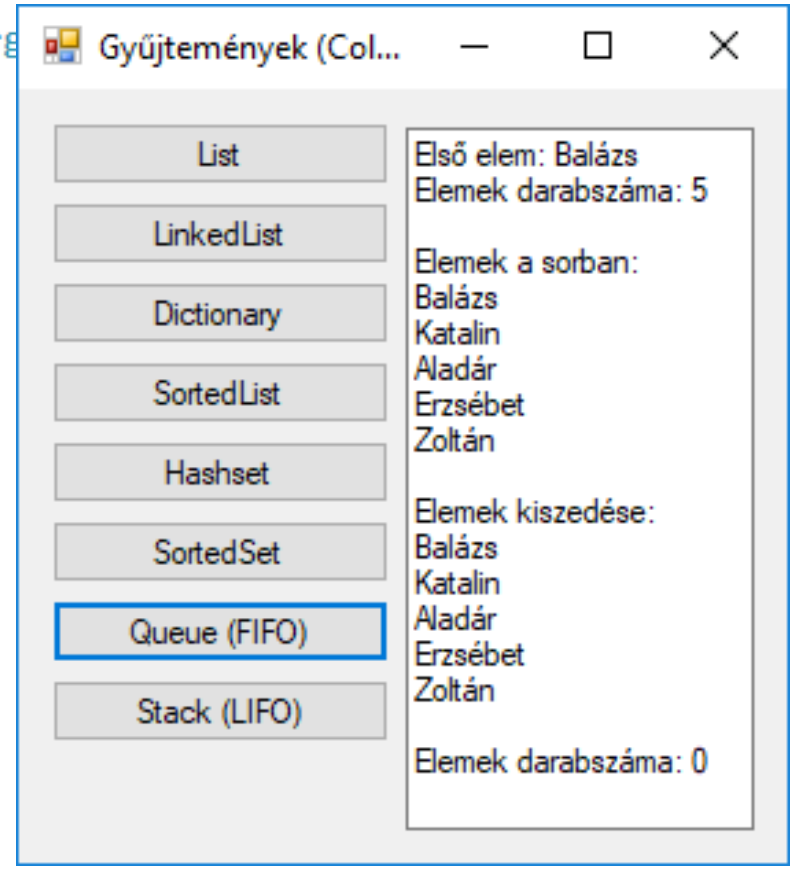
```
private void button5_Click(object sender, Event
{
    HashSet<int> szamok = new HashSet<int>();
    szamok.Add(59);
    szamok.Add(23);
    szamok.Add(40);
    szamok.Add(18);
    szamok.Add(77);
    szamok.Add(40);
    szamok.Add(40);
    textBox1.Clear();
    foreach (int szam in szamok)
    {
        textBox1.AppendText(szam + ", ");
    }
    textBox1.AppendText("\r\n\r\n");
    textBox1.AppendText("Legkisebb elem: " + szamok.Min() + "\r\n");
    textBox1.AppendText("Legnagyobb elem: " + szamok.Max() + "\r\n");
    textBox1.AppendText("Számok összege: " + szamok.Sum() + "\r\n");
    textBox1.AppendText("Számok átlaga: " + szamok.Average() + "\r\n\r\n");
    int[] a = { 8, 40, 17, 23, 55 };
    szamok.UnionWith(a);
    textBox1.AppendText("Unió a { 8, 40, 17, 23, 55 } tömbbel:\r\n");
    foreach (int szam in szamok)
    {
        textBox1.AppendText(szam + ", ");
    }
}
```



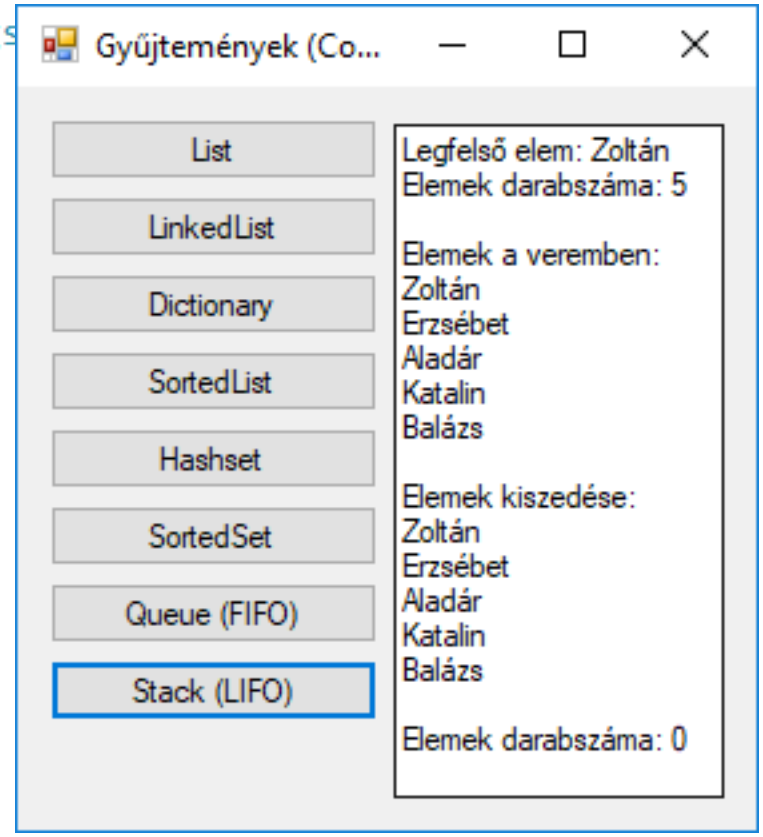
```
private void button6_Click(object sender, EventArg
{
    SortedSet<int> szamok = new SortedSet<int>();
    szamok.Add(59);
    szamok.Add(23);
    szamok.Add(40);
    szamok.Add(18);
    szamok.Add(77);
    szamok.Add(40);
    szamok.Add(40);
    textBox1.Clear();
    foreach (int szam in szamok)
    {
        textBox1.AppendText(szam + ", ");
    }
    textBox1.AppendText("\r\n\r\n");
    textBox1.AppendText("Legkisebb elem: " + szamok.Min() + "\r\n");
    textBox1.AppendText("Legnagyobb elem: " + szamok.Max() + "\r\n");
    textBox1.AppendText("Számok összege: " + szamok.Sum() + "\r\n");
    textBox1.AppendText("Számok átlaga: " + szamok.Average() + "\r\n\r\n");
    int[] a = { 8, 40, 17, 23, 55 };
    szamok.UnionWith(a);
    textBox1.AppendText("Unió a { 8, 40, 17, 23, 55 } tömbbel:\r\n");
    foreach (int szam in szamok)
    {
        textBox1.AppendText(szam + ", ");
    }
}
```



```
private void button7_Click(object sender, EventArg
{
    Queue<string> nevek = new Queue<string>();
    nevek.Enqueue("Balázs");
    nevek.Enqueue("Katalin");
    nevek.Enqueue("Aladár");
    nevek.Enqueue("Erzsébet");
    nevek.Enqueue("Zoltán");
    textBox1.Clear();
    textBox1.AppendText("Első elem: " +
        nevek.Peek() + "\r\n");
    textBox1.AppendText("Elemek darabszáma: " +
        nevek.Count + "\r\n\r\n");
    textBox1.AppendText("Elemek a sorban:\r\n");
    foreach (string s in nevek)
    {
        textBox1.AppendText(s + "\r\n");
    }
    textBox1.AppendText("\r\n");
    textBox1.AppendText("Elemek kiszedése: \r\n");
    while (nevek.Count > 0)
    {
        textBox1.AppendText(nevek.Dequeue() + "\r\n");
    }
    textBox1.AppendText("\r\n");
    textBox1.AppendText("Elemek darabszáma: " + nevek.Count);
}
```



```
private void button8_Click(object sender, EventArgs
{
    Stack<string> nevek = new Stack<string>();
    nevek.Push("Balázs");
    nevek.Push("Katalin");
    nevek.Push("Aladár");
    nevek.Push("Erzsébet");
    nevek.Push("Zoltán");
    textBox1.Clear();
    textBox1.AppendText("Legfelső elem: " +
        nevek.Peek() + "\r\n");
    textBox1.AppendText("Elemek darabszáma: " +
        nevek.Count + "\r\n\r\n");
    textBox1.AppendText("Elemek a veremben:\r\n");
    foreach (string s in nevek)
    {
        textBox1.AppendText(s + "\r\n");
    }
    textBox1.AppendText("\r\n");
    textBox1.AppendText("Elemek kiszedése: \r\n");
    while (nevek.Count > 0)
    {
        textBox1.AppendText(nevek.Pop() + "\r\n");
    }
    textBox1.AppendText("\r\n");
    textBox1.AppendText("Elemek darabszáma: " + nevek.Count);
}
```



Összefoglalás:

- Generikus osztályok (Generics):
  - Verem<T>
  - VeremEnumerator<T>
- Gyűjtemények (Collections):
  - List<T>
  - LinkedList<T>
  - Dictionary<K,V>
  - SortedList<K,V>
  - HashSet<T>
  - SortedSet<T>
  - Queue<T>
  - Stack<T>