

Programování desktopových aplikací

Semestrální projekt

Dan Humpál, Kristýna Kaplanová, Bára Skalická

Červen, 2019

Obsah

1	Úvod	4
2	Řešení	4
2.1	Databáze	4
2.2	Zobrazení	4
2.3	Vytváření dat	6
2.3.1	Přidání záznamu do tabulky Pony	6
2.3.2	Přidání obrázku do galerie	7
3	Závěr	9

Výpisy kódu

1	Metoda zajišťující naplnění galerie	5
2	Metoda obsluhující stisknutí tlačítka přidání	6
3	Metoda obsluhující stisknutí tlačítka upload	8

Seznam obrázků

1	Diagram databáze	4
2	Hlavní formulář aplikace	5
3	Formulář přidání poníka	6
4	Formulář přidání obrázku	7

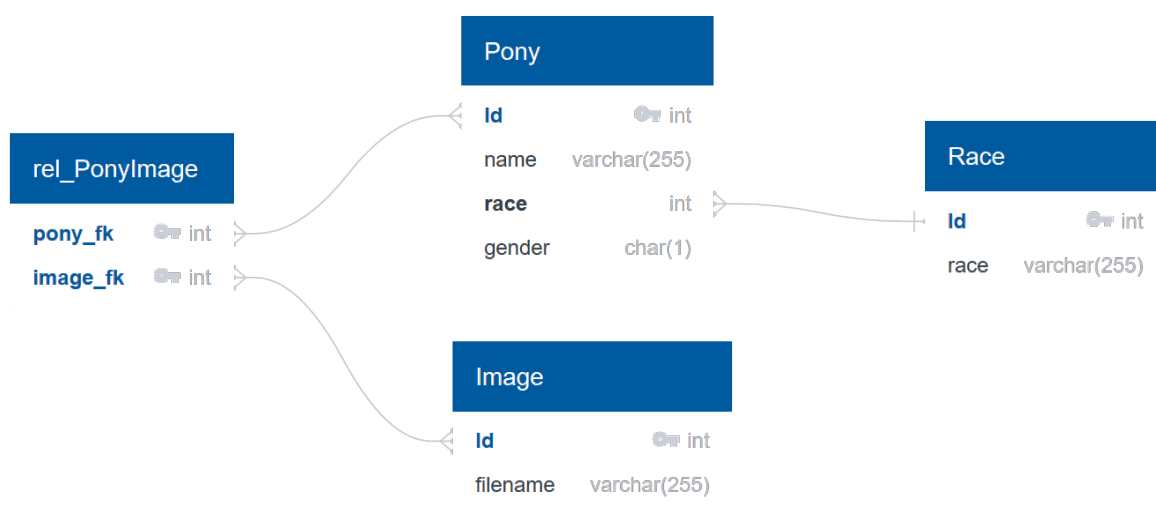
1 Úvod

Cílem práce je vytvoření fungující aplikace používající možnosti jazyka C# a Windows Forms. Aplikace musí splňovat základní stylizační nutnosti kódu a být ve vhodném grafickém provedení. Práce byla vytvářena ve skupině. Zde jsme se rozhodli vytvořit databázovou aplikaci na téma zpracování postav seriálu My Little Pony: Friendship is Magic, doplněné o jednoduché zpracování galerie obrázků.

2 Řešení

Řešení je vhodně rozděleno do částí, databáze, zobrazení, vytváření dat

2.1 Databáze

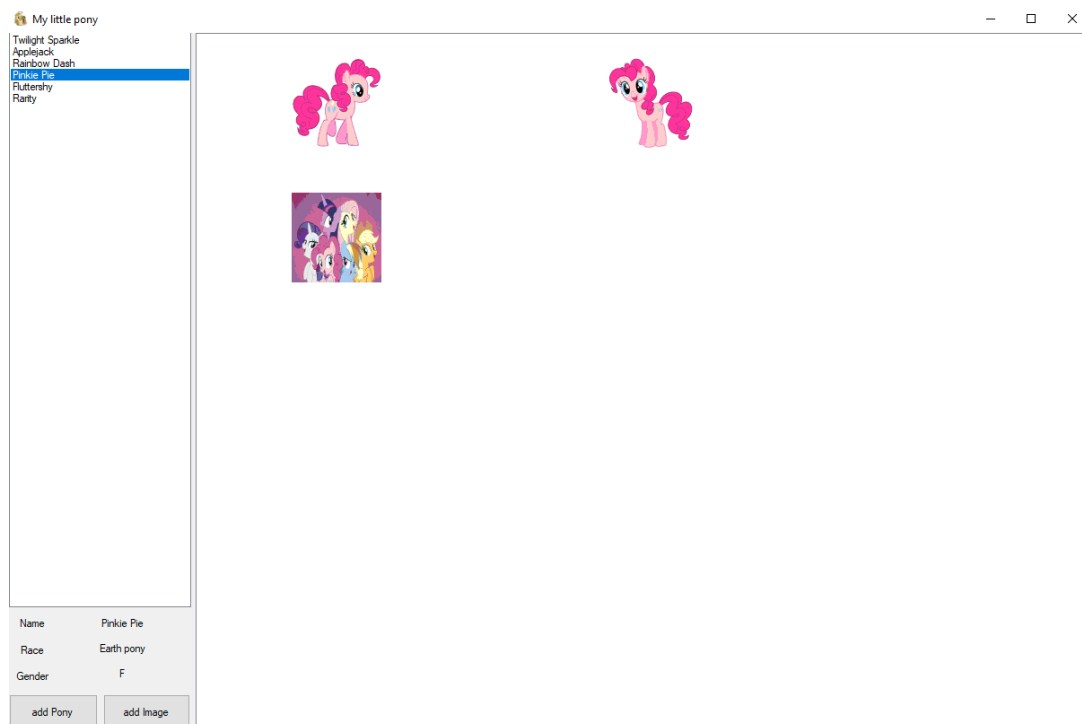


Obr. 1: Diagram databáze

Jak je patrné z diagramu, databáze obsahuje 4 vzájemně propojené tabulky. Obrázky jsou ukládány pouze ve formě jmen souborů, Není tak třeba řešit ukládání Blobů, či Filestreamů.

2.2 Zobrazení

Formulář zajišťující zobrazení dat je zároveň hlavním formulářem aplikace a prvním formulářem, který se po spuštění aplikace zobrazí. Formulář je rozdělený na 2 hlavní části. Po levé straně je komponenta ListView, která zobrazuje seznam objektů Pony v databázi. V této komponentě



Obr. 2: Hlavní formulář aplikace

je také možné, vybrat jeden ze záznamů, ke kterému se zobrazí podrobnosti a do pravé strany formuláře se nahraje seznam přiřazených obrázků. Toto je zajišťováno metodou refreshGallery.

Výpis 1: Metoda zajišťující naplnění galerie

```

1 private void refreshGallery()
2 {
3     if ((pony_list.SelectedItem as DataRowView) != null)
4     {
5         int pony_id = (int)((pony_list.SelectedItem as DataRowView) ["Id"]);
6         this.images.imagesview.Clear();
7         this.imagesviewTableAdapter.Fill(this.images.imagesview);
8         listView1.BeginUpdate();
9         listView1.Clear();
10        imageList1.Images.Clear();
11        foreach (DataRow dr in this.images.imagesview.Select(String.Format("
12            pony_fk = {0}", pony_id)))
13        {
14            imageList1.Images.Add(Image.FromFile(String.Format("./img/{0}", dr["
15                filename"])));
16        }
17        for (int j = 0; j < this.imageList1.Images.Count; j++)
18        {
19            ListViewItem item = new ListViewItem();
20            item.ImageIndex = j;
21            this.listView1.Items.Add(item);
22        }
23        listView1.EndUpdate();

```

```

22     //listView1.Update();
23 }

```

2.3 Vytváření dat

Zbylé 2 formuláře aplikace se zabývají vytvářením nových záznamů v databázi.

2.3.1 Přidání záznamu do tabulky Pony

Obr. 3: Formulář přidání poníka

Formulář umožňuje nastavení datových polí tabulky Pony. Záznamy v tabulce Pony jsou omezené existencí odpovídajícího záznamu v tabulce Race pomocí cizího klíče. Hlavní logika formuláře se nachází v metodě, která obsluhuje stisknutí tlačítka Přidání. Jednotlivé parametry jsou zde poté vytaženy z prvků formuláře a vloženy do databáze.

Výpis 2: Metoda obsluhující stisknutí tlačítka přidání

```

1 private void addpony_Click(object sender, EventArgs e)
2 {
3     string jmeno = textBox1.Text;
4
5     string rasa = "";
6
7     if (rb_earth.Checked == true)
8     {
9         rasa = "Earth Pony";
10    }
11    if (rb_uni.Checked == true)
12    {
13        rasa = "Unicorn";
14    }
15    if (rb_pegas.Checked == true)
16    {
17        rasa = "Pegasus";
18    }
19
20    string pohlavi = "";

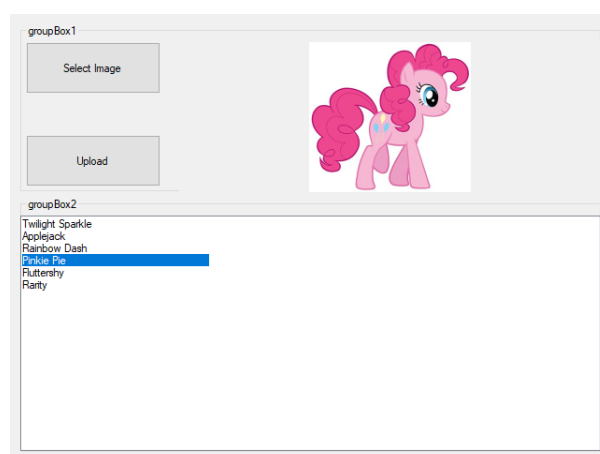
```

```

21
22     if (rb_M.Checked == true)
23     {
24         pohlavi = "M";
25     }
26     if (rb_F.Checked == true)
27     {
28         pohlavi = "F";
29     }
30
31
32
33     using (SqlConnection openCon = new SqlConnection(ConfigurationManager.
        ConnectionStrings["xPDA_LS_2019.Properties.Settings.
        poniesConnectionString"].ConnectionString))
34     {
35         string savePony = "INSERT INTO Pony (name, race, gender) VALUES (
            @name,(SELECT Id FROM Race WHERE race = @race),@gender)";
36
37         using (SqlCommand querySavePony = new SqlCommand(savePony))
38         {
39             querySavePony.Connection = openCon;
40             querySavePony.Parameters.AddWithValue("@name", jmeno);
41             querySavePony.Parameters.AddWithValue("@gender", pohlavi);
42             querySavePony.Parameters.AddWithValue("@race", rasa);
43
44             openCon.Open();
45             querySavePony.ExecuteNonQuery();
46             openCon.Close();
47         }
48     }
49     this.Close();
50 }

```

2.3.2 Přidání obrázku do galerie



Obr. 4: Formulář přidání obrázku

Obrázky je možné přiřadit k více poníkům. Přidání obrázku se provede jeho vybráním ze souborů počítače a stisknutím tlačítka upload. Aplikace pak nakopíruje vybraný obrázek do souborů aplikace a zajistí vytvoření nutných záznamů v databázi. Přiřazení poníci jsou vybrání pomocí komponenty Lisbox ve spodní části formuláře. Hlavní logika tohoto formuláře je implementována v metodě obsluhující tlačítko upload.

Výpis 3: Metoda obsluhující stisknutí tlačítka upload

```
1 private void Button2_Click(object sender, EventArgs e)
2 {
3     string imgpath = @"./img/";
4
5     FileInfo fi = new FileInfo(this._file);
6     if (fi.Exists)
7     {
8         if (!Directory.Exists(imgpath))
9         {
10             Directory.CreateDirectory(imgpath);
11         }
12         try
13         {
14             fi.CopyTo(string.Format("{0}{1}", imgpath, fi.Name));
15         }
16         catch (Exception ex)
17         {
18             MessageBox.Show("An error occured, try again");
19             this.Close();
20         }
21         //add img to db
22         String connectionString = ConfigurationManager.ConnectionStrings["
                xPDA_LS_2019.Properties.Settings.poniesConnectionString"].
                ConnectionString;
23         String imgquery = "INSERT INTO Image(filename) output INSERTED.ID
                VALUES (@path)";
24         int ? new_id = null;
25
26         using (SqlConnection conn = new SqlConnection(connectionString))
27         using (SqlCommand cmd = new SqlCommand(imgquery, conn))
28         {
29             try
30             {
31                 cmd.Parameters.AddWithValue("@path", fi.Name);
32                 conn.Open();
33                 new_id = (int)cmd.ExecuteScalar();
34                 conn.Close();
35             }
36             catch (Exception ex)
37             {
38                 File.Delete(string.Format("{0}{1}", imgpath, fi.Name));
39                 MessageBox.Show("An error occured, try again");
40                 this.Close();
41             }
42         }
43
44         foreach (DataRowView pony in ponyselect.SelectedItems)
45         {
46             String ponyimgquery = "INSERT INTO rel_PonyImage(pony_fk,
```



```

        image_fk) VALUES (@pony_id, @image_id)";
47      using (SqlConnection conn = new SqlConnection(connectionString))
48      using (SqlCommand cmd = new SqlCommand(ponyimgquery, conn))
49      {
50          cmd.Parameters.AddWithValue("@pony_id", pony["Id"]);
51          cmd.Parameters.AddWithValue("@image_id", new_id);
52          conn.Open();
53          cmd.ExecuteNonQuery();
54          conn.Close();
55      }
56  }
57  this.Close();
58  }
59  else
60  {
61      MessageBox.Show("An error occured, try again");
62      this.Close();
63  }
64  }

```

3 Závěr

Aplikace demonstruje využití základních komponent Windows Forms. Windows Forms obsahuje několik vrstev abstrakce, a je tak možná dosahovat stejných výsledků různými cestami, např. využití DataSet nebo volání dotazu přímo na databázi. V naší práci jsme různé přístupy zkombovali. Skupinová práce byla vytvářena a koordinována pomocí služby GitHub ve veřejném repozitáři.