Entity Framework alapú adatbáziselérés 2

Dr. Johanyák Zsolt Csaba http://johanyak.hu

A gyakorlat célja az, hogy a korábban létrehozott Telefonszám kezelő alkalmazást kiegészítsük egy WPF típusú felülettel.

1. Projekt és alapbeállítások

Töltsük le és nyissuk meg a kiinduló megoldást (Solution-t), ami a korábban létrehozott adathozzáférési réteg (Telefonszamok_DAL) és konzol alkalmazás (Telefonszamok_Konzol) projekteket tartalmazza. Hozzunk létre egy új WPF projektet (Telefonszamok_WPF) a megoldáson belül.

	Solu	tion Explorer	▼ ₽ ×
	C	o 🕼 'o - 2 Q 🗇 🖹 🖌 🗕	
	Sea	rch Solution Explorer (Ctrl+é)	<u>م</u>
 Build Solution Rebuild Solution Clean Solution Run Code Analysis on Solution Batch Build Configuration Manager Manage NuGet Packages for Solution Enable NuGet Package Restore New Solution Explorer View Show on Code Map Calculate Code Metrics Project Dependencies	Ctrl+Shift+B Alt+F11	Solution 'Telefonszamok' (2 projects) Telefonszamok_DAL Properties References App.Config demTelefonszamok.edmx cedmTelefonszamok.edmx.sql packages.config Telefonszamok_Konzol Properties References App.config packages.config Program.cs	
Add	•	New Project	
		New Project	

	Add New Project		? ×
▷ Recent	.NET Framework 4.5 Sort by: Default		Search Installed Templates (Ctrl+E)
▲ Installed	Blank App (Universal Apps)	Visual C#	Type: Visual C#
 ▶ Visual Basic ✓ Visual C# ▶ Store Apps 	Windows Forms Application	Visual C#	Windows Presentation Foundation client application
Windows Desktop	WPF Application	Visual C#	
 Office/SharePoint Cloud 	Console Application	Visual C#	
LightSwitch Reporting	Hub App (Universal Apps)	Visual C#	
Silverlight Test	ASP.NET Web Application	Visual C#	
WCF Workflow	Class Library	Visual C#	
▷ Visual C++ ▷ Visual F#	Class Library (Portable)	Visual C#	
SQL Server ▷ JavaScript	₩ebView App (Windows Phone)	Visual C#	
Python ▷ TypeScript	Silverlight Application	Visual C#	
 Other Project Types Modeling Projects 	Silverlight Class Library	Visual C#	
▷ Online	Class Library (Portable for Universal Apps)	Visual C#	•
	Click here to go online and find templates.		
Name: Telefonszamok_	WPF		
Location: E:\Dropbox\VP\S	egedletek\Sajat\Adatbaziskezeles\EF_Telefonszamok_WPF	•	Browse
			OK Cancel

Állítsuk be indító projektté az újonnan létrehozott projektet.

	-	Solution Explorer 🔹 म 🗙
	· · · · · · · · · · · · · · · · · · ·	
		Search Solution Explorer (Ctrl+é)
		 Solution 'Telefonszamok' (3 projects) C# Telefonszamok_DAL Properties I= References App.Config
		edmTelefonszamok.edmx
		edmTelefonszamok.edmx.sql
	Build	ges.config
	Rebuild	ties
	Clean	inces
	View	▶ onfig
	Analyze	 ges.config
6	Publish	im.cs
	Scope to This	amok_WPF
	New Colution Fundamentation	rties
	New Solution Explorer View	nces
52	Show on Code Map	aml
	Build Dependencies	Vindow.xaml
	Add	•
Ħ	Manage NuGet Packages	
ø	Set as StartUp Project	

A referenciák között állítsuk be a Telefonszamok_DAL projektet annak érdekének, hogy felhasználhassuk a későbbiekben az ott létrehozott entitás modellt.



	Reference M	anager - Telefonszamok_WPF	? 🗙
Assemblies	_		Search Solution (Ctrl+E)
▲ Solution	Name	Path	Name:
Projects	Telefonszamok DAL Telefonszamok_Konzol	e:\Dropbox\VP\Segedletek\Sajat\Adatbazisk e:\Dropbox\VP\Segedletek\Sajat\Adatbazisk	Telefonszamok_DAL
▶ COM			
▷ Browse			
	•	•	
		Browse	OK Cancel

Állítsuk be a DAL projekt névterét felhasznált névtérként az ablak C# kódjában (MainWindow.xaml.cs).

MainWindow.xaml	MainWi	ndow.xaml.cs* 👎	• X
Œ Telefonszamok_W	'PF		
1 <u>-</u> u	sing System	5	
2 u	sing System	.Collections.	Generic;
3 u	sing System	.Linq;	
4 u	sing System	.Text;	
5 u	sing System	.Threading.Ta	asks;
6 u	sing System	.Windows;	
7 u	sing System	.Windows.Cont	trols;
8 u	sing System	.Windows.Data	а;
9 u	sing System	.Windows.Docu	uments;
10 u	sing System	.Windows.Inpu	it;
11 u	sing System	.Windows.Medi	ia;
12 u	sing System	.Windows.Medi	ia.Imaging;
13 u	sing System	.Windows.Navi	igation;
14 u	sing System	.Windows.Shap	pes;
15 U	sing Telefo	nszamok_DAL;	
16			

Átmásoljuk az App.Config állományt a konzol projektből az aktuális projektbe, majd az ablak konstruktorában az alábbi két utasítás segítségével beállítjuk, hogy a Telefonszamok_DAL projekt könyvtárát tekintse a program adatkönyvtárnak, itt keresse az adatbázisállományt.

```
var DataDir =
   Directory.GetParent(Directory.GetParent(Directory.GetParent(Directory.GetParent(Directory.GetCurrentDirectory()).FullName).FullName) +
   "\\Telefonszamok_DAL";
AppDomain.CurrentDomain.SetData("DataDirectory", DataDir);
```

Engedélyezzük a System.IO.Directory referencia felvételét a referenciák közé.

```
public MainWindow()
{
    InitializeComponent();
    var_DataDir_=
    PostaDir_=
    PostaDir_=
    PostaDir_System.IO.Directory' and all other references in file? (Alt+Enterint(
    Directory.GetCurrentDirectory()).FullName).FullName) +
    "\\Telefonszamok_DAL";
    AppDomain.CurrentDomain.SetData("DataDirectory", DataDir);
```

Létrehozunk egy adattagot az entitáskonténer számára, majd felvesszük a referenciák közé az Entity Framework-öt.



Létrehozzuk a konténer objektumot a főablak konstruktorában.

```
cnTelefonszámok=new cnTelefonszámok();
```

2. A felület elkészítése

Az alkalmazás felületét a megkövetelt funkcionalitás határozza meg. A következő funkcionalitást várjuk el.

1. Legyen képes a módosított adatok mentésére



2. A tárolt adatok legyenek lekérdezhetőek egyben egy minden információt tartalmazó táblázatban.

	Telefor	- 🗆 ×	
Fájl	Lekérdezés	Adatfelvitel/Módosítás	
	Minde		
bázisel	Helysé		

	Telefonszám nyilvántartó – 🗖 🗙							
Fájl Lekérde	ezés Ad	latfelvi	tel/Módosítás					
Vezetéknév	Utónév	Irsz	Helységnév	Lakcím	Telefonszámok			
Senki	Alfonz	6000	Kecskemét	Kis út 1.	+36-11-555-5555, +36-12-444-4444			
Gipsz	Jakab	6000	Kecskemét	Malom Köz 1.				
Erős	Áron	2038	Sóskút	Alma rét 3.	+36-11-111-1111			
Olvasó	Jolán	2039	Pusztazámor	Hermelin sugárút 5.	+36-11-333-3333			
Argon	Géza	2090	Remeteszőlős	Ordas köz 6.	+36-13-555-555, +36-13-556-555			

A helységekre vonatkozó adatok legyenek lekérdezhetőek egy táblázatban.

🗈 Telefonszám nyilv – 🗖 🗙						
Fájl Lekérdez	és /	Adatfelvitel/Módosítás				
Név	lrsz					
Kecskemét	6000)				
Sóskút	2038	3				
Pusztazámor	2039	9				
Budaörs	2040)				
Törökbálint 20		5				
Remeteszőlős	2090	ו				

3. A helységadatok legyenek módosíthatóak

	Telefonszám i	nyilvántartó –	
Fájl Lekérdezés	Adatfelvitel/Módosí	tás	
Név Ir	Helységek		
	Telefonszám	nyilvántartó	- 🗆 ×
Fájl Lekérdezés	Adatfelvitel/Módosí	tás	
Keresés irányítószá	m alapján	6000	v
Keresés helységné	v alapján	Kecskemét	~
lrányítószám		6000	
Helységnév		Kecskemét	
Módosíto	tt adatpár rögzítése	Új helység	Vissza

A fenti igények megvalósításához StackPanel rétegmenedzsert használunk, amire egy menüt egy DataGrid és egy Grid komponenst fogunk elhelyezni. A Grid segítségével alakítjuk ki a helységadatok megjelenítéséhez szükséges űrlapot. Kezdetben sem a DataGrid sem a Grid nem látható. Az ablak XAML kódja az alábbi.

```
<MenuItem x:Name="miHelységek" Header="Helységek"
           Click="miHelységek_Click" />
     </MenuItem>
     <MenuItem x:Name="miAdatfelvitelMódosítás" Header="Adatfelvitel/Módosítás">
        <MenuItem x:Name="miHelységekAM" Header="Helységek"
           Click="miHelységekAM Click"/>
     </MenuItem>
  </Menu>
  <!-- Adatrács -->
  <DataGrid x:Name="dgAdatrács" ItemsSource="{Binding}" Visibility="Hidden"/>
  <!-- Helységadatok megjelenítése és módosítása -->
  <Grid x:Name="grHelység" Visibility="Hidden" Margin="0,10,0,0"</pre>
     DataContext="{Binding}">
     <Grid.RowDefinitions>
        <RowDefinition/>
        <RowDefinition/>
        <RowDefinition/>
        <RowDefinition/>
        <RowDefinition/>
     </Grid.RowDefinitions>
     <Grid.ColumnDefinitions>
        <ColumnDefinition/>
        <ColumnDefinition/>
     </Grid.ColumnDefinitions>
     <Label Content="Keresés irányítószám alapján" Grid.Row="0" Grid.Column="0"/>
     <ComboBox x:Name="cblrsz" ItemsSource="{Binding}" DisplayMemberPath="lrsz"
                   Grid.Row="0" Grid.Column="1" Margin="5"
                   SelectionChanged="cbIrsz_SelectionChanged"
                   IsSynchronizedWithCurrentItem="True"/>
     <Label Content="Keresés helységnév alapján" Grid.Row="1" Grid.Column="0" />
     <ComboBox x:Name="cbHelységnév" ItemsSource="{Binding}"
                   DisplayMemberPath="Név"
                   Grid.Row="1" Grid.Column="1" Margin="5"
                   SelectionChanged="cbIrsz SelectionChanged"
                   IsSynchronizedWithCurrentItem="True"/>
     <Label Content="Irányítószám" Grid.Row="2" Grid.Column="0"/>
     <TextBox x:Name="tblrsz" Grid.Row="2" Grid.Column="1" Margin="5"/>
     <Label Content="Helységnév" Grid.Row="3" Grid.Column="0"/>
     <TextBox x:Name="tbHelységnév" Grid.Row="3" Grid.Column="1" Margin="5"/>
     <StackPanel Grid.Row="4" Grid.Column="0" Grid.ColumnSpan="2"</pre>
                         Orientation="Horizontal" HorizontalAlignment="Center">
        <Button x:Name="btRögzít" Content="Módosított adatpár rögzítése"
                   Click="btRögzit Click" Margin="50,50,10,10"/>
        <Button x:Name="btÚjHelység" Content="Új helység"
Margin="50,50,10,10" Click="btÚjHelység_Click"/>
        <Button x:Name="btVissza" Content="Vissza"
                   Click="btVissza_Click" Margin="50,50,10,10" />
     </StackPanel>
  </Grid>
</StackPanel>
```

```
</Window>
```

Minden olyan menüponthoz, amelyből nem nyílik almenü, egy eseménykezelőt készítünk a Properties ablak események (Events) funkciójával.

A Mentés menüpont eseménykezelőjében elmentjük az adatbázisba az eddig végrehajtott módosításokat.

```
private void miMentés_Click(object sender, RoutedEventArgs e)
{
```

cnTelefonszámok.SaveChanges();

A Kilépés menüpont eseménykezelőjében kilépünk a programból.

```
private void miKilépés_Click(object sender, RoutedEventArgs e)
{
          Application.Current.Shutdown();
}
```

3. Egyszerű lekérdezés

}

A helységadatok megjelenítésénél egy táblázatot szeretnénk látni, amelyben a helységek nevei és irányítószámaik láthatóak. Az eseménykezelőben először láthatóvá tesszük az adatrácsot, majd végrehajtjuk a lekérdezést és végül a lekérdezés eredményét adatkötéssel kapcsoljuk a rácshoz. A lekérdezés eredményét egy listává alakítjuk, mert csak ekkor fog ténylegesen végrehajtódni a LINQ lekérdezés.

```
private void miHelységek_Click(object sender, RoutedEventArgs e)
{
    grHelység.Visibility=Visibility.Hidden;
    dgAdatrács.Visibility=Visibility.Visible;
    var er = (from x in cnTelefonszámok.enHelységek
        select new {x.Név, x.Irsz}).ToList();
    dgAdatrács.ItemsSource = er;
}

Fájl Lekérdezések
Név Irsz
```

Kecskemét	6000	
Sóskút	2038	
Pusztazámor	2039	
Budaörs	2040	
Törökbálint	2045	
Remeteszőlős	2090	

Próbáljuk ki a kódot úgy is, hogy elhagyjuk a ToList() metódushívást.

4. Komplex lekérdezés

Az összes adat megjelenítésénél táblázatos formában szeretnénk látni minden tárolt adatot. Ennek érdekében egy olyan gyűjteményt kell előállítanunk, aminek egy eleme egy személy összes adatát tartalmazza. Mivel egy személyhez több telefonszám is tartozik, ezért azt szeretnénk elérni, hogy ezek a számok egyetlen sztringben, egymástól vesszővel elválasztva jelenjenek meg. Ezt a részfeladatot úgy oldjuk meg, hogy készítünk egy bővítő metódust az enSzemély entitás osztályhoz a CustomExtensions névtérben. A bővítő metódust egy Bővítő nevű statikus osztályban helyezzük el.



A Telefonszámok bővítő metódusban megkapjuk egy enSzemély entitás referenciáját, és végigiterálunk a kapcsolódó telefonszám információkat tartalmazó entitásokon. Az utolsó kivételével mindegyik után teszünk egy vesszőt. A metódus visszaadja a telefonszámok listáját tartalmazó sztringet.

using Telefonszamok_DAL;
namespace CustomExtensions
{
 public static class Bővítő

A telefonszámlista előállításának egy másik útja az is lehetne, hogy a Telefonszamok_DAL projektben létrehozunk a személyeket leíró enSzemély entitás osztályhoz egy kiegészítést (partial class), amiben egy új tulajdonságot definiálunk Telefonszámok néven. Ehhez a Solution Explorerben egy új osztály adunk a Telefonszamok_DAL projekthez



A létrehozáskor az enSzemelyKieg nevet adjuk az új osztálynak, azonban a C# kódban ezt átírjuk.

		Add New Item - Tele	fonszamok_DAL		?	×
▲ Installed	Sort by:	Default	• # E		Search Installed Templates (Ctrl+E)	ρ-
✓ Visual C# Items Code	, •	Class	Visual C# Items	Â	Type: Visual C# Items	
Data General	•0	Interface	Visual C# Items			
Web Windows Forms	==	Windows Form	Visual C# Items			
WPF Reporting		User Control	Visual C# Items			
SQL Server Workflow	đg	Component Class	Visual C# Items			
© Online		User Control (WPF)	Visual C# Items			
	v. 1.0	About Box	Visual C# Items	-		
		<u>Click here to go online and</u>	l find templates.			
<u>N</u> ame: enSzemelyKieg.e	:5					
					<u>A</u> dd Ca	ancel

A tulajdonság csak get elérővel rendelkezik, és a lekérdezéshez tartozó kód gyakorlatilag azonos az előzőekben ismertetettel.

```
namespace Telefonszamok DAL
{
       public partial class enSzemély
       {
              public string Telefonszámok
              {
                     get
                     {
                            var s = "";
                            foreach (var x in enTelefonszámok)
                            {
                                   s = s + x.Szám;
                                   if(x!= enTelefonszámok.Last())
                                           s=s+ ", ";
                            }
                            return s;
                     }
              }
       }
}
```

Ennek a megoldásnak az az előnye, hogy a későbbiekben a Telefonszámok tulajdonságra ugyanúgy tudunk hivatkozni, mintha az az entitás egy tárolt tulajdonsága lenne.

Visszatérve a MainWindow osztály miMindenAdat_Click metódusához a következő feladatunk az, hogy előállítsuk azt a gyűjteményt, amit adatkötéssel az adatrácshoz kívánunk rendelni. Első ötletként az alábbi kódrészlet kínálkozik elegáns megoldásként. Egy LINQ lekérdezés, ami kihasználja az előzőekben alternatív megoldásként bemutatott Telefonszámok tulajdonságot.

var er = (from x in cnTelefonszámok.enSzemélyek

```
orderby x.Vezetéknév
select new
{    x.Vezetéknév, x.Utónév, x.enHelység.Irsz,
        x.enHelység.Név, x.Lakcím, x.Telefonszámok }).ToList();
```

Futtatáskor azonban azt kell tapasztalnunk, hogy az Entity Framework 6.1 változatához kapcsolódó LINQ to Entitities a fentiekben ismertetett két megoldás egyikét se támogatja LINQ lekérdezésben.

	var er = (from x in cnTelefonszámok.enSzemélyek orderby x.Vezetéknév select new { x.Vezetéknév, x.Utónév, x.enHelység	g.Ir	sz
1	1 NotSupportedException was unhandled	×	
	An unhandled exception of type 'System.NotSupportedException' occurred in EntityFramework.SqlServer.dll		év
]	Additional information: The specified type member 'Telefonszámok' is not supported in LINQ to Entities. Only initializers, entity members, and entity navigation properties are supported.		

Mivel LINQ-val nem tudjuk célunkat elérni, ezért egy foreach ciklussal haladunk végig a személy entitásokat tartalmazó gyűjteményen. LINQ nélkül az eredmény gyűjtemény előállításához először definiálnunk kell egy segédosztályt (SzemélyAdatok).

					/ C# T	elefr	nszamok_Wi	PF
			*	Build			perties	
				Rebuild			erences	
				Clean			o.config	
				View		•	vito cs	
				Analyze		•	inWindow.xa	ml
			ø	Publish			kages.config	
				Scope to This			er Team Exp	lorer Class View
				New Solution Explorer View				
		嚣	Show on Code Map			_WPF Proje	ct Properties	
				Build Dependencies		•		
	New from Template	•	1	Add		•	Level	Default
°	New Item	Ctrl+Shift+A	ă	Manage NuGet Packages			build	False
ta	Existing Item	Shift+Alt+A	ø	Set as StartUp Project				Default
*	New Folder			Debug		•	spector	Optimistic
	Deference			Serves Central				Telefonszamok_WPF.csproj
	Reference			Source Control			le Increation	
	Service Reference		۲¥.	Cut	Ctrl+X		ie inspection:	, on
	Window		a	Paste	Ctrl+V			
5	Page		X	Remove	Del			
2	User Control		Xo	Rename				
Ü	Resource Dictionary			Unload Project				
*	Class			Find Code Issues				
			· · ·				-	

		Add New Item - Telefo	onszamok_WPF		?	×
▲ Installed	Sort by:	Default	• # E		Search Installed Templates (Ctrl+E)	- م
✓ Visual C# Code	•	Class	Visual C#	1	Type: Visual C#	
Data General	••0	Interface	Visual C#		An empty class definition	
Web Windows Forms	EB	Windows Form	Visual C#			
WPF Reporting		User Control	Visual C#			
SQL Server Workflow		Component Class	Visual C#			
▶ Online		Window (WPF)	Visual C#			
		Page (WPF)	Visual C#	Ŧ		
		Click here to go online and find templates.				
<u>N</u> ame: SzemelyAdatok	cs					
					<u>A</u> dd Cance	I

```
public class SzemélyAdatok
{
    public string Vezetéknév { get; set; }
    public string Utónév { get; set; }
    public Int16 Irsz { get; set; }
    public string Helységnév { get; set; }
    public string Lakcím { get; set; }
    public string Telefonszámok { get; set; }
}
```

A MainWindow osztály miMindenAdat_Click metódusában létrehozunk egy típusos lista objektumot, amelynek elemtípusa SzemélyAdatok, majd az entitás halmazon végighaladva, minden személyhez létrehozunk egy SzemélyAdatok típusú névtelen objektumot, amit hozzáadunk a listához.

```
private void miMindenAdat_Click(object sender, RoutedEventArgs e)
{
       grHelység.Visibility=Visibility.Hidden;
      dgAdatrács.Visibility = Visibility.Visible;
      var er = new List<SzemélyAdatok>();
      foreach (var x in cnTelefonszámok.enSzemélyek)
       {
             er.Add(
                    new SzemélyAdatok()
                           Vezetéknév = x.Vezetéknév, Utónév = x.Utónév,
                    {
                           Helységnév = x.enHelység.Név, Irsz = x.enHelység.Irsz,
                           Lakcím = x.Lakcím, Telefonszámok = x.Telefonszámok()
                    });
       }
       dgAdatrács.ItemsSource = er;
}
```

A programot futtatva a Lekérdezések menü Minden adat táblázatosan menüpontját választva az alábbi ábrán látható táblázatot kapjuk.

			Telefo	onszám nyilvántai	rtó – 🗆 🗙			
Fájl Lekérde	ájl Lekérdezések							
Vezetéknév	Utónév	lrsz	Helységnév	Lakcím	Telefonszámok			
Senki	Alfonz	6000	Kecskemét	Kis út 1.	+36-11-555-5555, +36-12-444-4444			
Gipsz	Jakab	6000	Kecskemét	Malom Köz 1.				
Erős	Áron	2038	Sóskút	Alma rét 3.	+36-11-111-1111			
Olvasó	Jolán	2039	Pusztazámor	Hermelin sugárút 5.	+36-11-333-3333			
Argon	Géza	2090	Remeteszőlős	Ordas köz 6.	+36-13-555-555, +36-13-556-555			

5. Helységadatok módosítása

A helységadat adatfelvitel/módosítás során a felhasználó két funkcionalitás közül választhat. ezek az adatbázisban szereplő adatok módosítása és új adatok felvitele.

Módosítás esetén a két legördülő kombinált lista egyikével kiválasztja az aktuális települést, ezek adatai megjelennek a TextBox komponensekben, majd a módosítást követően kattint a "Módosított adatpár rögzítése" feliratú gombon.

Telefons	zám nyilvántartó 🛛 🗧	□ ×
Fájl Lekérdezés Adatfelvitel/Mo	ódosítás	
Keresés irányítószám alapján	6000	v
Keresés helységnév alapján	Kecskemét	~
lrányítószám	6000	
Helységnév	Kecskemét	
Módosított adatpár rögzít	Új helység	Vissza

A kívánt adatok megjelenítéséhez a láthatóság beállítását követően az adatok listáját adatkötéssel a Grid-hez rendeljük, majd kiválasztjuk az első helységet.

```
private void miHelységekAM_Click(object sender, RoutedEventArgs e)
{
     dgAdatrács.Visibility = Visibility.Collapsed;
     grHelység.Visibility = Visibility.Visible;
     grHelység.DataContext = cnTelefonszámok.enHelységek.ToList();
     cbIrsz.SelectedIndex = 0;
}
```

A két ComboBox együtt fog változni a "IsSynchronizedWithCurrentItem="True"" attribútum használatának köszönhetően. Ezért elegendő hozzájuk egy közös eseménykezelő

(cbIrsz_SelectionChanged) készítése. Amikor bármelyikben változik a kiválasztott elem, akkor a két szövegmező tartalmát frissítjük.

```
private void cbIrsz_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    var enAktuális = ((ComboBox) sender).SelectedItem as enHelység;
    cbHelységnév.SelectedItem = enAktuális;
    tbIrsz.Text = enAktuális.Irsz.ToString();
    tbHelységnév.Text = enAktuális.Név;
}
```

A Rögzít gombon történő kattintáskor a gyűjteményben módosítjuk az adatokat.

```
private void btRögzít_Click(object sender, RoutedEventArgs e)
{
    var enAktuális = cbIrsz.SelectedItem as enHelység;
    //
    // Ide ellenőrzés kellene az adatok rögzítése előtt.
    // ...
    //
    enAktuális.Irsz = Int16.Parse(tbIrsz.Text);
    enAktuális.Név = tbHelységnév.Text;
    grHelység.Visibility = Visibility.Hidden;
}
```

A fenti metódus nem tartalmazza a szövegmezőbe írt adatok ellenőrzését, ez az olvasó önálló feladata. A gyűjteménybe történő felvétel még nem jelenti az adatbázisban történő tárolást. Ez utóbbi csak a Fájl menü Mentés pontjának választása után történik meg.

A Vissza gombon történő kattintás hatására eltüntetjük az űrlapot.

```
private void btVissza_Click(object sender, RoutedEventArgs e)
{
    grHelység.Visibility=Visibility.Hidden;
}
```

Amennyiben a felhasználó új helységadatokat akar felvinni az adatbázisba, akkor az "Új helység" gombon kattint, ami a két ComboBox letiltását és a szövegmezők ürítését eredményezi.

Telefor	nszám nyilvántartó 🛛 🗕 🔍
Fájl Lekérdezés Adatfelvitel/	Módosítás
Keresés irányítószám alapján	6000 ~
Keresés helységnév alapján	Kecskemét ~
lrányítószám	
Helységnév	
Módosított adatpár rög:	zítése Új helység Vissza

```
private void btÚjHelység_Click(object sender, RoutedEventArgs e)
{
     Beállít(false);
     tbIrsz.Text = "";
     tbHelységnév.Text = "";
}
private void Beállít(bool b)
{
     btÚjHelység.IsEnabled = b;
     cbIrsz.IsEnabled = b;
     cbHelységnév.IsEnabled = b;
}
```

A fentiekben ismertetett btRögzít_Click metódust kis módosítással alkalmassá tesszük az új adatok rögzítésére is. Az új változatban létrehozunk egy új enHelység objektumot, majd felvesszük azt a helységek gyűjteményébe.

```
private void btRögzít_Click(object sender, RoutedEventArgs e)
{
       var enAktuális = cbIrsz.SelectedItem as enHelység;
       11
       // Ide ellenőrzés kellene az adatok rögzítése előtt.
       // ...
       11
       if (!btÚjHelység.IsEnabled)
       {
              enAktuális = new enHelység();
              cnTelefonszámok.enHelységek.Add(enAktuális)
       }
       enAktuális.Irsz = Int16.Parse(tbIrsz.Text);
       enAktuális.Név = tbHelységnév.Text;
       grHelység.Visibility = Visibility.Hidden;
}
```

Hasonlóan a módosítás esetéhez, tartós tárolás itt is csak a Mentés menüpont segítségével érhető el.

Tartalomjegyzék

Entity Framework alapú adatbáziselérés 2	1
1. Projekt és alapbeállítások	1
2. A felület elkészítése	6
3. Egyszerű lekérdezés	9
4. Komplex lekérdezés	9
5. Helységadatok	. 15