

## Tanulók adatainak kezelése (szöveges fájl kezelésével)

Készítsünk alkalmazást, amely a **tanulo.txt** állomány adatainak felhasználásával kezeli egy iskola diákjainak adatait.

Képek a program működéséről:

Válasszon települést!

Kattintson a sor elejére a kiválasztáshoz!

	Név	Igazolvány
--	-----	------------

☐ Igazolvány száma változik!

Igazolvány száma:  Osztály:

Név:

Született:  2017. március 1.

Lakcím:

Üzenet

Válasszon települést!

Budapest

Kattintson a sor elejére a kiválasztáshoz!

	Név	Igazolvány
<input type="checkbox"/>	Tremmel Orsolya	06552477487
<input type="checkbox"/>	Kósa Réka	10358548766
<input checked="" type="checkbox"/>	Nagy Árpád	12370577397
<input type="checkbox"/>	Kádár Marianna	13105657126
<input type="checkbox"/>	Balobás József	14450313769
<input type="checkbox"/>	Kovacs Dani	15268712628
<input type="checkbox"/>	Csoboth Franciska	15412684283
<input type="checkbox"/>	Lengyel Éva	15456452123

☐ Igazolvány száma változik!

Igazolvány száma:  Osztály:

Név:

Született:

Lakcím:

A kiválasztott adatokat módosíthatja!

A **tanulo.txt** állomány a következő adatokat tartalmazza soronként pontosvesszővel elválasztva:

**diákigazolvány száma;tanulo neve;születési idő;osztály;irányítószám;város;utca**

05436623775;Lorsch Gyula;1993.09.25. 0:00:00;10A;4541;Nyírájkó;Kakukkhegyi út 93.

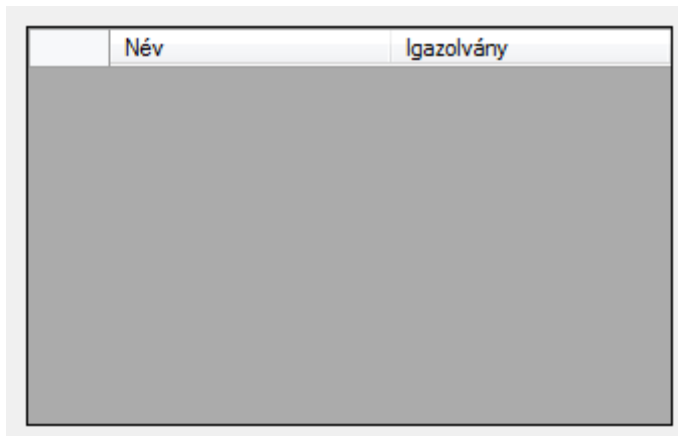
06552477487;Tremmel Orsolya;1991.08.18. 0:00:00;12A;1221;Budapest;Bánk bán u. 61.

10358548766;Kósa Réka;1991.08.22. 0:00:00;12A;1188;Budapest;Pázsit u. 51.

### A program funkciói:

- induláskor beolvassa az adatokat a fájlból és eltárolja egy struktúra alapú listába
- ezzel párhuzamosan töltse be a listában lévő településeket (mindegyiket csak egyszer) a legördülő listába
- a listában kiválasztott településhez tartozó tanulók nevét és igazolványszámát jelenítse meg a táblázatban
- a táblázat kiválasztott sorának elejére (header) kattintva töltse be az űrlapba az adatokat
- az űrlapban módosított adatokat írja vissza a fájlba

### Új vezérlő datagridview



Az üresen létrehozott datagridview-hoz adjunk oszlopokat:

jobb gomb – Add Column...

A HEADER TEXT jelenik meg a táblázat fejlécében!

## A megoldás lépései

1. Hozzuk létre a **struktúrát**, **egy változót**, amit a módosításnál használunk majd, **és a listát** osztályszinten (több metódus is tudja használni!)

```
namespace Iskola2._0
{
    public partial class Form1 : Form
    {
        string regi_igazolvany_szama = null;
        public struct tanulo
        {
            public string diakig;
            public string nev;
            public DateTime szuletett;
            public string osztaly;
            public string irsz;
            public string varos;
            public string utca;
        }
        List<tanulo> tlista = new List<tanulo>();

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

2. Hozzunk létre **egy saját metódust**, ami a fájlból olvasást végzi (kétszer is használjuk majd, ezért kerül saját függvénybe!)

- A függvény soronként olvas, darabol és az ideiglenes példányt feltölti, és beírja a listába.
- feltölti a combobox-ot (a **.Contains** az elem meglétét ellenőrzi, így csak egyszer kerül bele minden város)

```
private void adat_betolt()
{
    FileStream fs = new FileStream("tanulo.txt", FileMode.Open);
    StreamReader sr = new StreamReader(fs);
    string sor = null;
    while (!sr.EndOfStream)
    {
        sor = sr.ReadLine();
        string[] darabok = sor.Split(';');

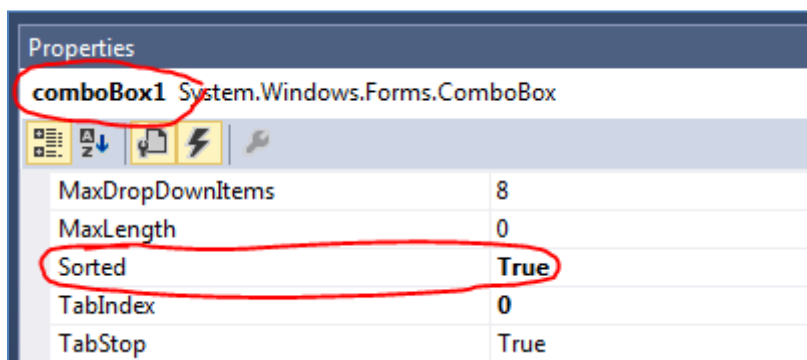
        tanulo t = new tanulo();

        t.diakig = darabok[0];
        t.nev = darabok[1];
        t.szulettett = Convert.ToDateTime(darabok[2]);
        t.osztaly = darabok[3];
        t.irszo = darabok[4];
        t.varos = darabok[5];
        t.utca = darabok[6];

        //töltsük fel a legördülő listát:
        if (!comboBox1.Items.Contains(t.varos))
        {
            comboBox1.Items.Add(t.varos);
        }

        //adjuk hozzá a listához a tanulót:
        tlista.Add(t);
    }
    sr.Close();
    fs.Close();
}
```

**Ha a combobox sorted tulajdonságát TRUE-a állítjuk, rendezett lesz!**



3. A saját módszerünket hívjuk meg a form betöltésekor (*két kattintás a formra*)

```
private void Form1_Load(object sender, EventArgs e)
{
    adat_betolt();
}
```

**ITT ÉRDEMES TESZTELNI, A PROGRAM AZ INDULÁS UTÁN EGY FELTÖLTÖTT LEGÖRDÜLŐ LISTÁVAL VÁR A TOVÁBBIAKRA...**

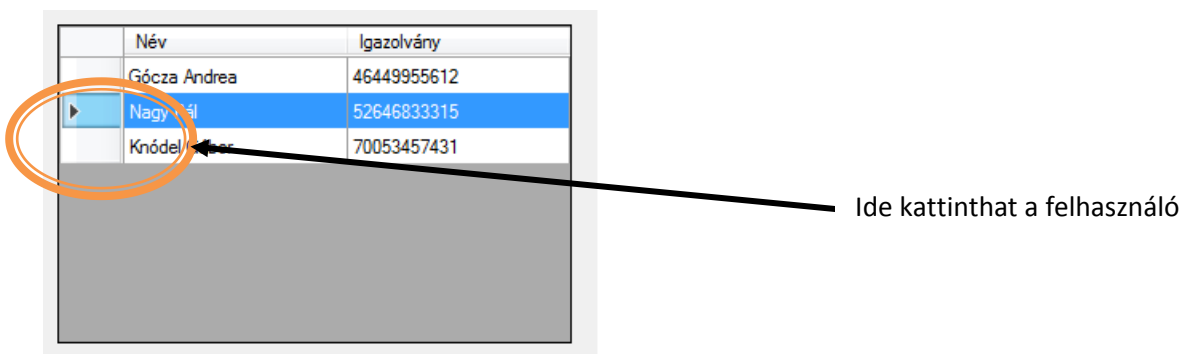
4. Mi történik, ha a listából kiválasztom az egyik települést? (*két kattintás a tervezőn a comboboxra!*)

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    //az összes adat törlése a táblázatból
    dataGridView1.Rows.Clear();

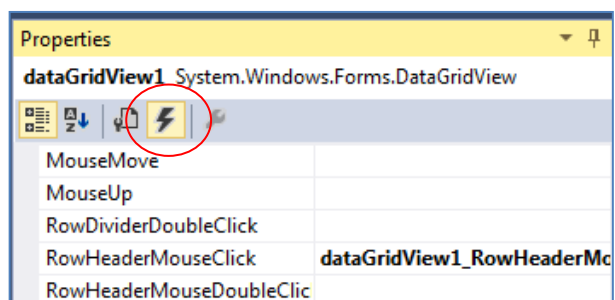
    //a kiválasztott város
    string varos = comboBox1.SelectedItem.ToString();

    //az adott városban lakók listázása a táblázatba
    foreach (tanulo t in tlista)
    {
        if (t.varos == varos)
        {
            dataGridView1.Rows.Add(t.nev, t.diakig);
        }
    }
}
```

5. A dataGridView egyik sorának kiválasztásával jelölheti meg a felhasználó, hogy kinek az adataira kíváncsi.



Ehhez a **RowHeaderMouseClick** módszerra van szükségünk. A datagridview tulajdonságainál:



```

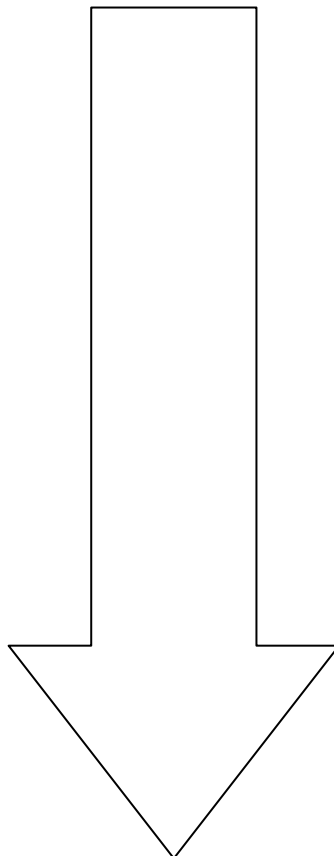
private void dataGridView1_RowHeaderMouseClick(object sender, DataGridViewCellEventArgs e)
{
    DataGridViewRow sor = dataGridView1.SelectedRows[0];
    string diakig = sor.Cells[1].Value.ToString();

    //keressük meg a tanulót igazolványszám alapján:
    bool megvan = false;
    int index = 0;
    while (!megvan)
    {
        if (tlista[index].diakig == diakig)
        {
            megvan = true;
        }
        else
        {
            index++;
        }
    }

    textBox1.Text = tlista[index].diakig;
    regi_igazolvany_szama = tlista[index].diakig;
    textBox2.Text = tlista[index].osztaly;
    textBox3.Text = tlista[index].nev;
    //textBox4.Text = tlista[index].szuletett.ToString();
    dateTimePicker1.Value = tlista[index].szuletett;
    textBox4.Text = tlista[index].irsz;
    textBox5.Text = tlista[index].varos;
    textBox6.Text = tlista[index].utca;
    label6.Text = "A kiválasztott adatokat módosíthatja!";
}

```

Eltároljuk a régi igazolvány számát, hátha pont azt módosítja és nem találjuk meg a listában ☺



## 6. A módosítás gombra kattintva

```
private void button1_Click(object sender, EventArgs e)
{
    //keressük meg a módosítandó rekordot a listában
    bool megvan = false;
    int index = 0;
    while (!megvan)
    {
        if (tlista[index].diakig == regi_igazolvany_szama)
        {
            megvan = true;
        }
        else
        {
            index++;
        }
    }

    //beolvassuk az adatokat a formról:
    tanulo t = new tanulo();
    t.diakig = textBox1.Text;
    t.osztaly = textBox2.Text;
    t.nev = textBox3.Text;
    t.szuletett = dateTimePicker1.Value;
    t.irszo = textBox4.Text;
    t.varos = textBox5.Text;
    t.utca = textBox6.Text;

    //töröljük a listából a tanulót
    tlista.RemoveAt(index);
    //hozzáadjuk a végéhez a módosított adatokat
    tlista.Add(t);

    //írjuk vissza fájlba a tartalmat
    FileStream fs = new FileStream("tanulo.txt", FileMode.Create);
    StreamWriter sw = new StreamWriter(fs);
    foreach (tanulo diak in tlista)
    {
        sw.WriteLine(diak.diakig + ";" + diak.nev + ";"
            + diak.szuletett + ";" + diak.osztaly + ";"
            + diak.irszo + ";"
            + diak.varos + ";" + diak.utca);
    }
    sw.Close();
    fs.Close();

    label6.Text = "Az adatokat frissítettem!";

    //újratöltjük az adatokat
    tlista.Clear();
    comboBox1.Items.Clear();
    comboBox1.Text = null;
    dataGridView1.Rows.Clear();
    adat_betolt();
}
```

## Kiegészítés

! A checkbox arra lehet jó, hogy az induláskor letiltott diákigazolvány számot módosítási igény esetén feloldja:

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    textBox1.Enabled = !textBox1.Enabled;
}
```

! **Vezessük** a felhasználót a vezérlők tiltásával, engedélyezésével!

! **Írjunk ki üzeneteket** az alsó label-be, hogy éppen mi történik, vagy mit tehet a felhasználó

## Fájlkezelés C# alapon

### Feladat: Fakultáció

A fakt.txt állomány pontosvesszővel elválasztott adatokat tartalmaz tanulók fakultáció választásáról. Egy sorban az összetartozó adatok szerepelnek: **név ; tantárgy; érdemjegy**

Készítsünk Windows Form alkalmazást az adatok kezelésére az alábbi elrendezésben:

The screenshot shows a Windows Form application with the following layout:

- Válasszon tantárgyat!** (Select subject!):
  - A dropdown menu.
  - A table with columns **Név** and **Osztályzat**. The first row has an asterisk (\*) in the first column.
  - Input fields for **Név:** and **Osztályzat:** (with a spinner control).
  - A **Módosít** (Modify) button.
- Válasszon tanulót!** (Select student!):
  - A dropdown menu.
  - A table with columns **Tantárgy** and **Osztályzat**. The first row has an asterisk (\*) in the first column.
  - Input fields for **Név:** and **Tantárgy:**.
  - A **Módosít** (Modify) button.
- Új tanuló felvétele** (New student admission):
  - Input fields for **Név:** and **Fakultáció:**.
  - A **Rögzít** (Save) button.

### Megjegyzések a feladat megoldásához:

1. Az adatokat az induláskor olvassa be a program (Form\_Load).
2. Hozzunk létre struktúrát és listát a tároláshoz.
3. Induláskor töltsük fel a bal oldali comboboxot a lehetséges tantárgyakkal!
4. A kiválasztott tantárgyhoz tartozó tanulókat listázzuk névvel, osztályzattal az alatta lévő táblázatba (datagridview)
5. A táblázat egy kiválasztott sorának adata kerüljön bele az alatta lévő mezőkbe, **ahol a név nem, de az osztályzat módosítható.**
6. A **MÓDOSÍT GOMB** írja felül **a listában és a comboboxokban** az adatokat!
7. Induláskor töltsük fel a jobb oldali comboboxot a tanulók nevével!
8. A kiválasztott tanulóhoz tartozó tanulókat listázzuk tantárggyal, osztályzattal az alatta lévő táblázatba (datagridview)
9. A táblázat egy kiválasztott sorának adata kerüljön bele az alatta lévő mezőkbe, **ahol a név és a tantárgy módosítható.**
10. A **MÓDOSÍT GOMB** írja felül **a listában és a comboboxokban** az adatokat!



**11.** A form alsó részén legyen lehetőségünk új tanulót rögzíteni. **A rögzítéssel frissüljön a fájl és a legördülő listák is!**

**A legördülő listák feltöltéséhez érdemes saját függvényt írni, ahogy az előző órán is tettük!**

(Iskola\_text\_fajlbol.pdf)

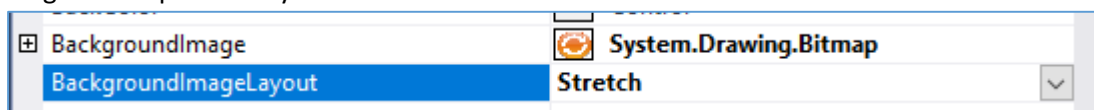
## Európa országai – Beadandó feladat

1. Készíts Windows Form alkalmazást Európa ország-adatainak kezelésére az alábbi elrendezésben:

The screenshot shows a Windows Form application titled "Európa országai". The interface includes a refresh button (FRISSÍTÉS) and a plus button (MÓDOSÍTÁS) on the left. A search bar labeled "Ország keresése" with a magnifying glass icon is on the top right. Below the buttons is a data grid with columns "Sorszám" and "Ország neve". To the right of the grid is a panel titled "Ország adatai" with input fields for "Név:", "Főváros:", "Terület:", "Lakosság:", "Érték:", and "Tagja az EU-nak:".

Megjegyzések a form-hoz:

- Az ábrák Button vezérlők, melyeknek a **BackgroundImage** tulajdonságában tárolhatod be a megfelelő kép-állományokat.



- A FRISSÍTÉS gomb a dataGridView vezérlő tartalmát frissíti
- A MÓDOSÍTÁS (+) gomb megnyomásával módosíthatjuk az adatokat (ki tudja, mi történik majd a következő években ☺)

2. Az adatok betöltéséhez használd az **Europa.txt** állományt a korábban tanult módon!

- Hozz létre egy struktúrát az adatok tárolására **ország** néven!
- Hozz létre egy **lista** néven, amely *ország típusú*, és alkalmas az adatok tárolására!
- A program indulásakor töltsd be az adatokat a fájlból a listába!

3. A program **üresen induljon**, csak a **FRISSÍTÉS** gomb legyen aktív!

4. A **FRISSÍTÉS** hatására **töltődjenek be** a táblázatba az országok megfelelő adatai!

5. Ekkor már legyen elérhető a többi gomb is.

6. A táblázatban **egy adott sor kiválasztásával** **jeljenek meg** az ország adatai a jobb oldali részben!

7. A **kereső mező kitöltésével és a KERESÉS gomb megnyomásával** jelenjenek meg a jobb oldali részben a keresett ország adatai!

8. A **MÓDOSÍTÁS gomb hatására** az űrlapon esetlegesen módosított adatokat írjuk a listába a korábbi módszerrel:

- töröljük a listából a régit
- írjuk a lista végére az új adatokat!