

1. Programozás 40 pont

Iskola¹

A `nevek.txt` állományban rögzítettük egy középiskola tanulóinak néhány adatát. Feltételezheti, hogy nincs két azonos nevű tanuló egy osztályban. Az állomány tartalma soronként:

- iskola kezdésének éve (2004-2007)
- az osztály betűjele (a-e)
- a diák neve (ékezetek nélkül).

Az adatok pontosvesszővel vannak elválasztva. Példa (részlet) a `nevek.txt` állományra:

```
2004;d;Vavrek Kristof
2006;e;Hidas Reka
2006;d;Kun Michael
```

Megoldásában:

- *A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 3. feladat)!*
- *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
- *Az ékezetmentes kiírások is elfogadottak.*
- *Az azonosítókat kis- és nagybetűkkel is kezdheti.*
- *A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.*
- *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!*

Oldja meg a következő feladatokat:

1. Készítsen `Iskola` néven **konzol típusú projektet**, melyben megoldja a következő feladatokat!
2. Olvassa be és tárolja el egy célszerűen megválasztott adatszerkezetben (pl.: vektor, lista) a `nevek.txt` állományban lévő adatokat!
3. Írja ki a képernyőre, hogy hány tanuló jár az iskolába!
4. Írja ki a képernyőre a minta szerint, hogy ki rendelkezik a leghosszabb névvel! A szóközöket ne számolja a név hosszához! Holtverseny esetén minden, leghosszabb névvel rendelkező tanuló neve jelenjen meg! Jelenítse meg a kiírásban a leghosszabb név/nevek hosszát is a minta szerint!
5. Az iskolai rendszergazdának egyedi azonosítókat kell készítenie a számítógép-hálózat használatához. Az azonosítókat a következő módon alakítja ki: első karaktere az évfolyam utolsó számjegye (pl.: 2006 esetén 6), következő karakter az osztály betűjele, majd a vezetékneve első három karaktere, végül első keresztnéve első három karaktere következik. Az azonosítóban mindenütt kisbetűk szerepelnek. Feltételezhetjük, hogy a vezetéknev és az első keresztnév legalább 3 karakteres. Készítsen jellemzőt vagy függvényt, melyben meghatározza a rendelkezésre álló adatokból a tanuló azonosítóját! Az azonosítókat ne tárolja! Az elkészített jellemzőt/függvényt felhasználva írja ki az adatszerkezetben tárolt első és utolsó tanuló azonosítóját a minta szerint!
6. Kérjen be egy azonosítót és állapítsa meg, hogy ki tarthat hozzá! A tanuló adatait írja a képernyőre! Ha nem talál megfelelő diákot, akkor a „Nincs megfelelő tanuló.” mondatot jelenítse meg a minta szerint!
7. A forráskódjába a meglévő osztálydefiníció fölé illessze be a `java.txt` vagy a `csharp.txt` forrásállományból a `JelszóGeneráló` osztályt definiáló kódrészletet, majd hozzon létre belőle egy osztálypéldányt (objektumot)! Az osztálypéldány konstruktorát a véletlenszámok előállítására alkalmas

¹ Forrás: <https://docplayer.hu/4402962-Informatika-emelt-szintu-probaerettsegi.html>

beépített objektummal paraméterezze! Válaszon ki véletlenszerűen egy tanulót, majd a JelszóGeneráló osztály Jelszó() metódusának hívásával írjon ki egy 8 karakter hosszú jelszót a minta szerint!

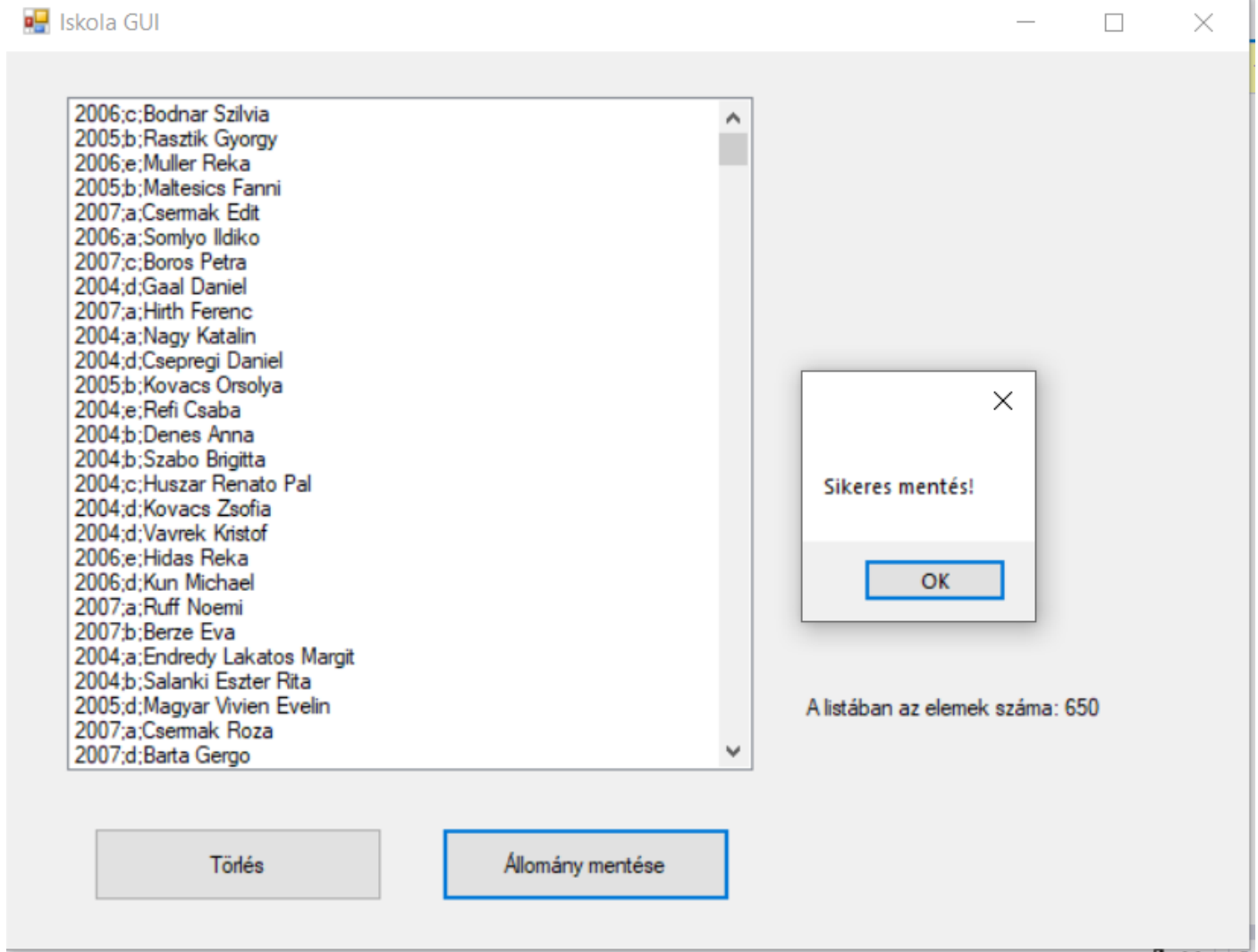
8. Készítsen **grafikus alkalmazást** a következő feladatok megoldására, melynek projektjét IskolaGUI néven mentse el!
9. Az alkalmazás grafikus felhasználói felületét alakítsa ki a minta szerint! Az ablak címsorában a „Iskola GUI” szöveget jelenítse meg!
10. Az ablakon található listába a program induláskor töltse be a nevekGUI.txt állomány sorait! A lista elemei a forrásfájl egy-egy sora (továbbiakban tanulója) legyen!
11. Oldja meg, hogy a kijelölt tanuló a „Törlés” parancsgomb lenyomása után törlésre kerüljön a listából! Ha a listában nincs kijelölt tanuló, akkor törléskor a „Nem jelölt ki tanulót!” szöveg jelenjen meg egy felugró ablakban!
12. Ha az „Állomány mentése” parancsgombra kattintunk, akkor történjen meg a listából a tanulók mentése a nevekNEW.txt állományba, melynek szerkezete a forrásállomány szerinti legyen! Ha a mentés sikeres volt, akkor a „Sikeres mentés!” felirat jelenjen meg egy felugró ablakban!

Minták a konzolos alkalmazás futására:

```
3. feladat:  
    Az iskolába 650 tanuló jár.  
4. feladat: A leghosszabb (25 karakter) nevű tanuló(k):  
    Karatson Patricia Mercedesz  
5. feladat: Azonosítók  
    Első: Bodnar Szilvia - 6cbodszi  
    Utolsó: Krizsan Vivien Evelin - 6ckriviv  
6. feladat: Kérek egy azonosítót [pl.: 4dvavkri]: 4atolrek  
2004      a      Tolcsvay-Nagy Reka Luca  
7. feladat: Jelszó generálása  
    Somlyo Martin - ibmy90lk
```

```
3. feladat:  
    Az iskolába 650 tanuló jár.  
4. feladat: A leghosszabb (25 karakter) nevű tanuló(k):  
    Karatson Patricia Mercedesz  
5. feladat: Azonosítók  
    Első: Bodnar Szilvia - 6cbodszi  
    Utolsó: Krizsan Vivien Evelin - 6ckriviv  
6. feladat: Kérek egy azonosítót [pl.: 4dvavkri]: 5cmodgab  
Nincs megfelelő tanuló.  
7. feladat: Jelszó generálása  
    Ignacz Balazs - 7rz5cuzg
```

Minták a grafikus alkalmazás futására:



Megoldás-konzolos:

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
/*A nevek.txt állományban rögzítettük egy középiskola tanulóinak néhány adatát.
 * Feltételezheti, hogy nincs két azonos nevű tanuló egy osztályban.
 * Az állomány tartalma soronként:
 * • iskola kezdésének éve (2004-2007)
 * • az osztály betűjele (a-e)
 * • a diák neve (ékezetek nélkül).
Az adattagok pontosvesszővel vannak elválasztva.
Példa (részlet) a nevek.txt állományra:
2004;d;Vavrek Kristof
2006;e;Hidas Reka
2006;d;Kun Michael */
namespace Iskola
{
    class JelszóGeneráló
    {
        private Random Rnd;

        public JelszóGeneráló(Random r)
        {
            Rnd = r;
        }

        public string Jelszó(int jelszóHossz)
        {
```

```

string jelszó = "";
while (jelszó.Length < jelszóHossz)
{
    char c = (char)Rnd.Next(48, 123);
    if ((c >= '0' && c <= '9') || (c >= 'a' && c <= 'z'))
    {
        jelszó += c;
    }
}
return jelszó;
}
}
class Program
{
    struct diakok
    {
        public int ev;
        public char osztaly;
        public string nev;
    }
    static diakok[] adatok = new diakok[1000];
    static string azonosito(int evszam,char oszt,string neve)
    {
        char evfolyamutolso = evszam.ToString()[3];
        string vnev3 = neve.Substring(0, 3).ToLower();
        string knev3 = neve.Split()[1].Substring(0, 3).ToLower();
        return Convert.ToString(evfolyamutolso)+ Convert.ToString(oszt) +vnev3+knev3;
    }
    static void Main(string[] args)
    {
        /*2. Olvassa be és tárolja el egy célszerűen megválasztott adatszerkezetben
        * (pl.: vektor, lista) a nevek.txt állományban lévő adatokat!
        * 3. Írja ki a képernyőre, hogy hány tanuló jár az iskolába! */

        string[] fajlbol = File.ReadAllLines("nevek.txt");

        int sorokszama = 0;//sorok száma a fájlban
        int i;//ciklusváltozó
        for (int k = 0; k < fajlbol.Count(); k++)
        {
            string[] egysordarabolva = fajlbol[k].Split(';');
            adatok[sorokszama].ev = Convert.ToInt32(egysordarabolva[0]);
            adatok[sorokszama].osztaly = Convert.ToChar(egysordarabolva[1]);
            adatok[sorokszama].nev = egysordarabolva[2];
            sorokszama++;
        }

        //Console.WriteLine("A tanulók listája fájlból");
        int tanuloyszama = sorokszama;
        /* Console.WriteLine("ev          osztaly          nev");//adatok kiírása
        táblázatosan
        for (i = 0; i < tanuloyszama; i++)
        {
            Console.WriteLine("{0,-10} {1,-5} {2}", adatok[i].ev, adatok[i].osztaly,
        adatok[i].nev);
        }*/
        Console.WriteLine(" 3. feladat:  \n\tAz iskolába {0} tanuló jár.", tanuloyszama);

        /* Írja ki a képernyőre a minta szerint, hogy ki rendelkezik a leghosszabb névvel!
        * A szóközöket ne számolja a név hosszához! Holtverseny esetén minden, leghosszabb
        névvel rendelkező tanuló neve jelenjen meg!
        * Jelenítse meg a kiírásban a leghosszabb név/nevek hosszát is a minta szerint! */
        int max = adatok[0].nev.Count();
        int maxi = 0, szokozdb = 0;

        for (i = 0; i < tanuloyszama; i++)
        {
            szokozdb = 0;

```

```

        if (adatok[i].nev.Count() > max)
        {
            foreach (var d in adatok[i].nev)
            {
                if (d == ' ') szokozdb++;
            }
            max = adatok[i].nev.Count();
            max = max - szokozdb;
        }
    }
    Console.WriteLine("4. feladat: A leghosszabb ({0} karakter) nevű tanuló(k):",max);
    maxi = max;
    for (i = 0; i < tanuloyszama; i++)
    {
        szokozdb = 0;
        if (adatok[i].nev.Count() > max)
        {
            foreach (var d in adatok[i].nev)
            {
                if (d == ' ') szokozdb++;
            }
            max = adatok[i].nev.Count();
            max = max - szokozdb;
            if (max == maxi)
            {
                Console.WriteLine("\t"+adatok[i].nev);
            }
        }
    }
    /*Az iskolai rendszergazdának egyedi azonosítókat kell készítenie a számítógép-hálózat
    használatához.
    * Az azonosítókat a következő módon alakítja ki: első karaktere az évfolyam utolsó
    számjegye (pl.: 2006 esetén 6),
    * következő karakter az osztály betűjele, majd a vezetéknév első három karaktere,
    végül első keresztnév első három karaktere következik.
    * Az azonosítóban mindenütt kisbetűk szerepelnek. Feltételezhetjük, hogy a vezetéknév
    és az első keresztnév legalább 3 karakteres.
    * Készítsen jellemzőt vagy függvényt, melyben meghatározza a rendelkezésre álló
    adatokból a tanuló azonosítóját! Az azonosítókat ne tárolja!
    * Az elkészített jellemzőt/függvényt felhasználva írja ki az adatszerkezetben tárolt
    első és utolsó tanuló azonosítóját a minta szerint! */
    Console.WriteLine("5. feladat: Azonosítók");
    Console.WriteLine("\tElső: {0} - {1}", adatok[0].nev, azonosito(adatok[0].ev,
    adatok[0].osztaly, adatok[0].nev));
    int utolso = tanuloyszama - 1;
    Console.WriteLine("\tUtolsó: {0} - {1}", adatok[utolso].nev,
    azonosito(adatok[utolso].ev, adatok[utolso].osztaly, adatok[utolso].nev));

    /*Kérjen be egy azonosítót és állapítsa meg, hogy ki tartozhat hozzá! A tanuló adatait
    írja a képernyőre!
    * Ha nem talál megfelelő diákot, akkor a „Nincs megfelelő tanuló.” mondatot jelenítse
    meg a minta szerint! */

    Console.WriteLine("6. feladat: Kérek egy azonosítót [pl.: 4dvavkri]: ");
    string inputazon = Console.ReadLine();
    bool van = false;
    i = 0;
    int sorszam = 0;
    while(i<tanuloyszama && !van)
    {
        if(inputazon== azonosito(adatok[i].ev, adatok[i].osztaly, adatok[i].nev))
        {
            van = true;
            sorszam = i;
        }
        i++;
    }

```

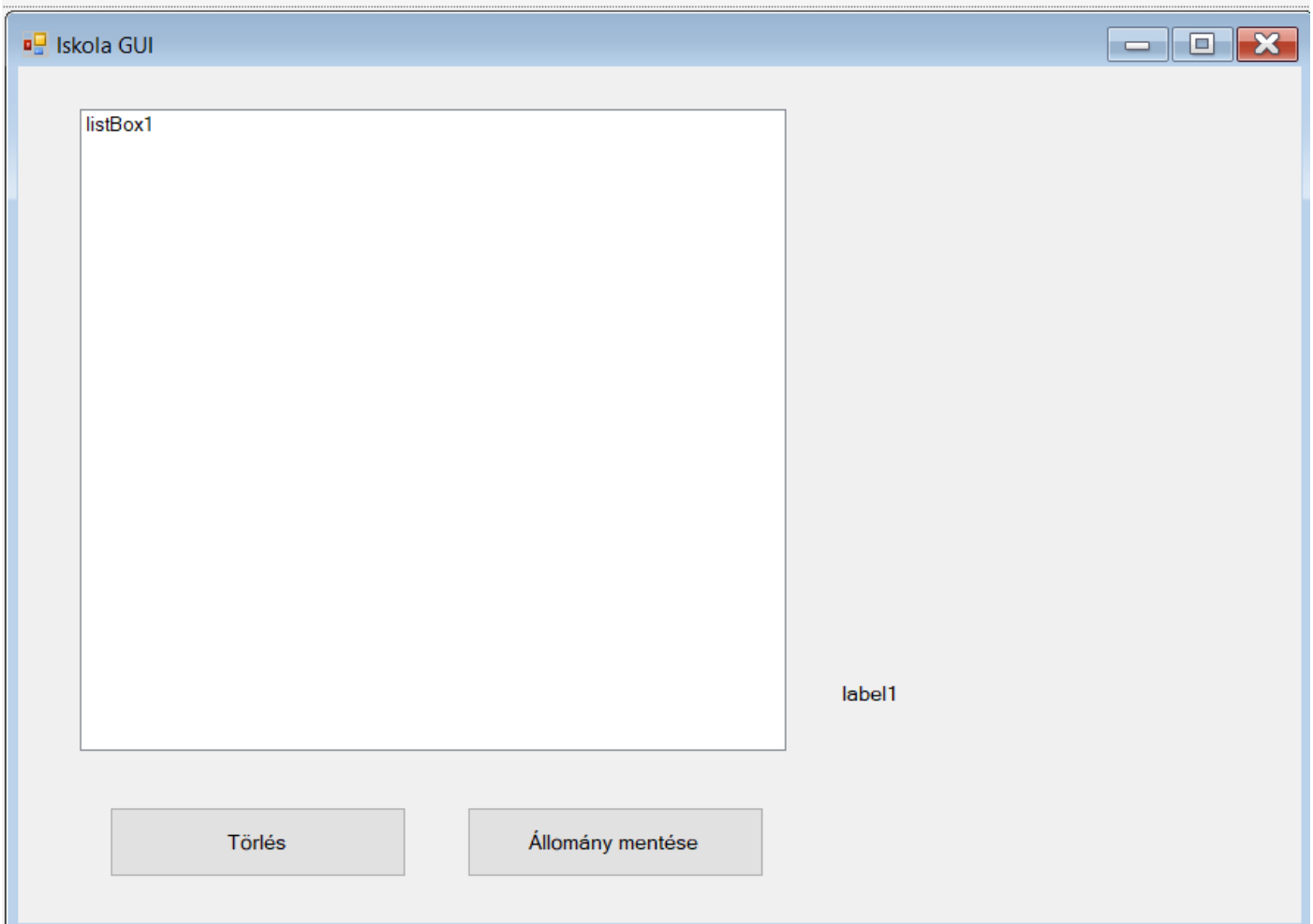
```

    }
    if (van)
    {
        Console.WriteLine("{0,-10} {1,-5} {2}", adatok[sorszam].ev, adatok[sorszam].osztaly,
        adatok[sorszam].nev);
    }
    else
    {
        Console.WriteLine("Nincs megfelelő tanuló.");
    }

    /*A forráskódjába a meglévő osztálydefiníció fölé illessze be a java.txt vagy a
    csharp.txt forrásállományból a
    * JelszóGeneráló osztályt definiáló kódrészletet, majd hozzon létre belőle egy
    osztálypéldányt (objektumot)!
    * Az osztálypéldány konstruktorát a véletlenszámok előállítására alkalmas beépített
    objektummal paraméterezze!
    * Válasszon ki véletlenszerűen egy tanulót, majd a JelszóGeneráló osztály Jelszó()
    metódusának hívásával írjon ki egy 8 karakter hosszú jelszót a minta szerint! */
    Console.WriteLine("7. feladat: Jelszó generálása");
    Random r = new Random();
    string randomTanuló = adatok[r.Next(0, tanuloyszama)].nev;
    JelszóGeneráló jg = new JelszóGeneráló(r);
    Console.WriteLine("\t {0} - {1}",randomTanuló, jg.Jelszó(8));
    Console.ReadKey();
}
}
}
}
}

```

Megoldás- grafikus:



```
using System;
```

```

using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
/*Készítsen grafikus alkalmazást a következő feladatok megoldására,
 * melynek projektjét IskolaGUI néven mentse el!
 * 9. Az alkalmazás grafikus felhasználói felületét alakítsa ki a minta szerint!
 * Az ablak címsorában a „Iskola GUI” szöveget jelenítse meg!
 * 10. Az ablakon található listába a program induláskor
 * töltse be a nevekGUI.txt állomány sorait!
 * A lista elemei a forrásfájl egy-egy sora (továbbiakban tanulója) legyen! */
namespace IskolaGUI
{
    public partial class Form1 : Form
    {
        string[] fajlbol = File.ReadAllLines("nevekGUI.txt");

        Random n = new Random();
        int sorokszama = 0; //sorok száma a fájlban
        int i, j; //ciklusváltozó
        private void adatokbetoltese()
        {
            sorokszama = 0;
            for (int k = 0; k < fajlbol.Count(); k++)
            {
                listBox1.Items.Add(fajlbol[k]);
            }
            elemekszamalabel.Text = "A listában az elemek száma: " +
listBox1.Items.Count.ToString(); //lista elemek száma
        }
        public Form1()
        {
            InitializeComponent();
            adatokbetoltese();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            /*Ha az „Állomány mentése” parancsgombra kattintunk,
            * akkor történjen meg a listából a tanulók mentése a nevekNEW.txt állományba,
            * melynek szerkezete a forrásállomány szerinti legyen!
            * Ha a mentés sikeres volt, akkor a „Sikeres mentés!” felirat jelenjen meg egy felugró
            * ablakban! */

            FileStream fnev = new FileStream("nevekNEW.txt", FileMode.Create);
            StreamWriter fajlbairo = new StreamWriter(fnev);
            foreach (var o in listBox1.Items)
                fajlbairo.WriteLine(o);

            fajlbairo.Close();
            fnev.Close();
            MessageBox.Show("Sikeres mentés!");
        }

        private void button1_Click(object sender, EventArgs e)
        {
            /*. Oldja meg, hogy a kijelölt tanuló a „Törlés” parancsgomb lenyomása után
            * törlésre kerüljön a listából!
            * Ha a listában nincs kijelölt tanuló, akkor törléskor a „Nem jelölt ki tanulót!”
            * szöveg jelenjen meg egy felugró ablakban! */
            int i = 0, gyujto = 0;
            while (i <= listBox1.Items.Count - 1)

```

```
{
    if (listBox1.SelectedIndex == i)
    {
        listBox1.Items.RemoveAt(listBox1.SelectedIndex);
        gyujto = 1;
        i--;
    }
    i++;
}
if (gyujto == 0)
{
    MessageBox.Show("Válasszon legalább egy elemet");
}
elemekszamalabel.Text = "A listában az elemek száma: " +
listBox1.Items.Count.ToString();//lista elemek száma
}
}
}
```