

8.3 Példa: Véletlen számok generálása lottószámokhoz, majd kiírásuk fájlba és visszaolvasásuk fájlból (szöveg, bináris)

Példánkban szerepeljen három rádiógomb (RadioButton) egy konténerben (GroupBox), amelyben a felhasználó kiválaszthatja, hogy ötös, hatos vagy skandináv lottóhoz kíván véletlen számokat generálni. Az első esetben 5 számot generálunk 90-ből, a másodikban 6-ot 45-ből, míg a harmadikban 7-et 35-ből. A generálást a „Generál” feliratú gombra kattintással indítjuk, melyeket egy „ListBox” komponensben jelenítünk meg. A felhasználónak lehetősége van szöveges és bináris fájlba kiírni illetve onnan visszaolvasni az adatokat igény szerint. Mivel a fájlban lévő adatok száma nem ismert, hiszen nem tudhatjuk, hogy utoljára hány számot írtunk ki az állományba, így a számok beolvasását előtesztelő ciklussal oldjuk meg. Kiíraskor két lehetőségünk is van egyrészt a listában lévő számok darabszámát kell meghatározni és ezt követően már előírt lépésszámú ciklussal is kiírhatjuk a számokat állományba, de itt is alkalmazhatjuk a már jól bevált előtesztelő ciklust.

A „Kilépés” feliratú gombra kattintva az alkalmazásunk fejezze be a program futtatását! Egy üzenet ablakban kérdezzünk vissza, hogy valóban be szeretné-e zárni az alkalmazást a felhasználó és csak „igen” választása esetén lépünk ki a programból.

1. Hozzunk létre Windows alapú új projektet „Véletlenfajlba” néven !
2. Tervezzük meg az űrlapot az alábbiak szerint:

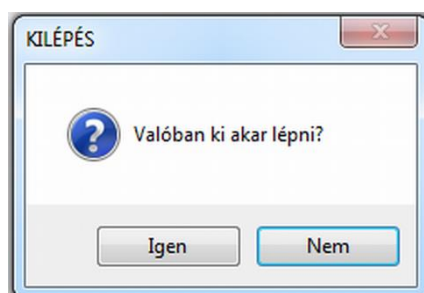
The screenshot shows a Windows application window titled "Lottószámok". Inside the window, there is a yellow rectangular area with the title "Lottószámok". Within this yellow area, there are three radio buttons: "Ötös lottó" (which is selected), "Hatos lottó", and "Skandináv lottó". Below the yellow area is a large, empty white rectangular box. To the right of the yellow area, there are five buttons stacked vertically: "Generál", "Textfájlba ír", "Textfájlból olvas", "Bináris fájlba ír", and "Bináris fájlból olvas". At the bottom right of the window, there is a button labeled "Kilépés".

3. Az űrlapon szereplő komponensek tulajdonságait állítsuk be az alábbiak szerint:

Komponens	Text	Name
Form	Lottószámok	frmLotto
GroupBox	Lottószámok	grpLotto
RadioButton	Ötös lottó	rgOtos
RadioButton	Hatos lottó	rgHatos
RadioButton	Skandináv lottó	rgHetes
ListBox	-----	lstSzamok
Button	Generál	btnGeneral
Button	Textfájlba ír	btnFajlbair
Button	Textfájlból olvas	btnFajbololvas
Button	Bináris fájlba ír	btnBinarisbair
Button	Bináris fájlból olvas	btnBinarisbololvas
Button	Kilépés	btnKilepes
openFileDialog	----	openFileDialog1
saveFileDialog	----	saveFileDialog1

A feladat kódja tartalmaz egy eljárásszintű metódust, amellyel a különböző típusú lottószámokból tudunk véletlenszerűen generálni megadott darabszámot. Két bemenő paramétere van a tartomány felsőhatára, amelyből húzhatunk, valamint a kihúzható számok darabszáma (mindkettő egész típusú), `void lottoszamok(int szam, int darab)`.

A „Kilépés” feliratú gomb kódját a `private void btnKilepes_Click(object sender, EventArgs e)` eljárás valósítja meg, ahol a „DialogResult” osztály által visszaadható értékek közül a „Yes”-t vizsgáljuk, mert ez esetben választotta a felhasználó az „igen” gombot. A rafináltabb felhasználók megpróbálkozhatnak az űrlap jobb felső sarkában



lévő „X” gomb vagy az F4 funkcióbillentyű megnyomásával, hogy kikerüljék a fenti eseményt. Ezért került beépítésre a következő eljárás: `protected override void OnClosing(CancelEventArgs e)`, amely az ablak bezárásakor kerül meghívásra, így mindkét említett eseményt egyszerre levédi a program. Itt fontos az `e.Cancel = true;` parancs, mert egyéb esetben ugyanazt csinálja a program mindkét gomb lenyomására.

A fájlba írásnál és olvasásnál is függetlenül attól, hogy szöveges vagy bináris állománnyal dolgozunk beállítjuk a Fájldialogus ablak címét, a kezdeti könyvtárat, hogy alapértelmezés szerint honnan illetve hova olvasson vagy írjon illetve beállítunk egy fájlshűrőt, hogy az ablak csak az adott kiterjesztésű állományokat mutassa (*.txt vagy *.* illetve (*.dat vagy *.*)). A *.* azt jelenti, hogy az összes fájlt mutassa függetlenül a nevétől és kiterjesztésétől. Ezzel a három beállítással sok kellemetlen plusz lépéstől tudjuk megóvni a felhasználót, azaz felhasználóbaráttá tesszük a programunkat. A „ShowDialog()” metódus szolgál a fájldialogus ablakok megnyitására. Mindenesetben a fájlműveletek előtt ellenőrizni kell, hogy a felhasználó adott-e meg fájlnevet illetve, hogy az OK gombot nyomta-e meg. Mégsem esetén ugyanis nem kell fájlműveletet végezni, hiszen a felhasználó közben meggondolta magát, míg fájlnev nélkül az operációs rendszer nem engedi meg fájl létrehozását! Ezt követően már természetesen különböznek a feltételek belső utasításai aszerint, hogy írunk vagy olvasunk és aszerint is, hogy szöveg vagy bináris állománnyal van dolgunk.

Szövegfájlba íráskor használjuk a `TextWriter` osztályt:

```
TextWriter tw = File.CreateText(saveFileDialog1.FileName)
```

Szövegfájlból olvasáskor pedig használjuk a `TextReader` osztályt:

```
TextReader tr = File.OpenText(openFileDialog1.FileName)
```

Itt fontos megemlíteni, hogy alából ezeket az osztályokat a C# nem ismeri fel, ezért a kód elején vegyük fel hiatozasként a using-ok közé a `using System.IO`-t.

Szövegfájlból beolvasásnál addig olvasunk, amíg a beolvasott érték már nem null érték:

```
while ((sor = tr.ReadLine()) != null).
```

Binráis állományból való beolvasáskor viszont más lesz a leállási feltétel:

```
while (br.PeekChar() != -1).
```

Figyelem, minden visszaolvasás előtt a listát ürítsük ki a „Clear()” metódussal, mert egyébként a korábbi beolvasások eredményeit is benne hagyja, ami nagyon zavaró lehet a felhasználó számára!

Minden állomány kezelésnél kulcskérdés az állományok lezárása, ez szövegfájlnál mindeképpen kötelező, binárisnál ajánlott, addig ugyanis üresnek mutatja az operációs rendszer a fájl méretét. Ezt a „Close()” metódussal tehetjük meg.

4. A feladat példakódja: frmLotto.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.IO;
using System.Windows.Forms;
namespace Fajlkezeles
{
    public partial class frmLotto : Form
    {
        public frmLotto()
        {
            InitializeComponent();
        }
        private void btnKilepes_Click(object sender, EventArgs e)
        {
            DialogResult valasz;
            valasz = MessageBox.Show("Valóban ki akar lépni?", "KILÉPÉS",
                MessageBoxButtons.YesNo,
                MessageBoxIcon.Question, MessageBoxDefaultButton.Button2);
            if (valasz == DialogResult.Yes)
                Application.Exit();
        }
        protected override void OnClosing(CancelEventArgs e)
        {
            e.Cancel = true;
            base.OnClosing(e);
            btnKilepes_Click(btnKilepes, e);
        }
        void lottoszamok(int szam, int darab)
        {
            int veletlen;
            Random rnd = new Random();
            int i = 1;
            while (i <= darab)
            {
                veletlen = rnd.Next(1, szam + 1);
                if (!lstLista.Items.Contains(veletlen))
                {
                    lstLista.Items.Add(veletlen);
                    i++;
                }
            }
        }
        private void btnGeneral_Click(object sender, EventArgs e)
        {
            lstLista.Items.Clear();
            if (rgOtos.Checked)
                lottoszamok(90, 5);
            if (rgHatos.Checked)
                lottoszamok(45, 6);
            if (rgHetes.Checked)
```

```

        lottoszamok(35, 7);
    }
    private void btnFajlbair_Click(object sender, EventArgs e)
    {
        saveFileDialog1.Title = "Lottószámok mentése";
        saveFileDialog1.InitialDirectory = "c:\\";
        saveFileDialog1.Filter = "Szövegfájlok (*.txt)|*.txt|Minden  
fájl(*.*)|*.*";
        if (saveFileDialog1.ShowDialog() == DialogResult.OK &&
            saveFileDialog1.FileName != "")
        {
            int i = 0;
            TextWriter tw = File.CreateText(saveFileDialog1.FileName);
            while (i < lstLista.Items.Count)
            {
                lstLista.SelectedIndex = i;
                tw.WriteLine(lstLista.SelectedItem);
                i++;
            }
            tw.Close();
        }
    }
    private void btnFajbololvas_Click(object sender, EventArgs e)
    {
        lstLista.Items.Clear();
        openFileDialog1.Title = "Lottószámok beolvasása";
        openFileDialog1.InitialDirectory = "c:\\";
        openFileDialog1.Filter = "Szövegfájlok (*.txt)|*.txt|Minden  
fájl(*.*)|*.*";
        if (openFileDialog1.ShowDialog() == DialogResult.OK &&
            openFileDialog1.FileName != "")
        {
            TextReader tr = File.OpenText(openFileDialog1.FileName);
            string sor = "";
            while ((sor = tr.ReadLine()) != null)
            {
                lstLista.Items.Add(sor);
            }
            tr.Close();
        }
    }
    private void btnBinarisbair_Click(object sender, EventArgs e)
    {
        saveFileDialog1.Title = "Lottószámok mentése";
        saveFileDialog1.InitialDirectory = "c:\\";
        saveFileDialog1.Filter = "Szövegfájlok (*.dat)|*.dat|Minden  
fájl(*.*)|*.*";
        if (saveFileDialog1.ShowDialog() == DialogResult.OK &&
            saveFileDialog1.FileName != "")
        {
            int i = 0;
            FileStream fs = new FileStream(saveFileDialog1.FileName,
                FileMode.Create);
            BinaryWriter bw = new BinaryWriter(fs);
            while (i < lstLista.Items.Count)
            {
                lstLista.SelectedIndex = i;
                bw.Write((int)lstLista.SelectedItem);
                i++;
            }
            bw.Close();
        }
    }
    private void btnBinarisbololvas_Click(object sender, EventArgs e)

```

```

    {
        lstLista.Items.Clear();
        openFileDialog1.Title = "Lottószámok beolvasása";
        openFileDialog1.InitialDirectory = "c:\\";
        openFileDialog1.Filter = "Szövegfájlok (*.dat)|*.dat|Minden  
fájl(*.*)|*.*";
        if (openFileDialog1.ShowDialog() == DialogResult.OK &&  
            openFileDialog1.FileName != "")
        {
            FileStream fs = new FileStream(openFileDialog1.FileName,  
                                           FileMode.Open, FileAccess.Read);
            BinaryReader br = new BinaryReader(fs);
            while (br.PeekChar() != -1)
                lstLista.Items.Add(br.ReadInt32());
            br.Close();
        }
    }
}

```