



**BME**  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**KJIT**  
Közlekedés- és Járműirányítási Tanszék

# Programozás

## BMEKOKAA146

Dr. Bécsi Tamás

8. előadás

# Visszatekintés

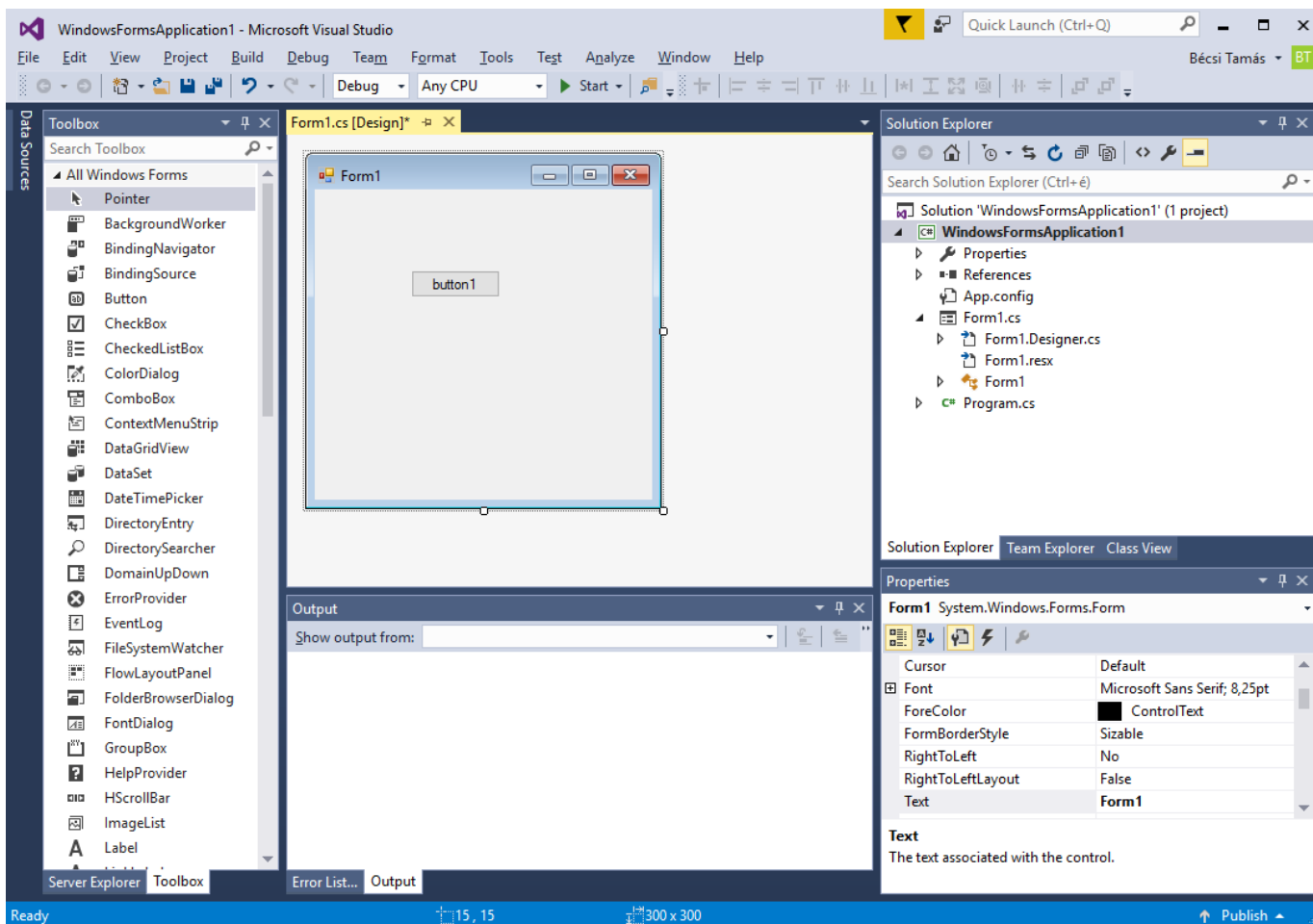
- A Windows Console alkalmazások egy karakteres képernyőt biztosítottak, ahol a kimenet a kiírt szöveg, míg a bemenet a billentyűzet volt.
- A megírt program a Main függvényben indult, ahol egymás után végrehajtotta az utasításokat, elágazásokat, ciklusokat és meghívta a függvényeket, ahonnan visszatérve folytatta tovább a futását. Végül a Main végére érve kilépett és a konzol ablakot az operációs rendszer bezárta.
- A program futását felhasználói interakció kérésével (`Console.ReadKey()` és `Console.ReadLine()`) lehetett megszakítani. Ebben az esetben a program egy billentyű, vagy az Enter leütésig várakozott.
- Az eredményeinket szöveges formában file-ba, vagy a képernyőre tudtuk kiíratni (`Console.Write()` és `Console.WriteLine()`).

# Windows Forms alkalmazások

- A Windows Forms alkalmazások grafikus felhasználói felülettel (GUI) rendelkező, objektum-orientált, eseményvezérelt programok.
- **GUI:** a programunk egy vagy több ablaka (form), melyekre további komponenseket helyezhetünk el. A komponenseket felrakhatjuk előre a grafikus tervezői felületen, vagy létrehozhatjuk futásidőben.
  - A komponensek tulajdonságokkal és eseményekkel rendelkeznek, melyek definiálják a komponens kinézetét és működését.
- **Eseményvezérelt program:** az eddig tanultaktól eltérően, az eseményvezérelt program nem fog folyamatosan utasításokat végrehajtani, hanem indulás után létrehozza a felhasználói felületét, majd interakcióra (eseményre) vár. Ez érkezhets **kívülről** (valamely perifériáról), vagy **belülről** (időzítő segítségével). Esemény érkezésekor lefut a korábban hozzárendelt és megírt eseménykezelő függvény. Ezek a függvények szigorúan definiált paraméterlistával rendelkeznek, amely tartalmazza az esemény kiváltóját (küldőjét) és a küldőre jellemző egyéb információkat (pl.: egérmutató koordinátája, lenyomott billentyű kódja stb.).

# Visual Studio tervező nézet

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járőrnyitási Tanszék



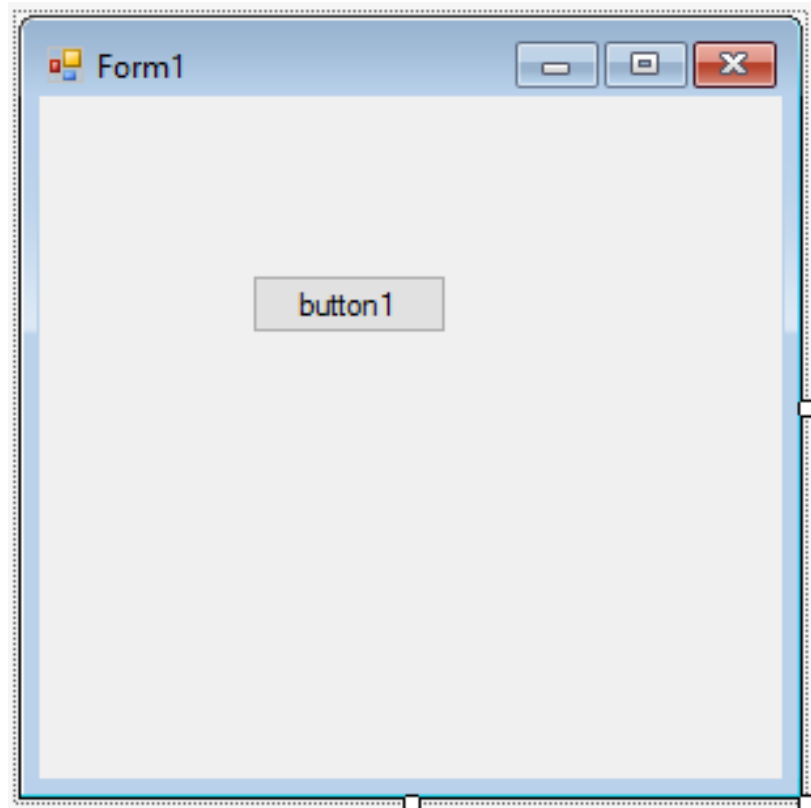
# Alapok

- Projekt létrehozása:
  - File - New Project - (Visual C#) – Windows Forms Application
- Kezelőfelület:
  - Solution Explorer
  - Toolbox
  - Design/Code View
  - Properties/Events
  - Program elemei

# Komponensek bevezetés

## A Form, Button osztályok

- Egyszerű tulajdonságok:
  - Text
  - Width Height
- Alapvető események:
  - Load
  - Click



# A Form Komponens (System.Windows.Forms.Form)

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék

tulajdonság	<b>int Width</b>	Szélesség Pixelben
tulajdonság	<b>int Height</b>	Magasság pixelben
tulajdonság	<b>int Top</b>	pozíció (Form teteje a tároló tetejétől)
tulajdonság	<b>int Bottom</b>	pozíció (Form alja a tároló tetejétől)
tulajdonság	<b>int Left</b>	pozíció (Form bal széle a tároló bal szélétől)
tulajdonság	<b>int Right</b>	pozíció (Form jobb széle a tároló bal szélétől)
tulajdonság	<b>string Text</b>	Fejlécfelirat
tulajdonság	<b>bool Enabled</b>	Engedélyezve
tulajdonság	<b>Color BackColor</b>	Háttérszín
tulajdonság	<b>Color ForeColor</b>	Előtér színe
tulajdonság	<b>bool Visible</b>	Látható
esemény	<b>Load</b>	A Form betöltésekor
esemény	<b>Click</b>	A Form-ra klikkeléskor

# MessageBox

- `public static DialogResult Show( string text )`
  - Például: `MessageBox.Show("Helló!");`
- `public static DialogResult Show( string text, string caption )`
  - Például: `MessageBox.Show("Helló!", "De most van cím is");`
- `public static DialogResult Show( string text, string caption, MessageBoxButtons buttons )`



# MessageBox

## MessageBoxButtons

AbortRetryIgnore	Abort, Retry, and Ignore
OK	OK button.
OKCancel	OK and Cancel buttons.
RetryCancel	Retry and Cancel buttons.
YesNo	Yes and No buttons.
YesNoCancel	Yes, No, and Cancel buttons.

## DialogResult

Abort
Cancel
Ignore
No
None
OK
Retry
Yes

```
DialogResult result;
```

```
result = MessageBox.Show("Nyomj egy gombot", "demo",  
                          MessageBoxButtons.OKCancel);
```

```
if (result == DialogResult.OK)  
    MessageBox.Show("OK");
```

# A Button Komponens (System.Windows.Forms.Button)

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék

tulajdonság	<b>int Width</b>	Szélesség Pixelben
tulajdonság	<b>int Height</b>	Magasság pixelben
tulajdonság	<b>int Top</b>	pozíció (Button teteje a tároló tetejétől)
tulajdonság	<b>int Bottom</b>	pozíció (Button alja a tároló tetejétől)
tulajdonság	<b>int Left</b>	pozíció (Button bal széle a tároló bal szélétől)
tulajdonság	<b>int Right</b>	pozíció (Button jobb széle a tároló bal szélétől)
tulajdonság	<b>string Text</b>	Fejlécfelirat
tulajdonság	<b>bool Enabled</b>	Engedélyezve
tulajdonság	<b>Color BackColor</b>	Háttérszín
tulajdonság	<b>Color ForeColor</b>	Előtér színe
tulajdonság	<b>bool Visible</b>	Látható
esemény	<b>Click</b>	Szélesség Pixelben

# A TextBox Komponens (System.Windows.Forms.TextBox)

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járőrirányítási Tanszék

tulajdonság	int Width, Height	Méret
tulajdonság	int Top, Bottom, Left, Right	Pozíció
tulajdonság	string Text	Fejlécfelirat
tulajdonság	bool Enabled	Engedélyezve
tulajdonság	Color BackColor, Forecolor	Szinek
tulajdonság	bool Visible	Látható
tulajdonság	bool MultiLine	
metódu	AppendText(string s)	szöveg hozzáfűzése
esemény	TextChanged	Szöveg megváltozásakor

# is, as operátorok

- Az is operátor ellenőrzi, hogy egy objektum kompatibilis-e egy típussal. Típusa boolean
  - használata: `<változó> is <típus>`
- Az as operátor kompatibilis objektumok közötti típus „konverzióra” ad lehetőséget.
  - használata: `<változó> as <típus>`

# A sender

```
private void button1_Click(object sender,
EventArgs e)
{
    if (sender is Button)
    {
        Button b = sender as Button;
        MessageBox.Show("a " + b.Text +
            " feliratú gombot nyomtad meg");
    }
}
```