

Programozás C# nyelven

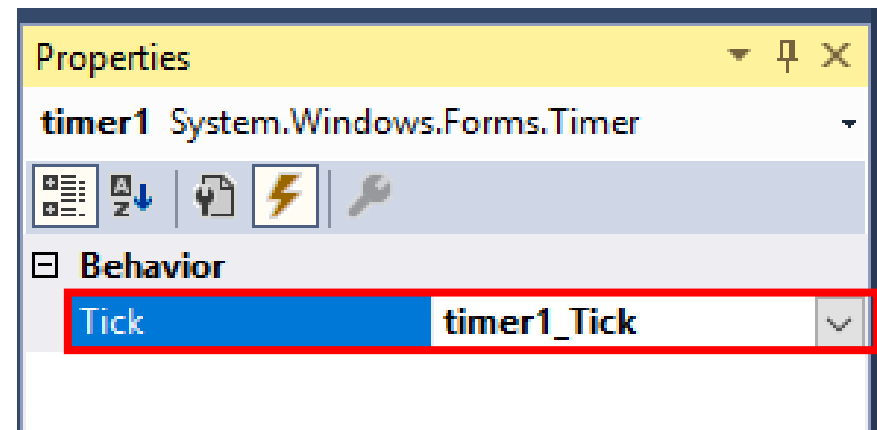
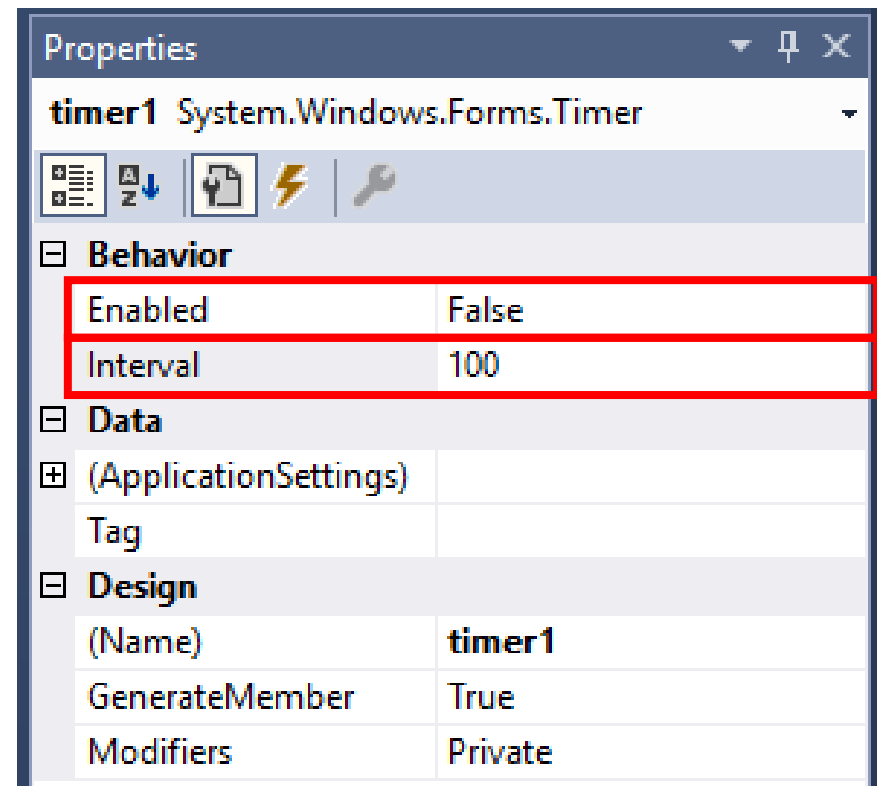
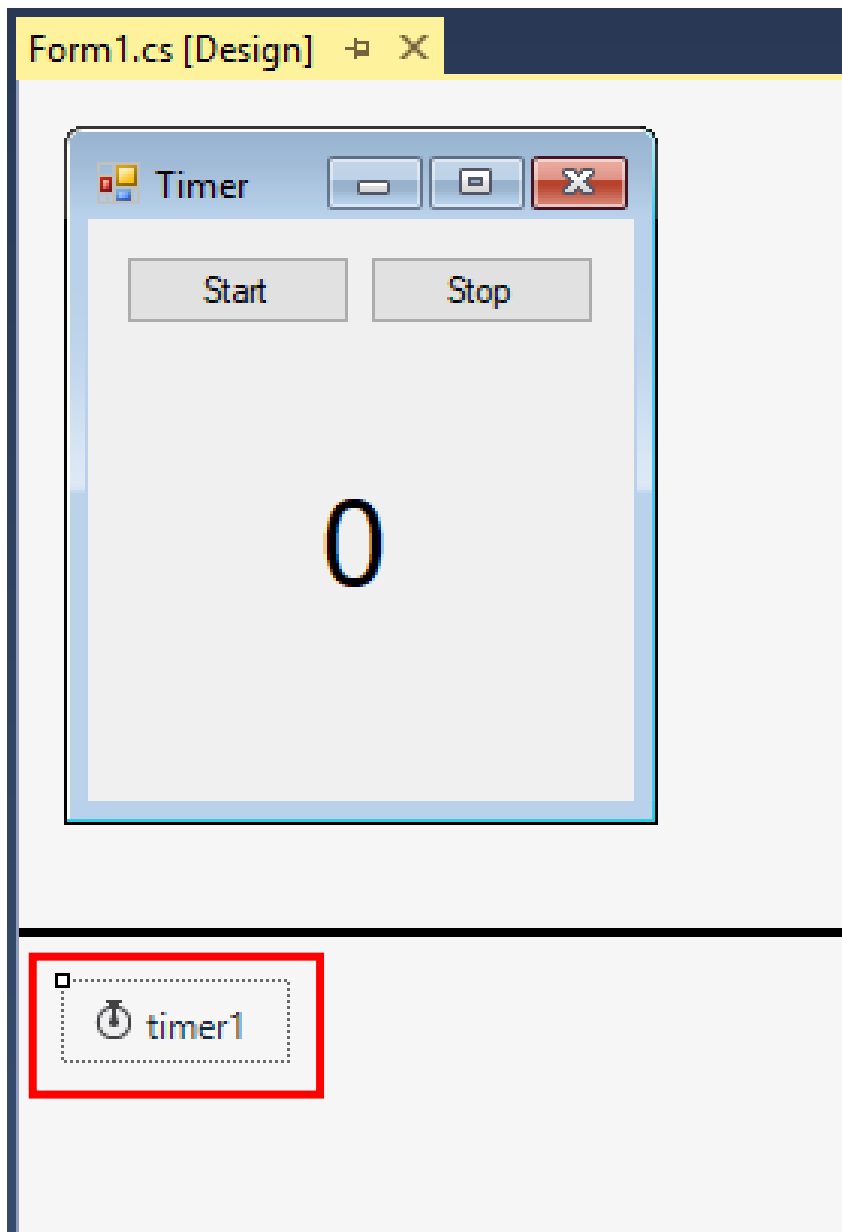
8. előadás



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

Timer (időzítő) használata:

030 Timer



```
private int szamlalo = 0;
```

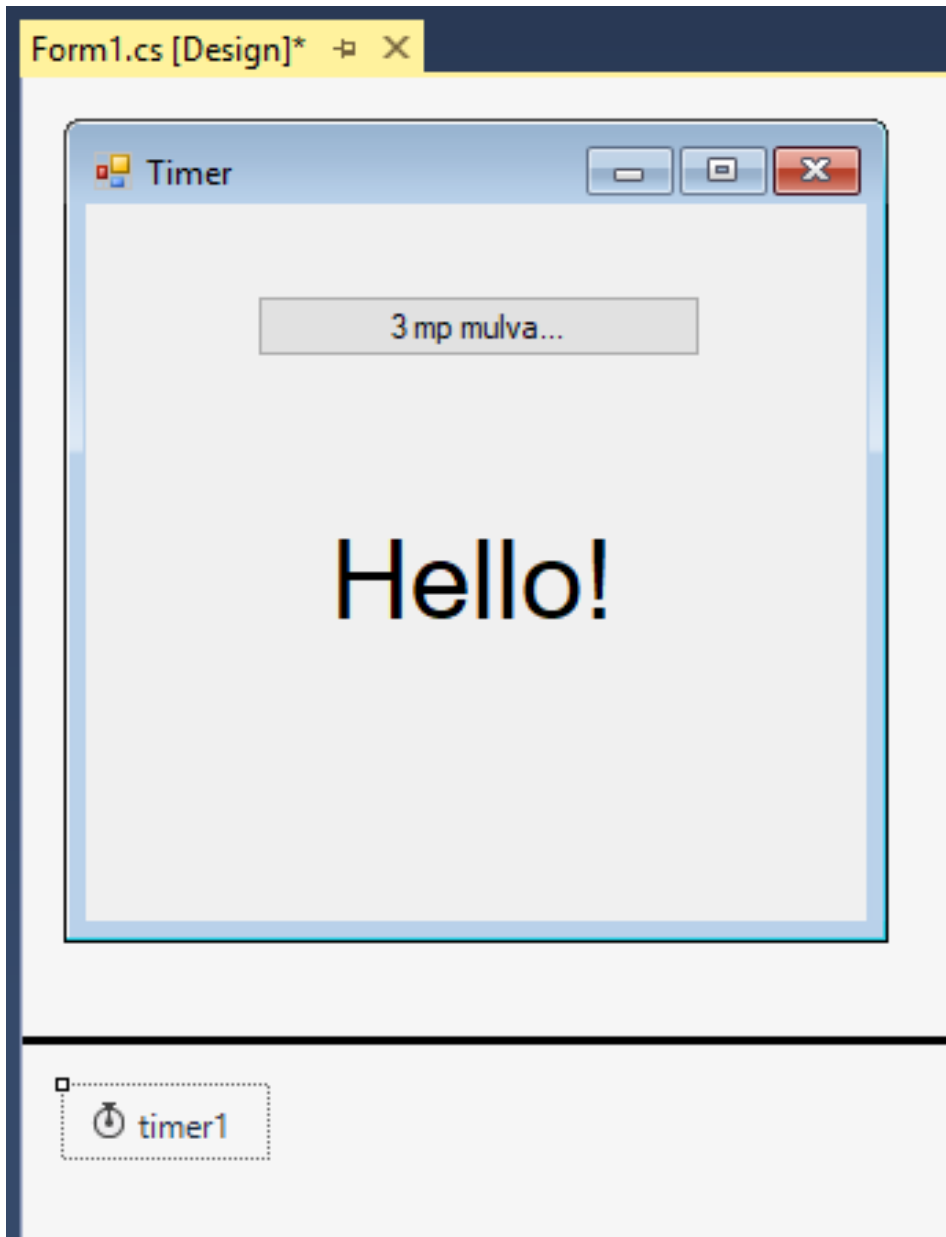
```
private void timer1_Tick(object sender, EventArgs e)
{
    szamlalo++;
    label1.Text = szamlalo.ToString();
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
}
```

Timer használata késleltetésre:

031 Timer (késleltetes)



`label1.Visible = false;`

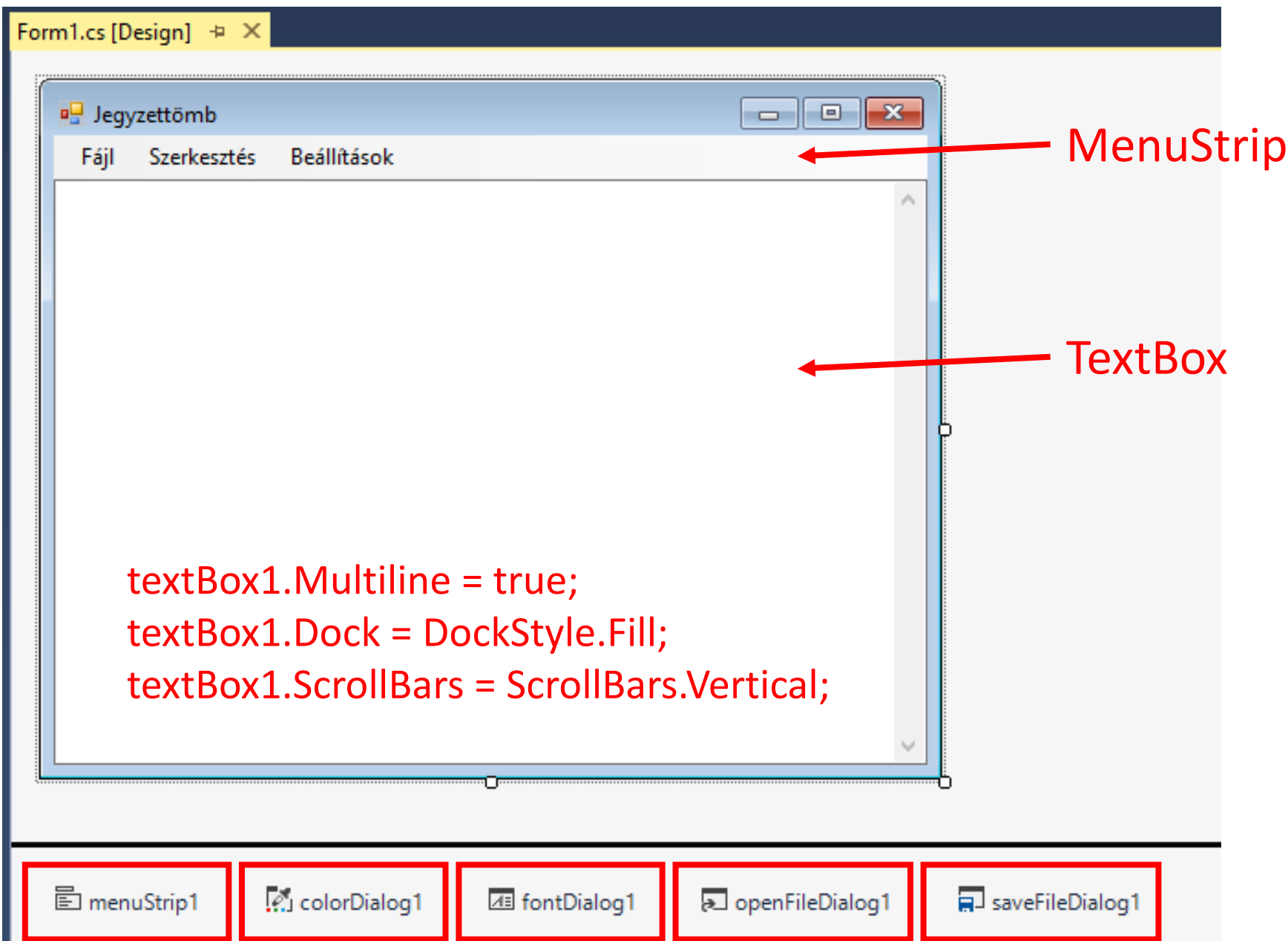
`timer1.Interval = 3000;`

```
private void button1_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;
    button1.Enabled = false;
}
```

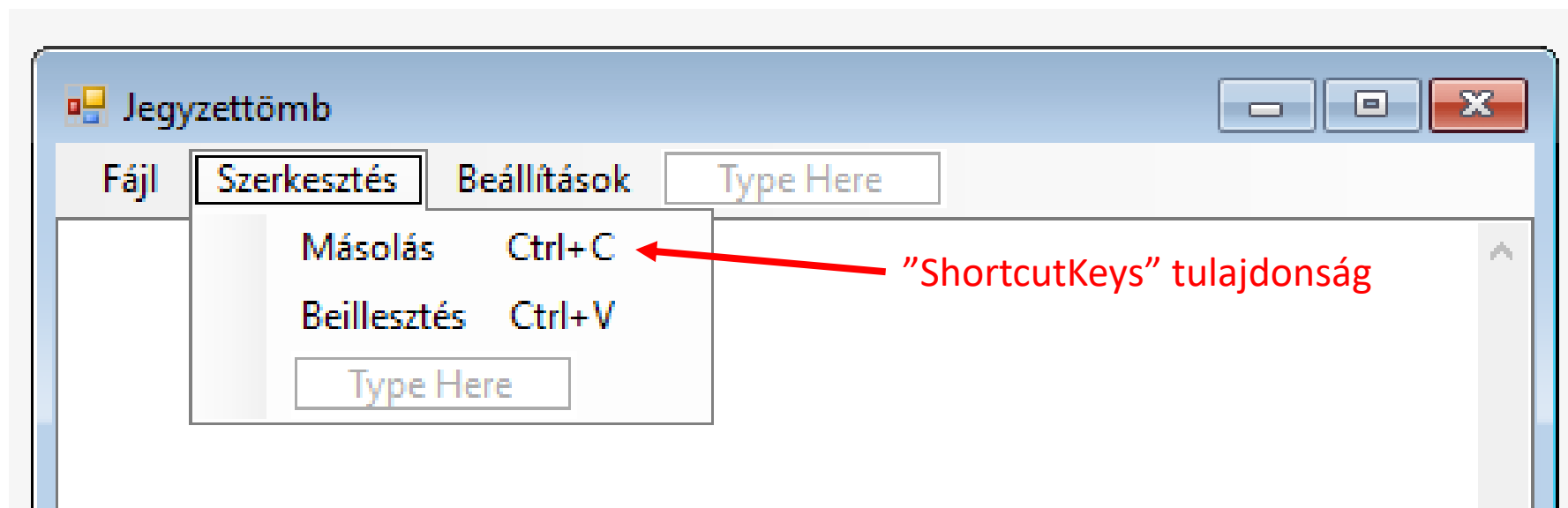
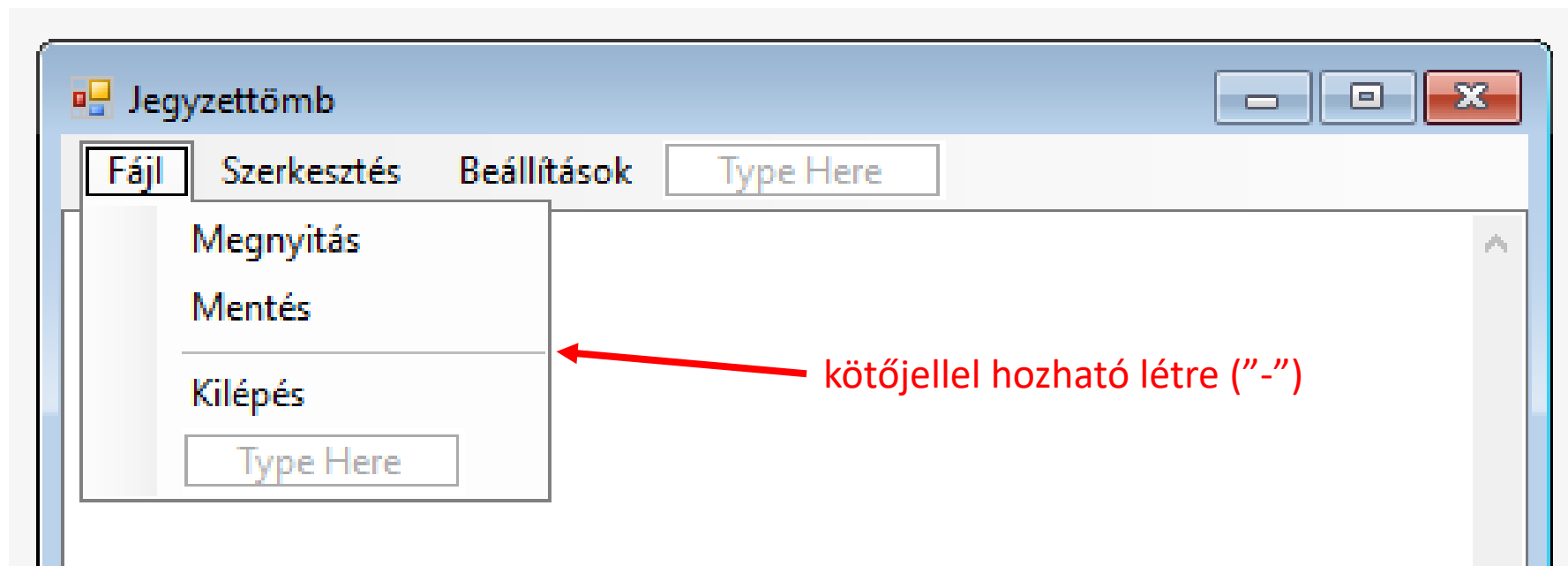
```
private void timer1_Tick(object sender, EventArgs e)
{
    timer1.Enabled = false;
    label1.Visible = true;
}
```

Jegyzetömb alkalmazás létrehozása:

032 Jegyzetömb

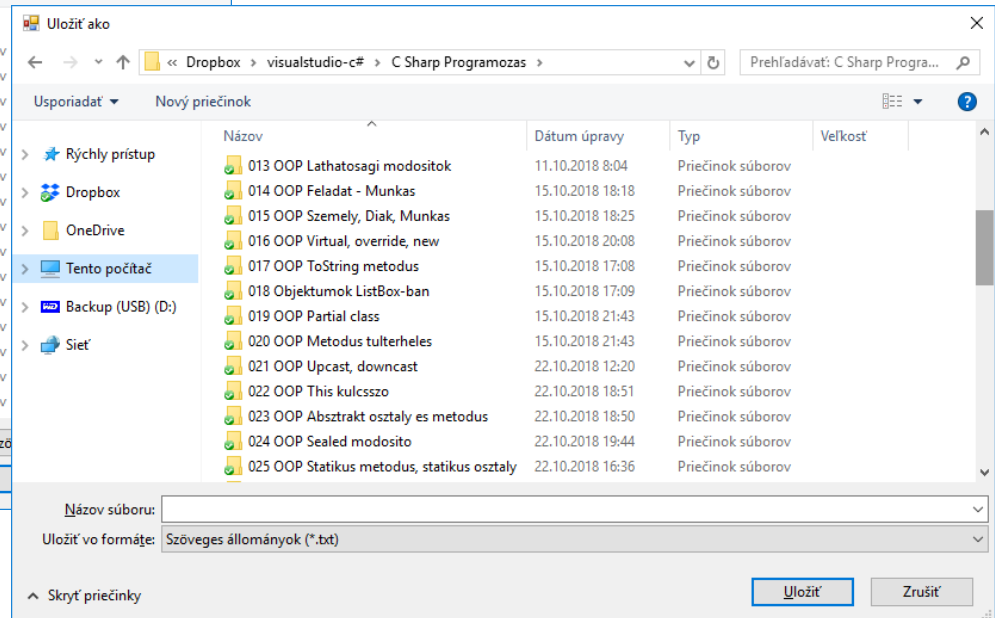
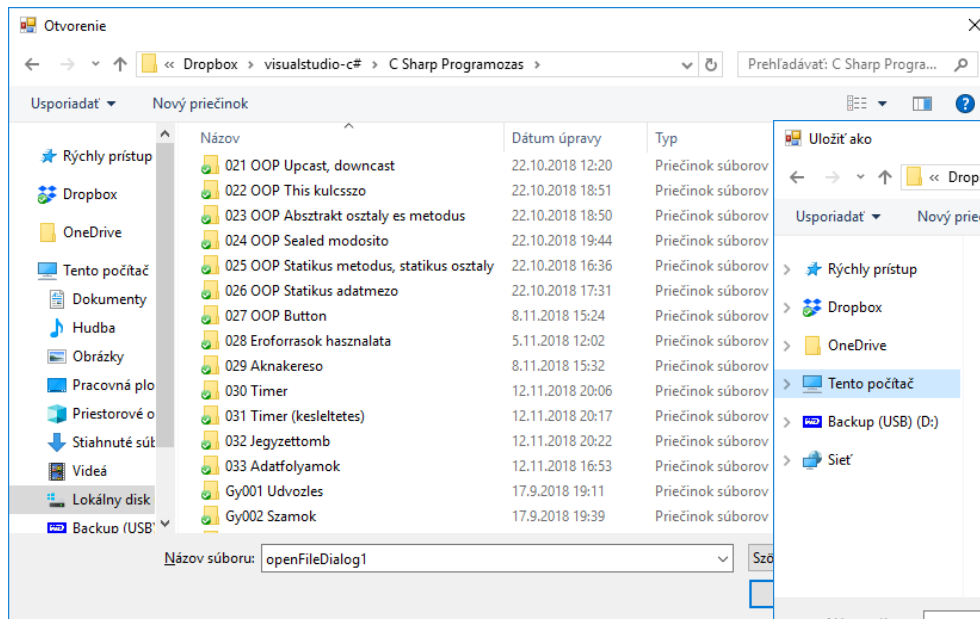
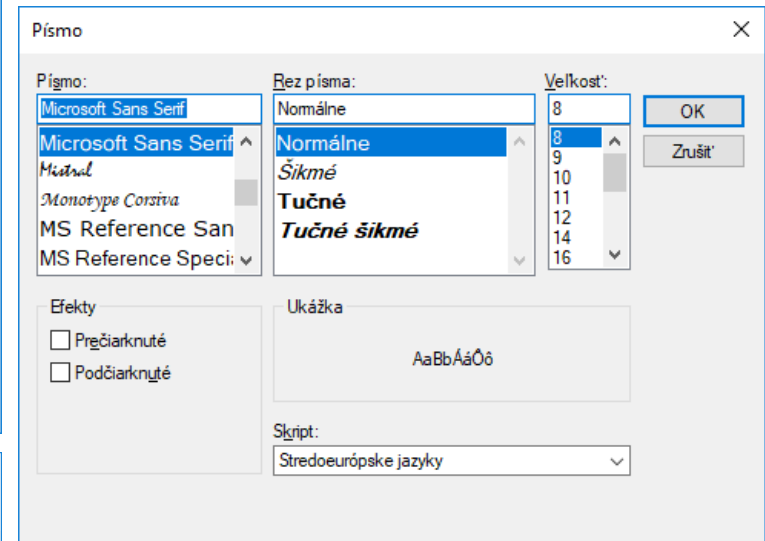
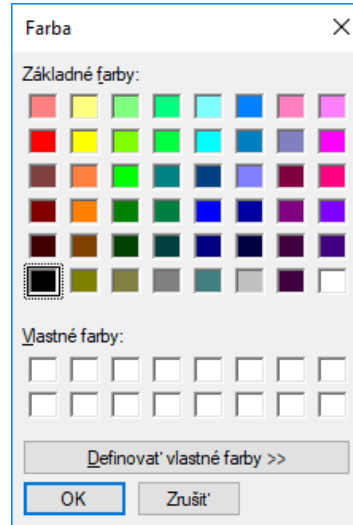


Menü létrehozása:



Dialógusablakok:

- ColorDialog
- FontDialog
- OpenFileDialog
- SaveFileDialog




```
private void betűtípusToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (fontDialog1.ShowDialog() == DialogResult.OK)
    {
        textBox1.Font = fontDialog1.Font;
    }
}
```

```
private void betűszínToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        textBox1.ForeColor = colorDialog1.Color;
    }
}
```

```
private void megnyitasToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        textBox1.Text = File.ReadAllText(openFileDialog1.FileName);
    }
}
```

```
private void mentésToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        File.WriteAllText(saveFileDialog1.FileName, textBox1.Text);
    }
}
```

Kilépés menüpont:

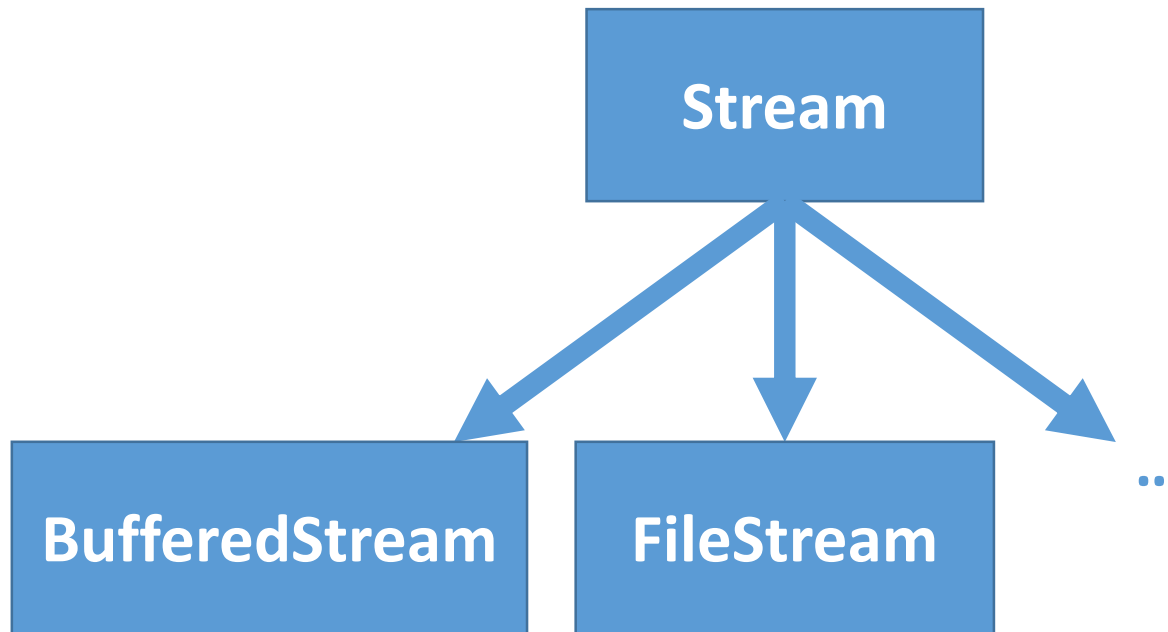
```
private void kilépésToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Másolás és Beillesztés menüpontok:

```
private void másolásToolStripMenuItem_Click(object sender, EventArgs e)
{
    textBox1.Copy();
}
```

```
private void beillesztésToolStripMenuItem_Click(object sender, EventArgs e)
{
    textBox1.Paste();
}
```

Adatfolyam (Stream) osztályok:

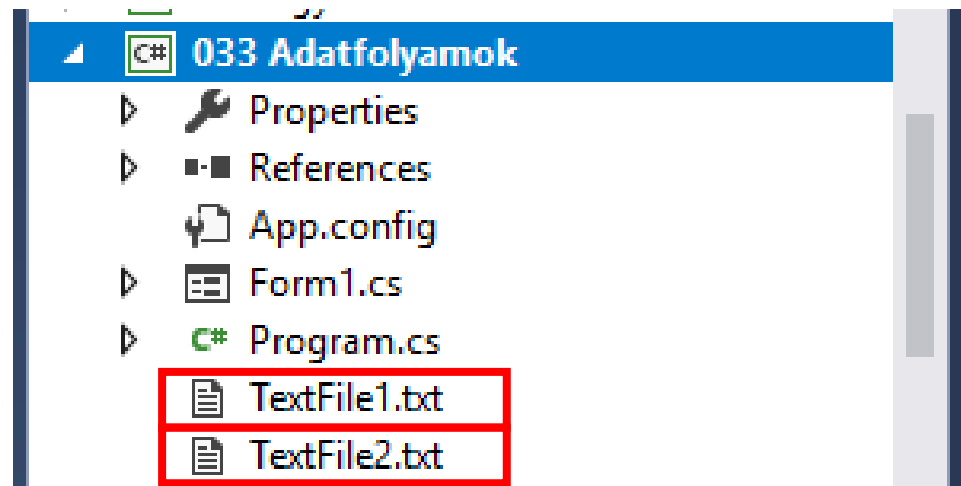
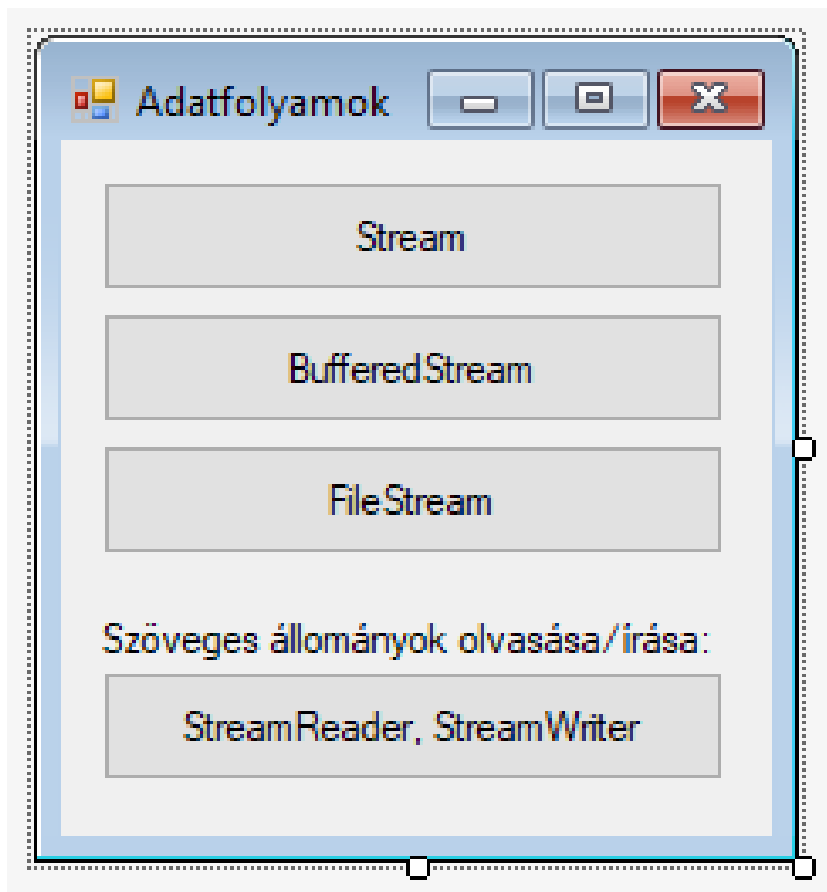


Szöveges állományok olvasása / írása:



Állományok olvasása/írása különféleképpen:

033 Adatfolyamok



```
const string fileNev1 = "../../../TextFile1.txt"; // állomány, amelybol olvasni fogunk
const string fileNev2 = "../../../TextFile2.txt"; // állomány, amelybe irni fogunk
const int BufferMerete = 1024; // buffer merete bajtokban
```

```
private void button1_Click(object sender, EventArgs e)
{
    if (!File.Exists(fileNev1))
    {
        MessageBox.Show(fileNev1 + " állomány nem található!", "Hiba!",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        // Stream objektumok használata
        Stream inStream = File.OpenRead(fileNev1);
        Stream outStream = File.OpenWrite(fileNev2);
        // olvasashoz/irashoz használt buffer létrehozása
        byte[] buffer = new Byte[BufferMerete];
        // olvas/iras
        int numBytes;
        while ((numBytes = inStream.Read(buffer, 0, BufferMerete)) > 0)
        {
            outStream.Write(buffer, 0, numBytes);
        }
        // adatfolyamok bezarasa
        inStream.Close();
        outStream.Close();
    }
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    if (!File.Exists(fileNev1))
    {
        MessageBox.Show(fileNev1 + " állomány nem található!", "Hiba!",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        // Stream es BufferedStream objektumok használata
        Stream inStream = File.OpenRead(fileNev1);
        Stream outStream = File.OpenWrite(fileNev2);
        BufferedStream inBuffStream = new BufferedStream(inStream);
        BufferedStream outBuffStream = new BufferedStream(outStream);
        // olvasashoz/irashoz használt buffer létrehozása
        byte[] buffer = new Byte[BufferMerete];
        // olvas/iras
        int numBytes;
        while ((numBytes = inBuffStream.Read(buffer, 0, BufferMerete)) > 0)
        {
            outBuffStream.Write(buffer, 0, numBytes);
        }
        // adatfolyamok bezarasa
        inBuffStream.Close();
        outBuffStream.Close();
    }
}
```

```
private void button3_Click(object sender, EventArgs e)
{
    if (!File.Exists(fileNev1))
    {
        MessageBox.Show(fileNev1 + " állomány nem található!", "Hiba!",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        // FileStream objektumok használata
        FileStream inStream = new FileStream(fileNev1, FileMode.Open, FileAccess.Read);
        FileStream outStream = new FileStream(fileNev2, FileMode.Create, FileAccess.Write);
        // olvasashoz/irashoz használt buffer létrehozása
        byte[] buffer = new Byte[BufferMerete];
        // olvas/iras
        int numBytes;
        while ((numBytes = inStream.Read(buffer, 0, BufferMerete)) > 0)
        {
            outStream.Write(buffer, 0, numBytes);
        }
        // adatfolyamok bezarasa
        inStream.Close();
        outStream.Close();
    }
}
```



```
private void button4_Click(object sender, EventArgs e)
{
    if (!File.Exists(fileNev1))
    {
        MessageBox.Show(fileNev1 + " állomány nem található!", "Hiba!",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        // StreamReader, StreamWriter objektumok használata (TXT állományoknál)
        StreamReader sr = File.OpenText(fileNev1);
        StreamWriter sw = File.CreateText(fileNev2);
        // olvasas/iras soronkent
        String sor;
        while ((sor = sr.ReadLine()) != null)
        {
            sw.WriteLine(sor);
        }
        // adatfolyamok bezarasa
        sr.Close();
        sw.Close();
    }
}
```

Összefoglalás:

- Timer
- MenuStrip
- FontDialog, ColorDialog, OpenFileDialog, SaveFileDialog
- TextBox-ben levő szöveg mentése/beolvasása (File osztállyal)
- Adatfolyamok (Stream, BufferedStream, FileStream)
- StreamReader , StreamWriter