**KONZOL**

1. Create New Project -> Class library (VikingLottoAdatbazis)

-> jobb klikk solutionre, Add -> New project -> Console Application (VikingAdatImport)

-> jobb klikk solutionre, Add -> New project -> Windows Forms App (VikingSzimulacio)

1. Jobb klikk VikingAdatImport -> Add -> Project Reference… -> VikingLottoAdatbazist kipipáljuk és leokézzuk.
2. Jobb klikk VikingSzimulacio -> Add -> Project Reference… -> VikingLottoAdatbazist kipipáljuk és leokézzuk.
3. VikingLottoAdatbazisban található „class1.cs” fájlt megnyitjuk.
4. Tools -> NuGet Package Manager -> Manage NuGet Packages for Solution… -> Browse és itt kikeressük és installáljuk a következőket:

EntityFrameworkCore(5.0.16), EntityFrameworkCore.SqlServer(5.0.16) és EntityFrameworkCore.Tools(5.0.16).

1. „class1.cs” átnevezése (Rename) tetszőlegesen, ez esetben: LottoSzam.cs-re.
2. public class LottoSzam

{

public int Id { get; set; }

public int Szam1 { get; set; }

…

public int Szam6 { get; set; }

}

1. Szam 6 alá új sorban Alt + Enter -> Generate constructor… -> összesből pipa ki (Deselect all) és leokézzuk.
2. Üres constructur alá új sorban megint Alt + Enter -> Generate constructor… -> csak Id-ból szedjük ki a pipát és ok.
3. A kapott constructurunkat átalakítjuk erre:

public LottoSzam(string sor)

{

string[] tombSzoveg = sor.Trim().Split(";");

if(tombSzoveg.Length == 6)

{

int[] tombSzam = new int[6];

for (int i = 0; i < 6; i++)

{

try

{

tombSzam[i] = Convert.ToInt32(tombSzoveg[i].Trim());

}

catch (Exception)

{

throw new Exception();

}

if((tombSzam[i] < 1) || (tombSzam[i] > 48)) throw new Exception();

}

HashSet<int> halmaz = new HashSet<int>(tombSzam);

if(halmaz.Count() != 6) throw new Exception();

Szam1 = tombSzam[0];

Szam2 = tombSzam[1];

Szam3 = tombSzam[2];

Szam4 = tombSzam[3];

Szam5 = tombSzam[4];

Szam6 = tombSzam[5];

}

else

{

throw new Exception();

}

}

* jobb klikk VikingLottoAdatbazisra -> Add -> Class -> tetszőleges név, itt: VikingContext
* Ezt: class VikingContex {}

Átírjuk erre :

public class VikingContext:DbContext

{

public DbSet<LottoSzam> LottoSzamok { get; set; }

}

* VikingContext:DbContext-ben maradva, a public DbSet<LottoSzam> LottoSzamok { get; set; } alá új sorba beírjuk, hogy „override on” majd tab.
* Itt a „base.OnConfiguring(optionsBuilder);”-t átírjuk erre:

optionsBuilder.UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=Viking0503;Trusted\_Connection=true");

(Database=Viking0503 <- tetszőleges adatbázis név)

* Migration-Update
* VikingAdatImporton belül található „Program.cs”-t megnyitjuk és átírjuk a „Console.WriteLine("Hello World!");”-öt a következőre:

VikingContext db = new VikingContext();

if(!db.LottoSzamok.Any())

{

string[] sorok = null;

bool sikeresFileNyitas = true;

try

{

sorok = File.ReadAllLines(args[0]);

}

catch (Exception)

{

sikeresFileNyitas = false;

Console.WriteLine("A megadott fájlnév nem megfelelő! Pl: filename.csv");

}

if(sikeresFileNyitas)

{

List<LottoSzam> szamok = new List<LottoSzam>();

foreach(var sor in sorok)

{

try

{

szamok.Add(new LottoSzam(sor));

}

catch (Exception)

{}

}

db.LottoSzamok.AddRange(szamok);

db.SaveChanges();

Console.WriteLine("Sorok száma: "+db.LottoSzamok.Count()+" db");

}

}

else

{

Console.WriteLine("Az adattábla már létezik!");

}

!! csv fájl beolvasásával, ha probléma van akkor ***1:12:20 a videón***!!

Én esetemben: VikingAdatImport.exe C:\Users\User\OneDrive\Desktop\aspnet+js\vl.csv

**vagy** jobb klikk VikingAdatImport -> Properties -> Debug -> Application Arguments (ide a file elérési útvonala).

Értelemszerűen itt meg kell adni a vl.csv fájl útvonalát.

És a végén buildelünk egyet!

**WINFORM**

* VikingSzimulacio -> Form1.cs[Design] -> 2 gomb berakni (bal oldalt -> Toolbox - > all windows forms - > Button)
* Jobb oldal -> Properties -> ABC sorrend ->  
  2 gombot átnevezni
  + (Name): btnSorsol , btnBezar
  + Text: Sorsolás, Bezár
* Toolbox - > checkBoxot berakni -> 2x katt a checkboxon -> létre jön :

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{}

* Checkbox törlése a Form1.cs[Design] résznél ->
* Form1.cs[Design] üres fehér részén valahol kétszer kattintunk és a következőre alakítjuk a kapott függvényt:

private void Form1\_Load(object sender, EventArgs e)

{

btnSorsol.Enabled = false;

for (int i = 0; i < 6; i++)

for (int j = 0; j < 8; j++)

{

CheckBox box = new CheckBox();

box.AutoSize = true;

box.Location = new Point(j \* 50 + 50, i \* 50 + 50);

box.Text = (i \* 8 + j + 1).ToString();

box.CheckedChanged += checkBox1\_CheckedChanged;

this.Controls.Add(box);

}

}

* Form1.cs[Design] -> „Bezár” gombra kétszer katt és erre alakítjuk:

private void btnBezar\_Click(object sender, EventArgs e)

{

this.Close();

}

* Form1.cs -> public Form1() **felett** egy listát hozunk létre:

private List<int> tipp = new List<int>();

* picit a lejjebb, private void checkBox1\_CheckedChanged(object sender, EventArgs e) a következőre alakítjuk:

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

CheckBox box = (CheckBox)sender;

if(box.Checked)

{

tipp.Add(Convert.ToInt32(box.Text));

if(tipp.Count() == 6)

{

btnSorsol.Enabled = true;

foreach (var item in this.Controls)

{

if(item.ToString().Contains("CheckBox"))

{

if (!((CheckBox)item).Checked)

((CheckBox)item).Enabled = false;

}

}

}

}

else

{

tipp.Remove(Convert.ToInt32(box.Text));

if (tipp.Count() == 5)

{

btnSorsol.Enabled = false;

foreach (var item in this.Controls)

{

if (item.ToString().Contains("CheckBox"))

{

((CheckBox)item).Enabled = true;

}

}

}

}

}

* Form1.cs[Design] -> Sorsolás gombra 2x katt.
* Feljebb található privát tipp listánk (private List<int> tipp = new List<int>();) alatt egy új db-t hozunk létre:

private VikingContext db = new VikingContext();

* private void btnSorsol\_Click(object sender, EventArgs e) -> random létrehozása:

private void btnSorsol\_Click(object sender, EventArgs e)

{

Random vel = new Random();

HashSet<int> halmaz = new HashSet<int>();

do

{

halmaz.Add(vel.Next(1, 49));

} while (halmaz.Count() != 6);

db.LottoSzamok.Add(new LottoSzam(string.Join(";", halmaz)));

db.SaveChanges();

MessageBox.Show(halmaz.Intersect(tipp).Count().ToString() + " találatod lett!");

}