A screenshot of a computer

Description automatically generated

WPF Student

<Window x:Class="\_27decembar.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:\_27decembar"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<StackPanel Margin="10">

<Image Source="pionir-logo.png" Width="100" Height="100" Margin="5"/>

<TextBlock Text="Ime:" Margin="10"/>

<TextBox x:Name="txtIme" Margin="10,0,0,0" Width="150" HorizontalAlignment="Left"/>

<TextBlock Text="Index:" Margin="10"/>

<TextBox x:Name="txtIndex" Margin="10,0,0,0" Width="150" HorizontalAlignment="Left"/>

<TextBlock Text="OOP ocena:" Margin="10"/>

<TextBox x:Name="txtOOPocena" Margin="10,0,0,0" Width="150" HorizontalAlignment="Left"/>

<TextBlock Text="ASP ocena:" Margin="10"/>

<TextBox x:Name="txtASPocena" Margin="10,0,0,0" Width="150" HorizontalAlignment="Left"/>

<StackPanel Orientation="Horizontal" Margin="10">

<Button Content="Dodaj u listu" Click="Dodaj" Margin="0,0,5,0"/>

<Button Content="Obrisi sve" Click="BrisiSve" Margin="0,0,5,0"/>

<Button Content="Obrisi selektovan" Click="BrisiSelektovan" Margin="0,0,5,0"/>

<Button Content="Proveri podatke" Click="Provera" Margin="0,0,5,0"/>

<Button Content="Prosek ocena" Click="Prosek" Margin="0,0,5,0"/>

</StackPanel>

<ListView x:Name="lstPodaci" SelectionMode="Single" Margin="5">

<ListView.View>

<GridView>

<GridViewColumn Header="Ime" DisplayMemberBinding="{Binding Ime}" Width="150"/>

<GridViewColumn Header="Index" DisplayMemberBinding="{Binding BrojIndexa}" Width="50"/>

<GridViewColumn Header="OOP ocena" DisplayMemberBinding="{Binding OOPocena}" Width="50"/>

<GridViewColumn Header="ASP ocena" DisplayMemberBinding="{Binding ASPocena}" Width="50"/>

</GridView>

</ListView.View>

</ListView>

</StackPanel>

</Grid>

</Window>

Klasa Student

namespace WPF\_student\_ocene

{

internal class Student

{

public string Ime { get; set; }

public string BrojIndexa { get; set; }

public int OOPOcena { get; set; }

public int AlgoritmiOcena { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Diagnostics;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WPF\_student\_ocene

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private List<Student> studenti= new List<Student>();

const int duzinaIndexa = 8;

public MainWindow()

{

InitializeComponent();

lstPodaci.ItemsSource = studenti;

}

private void Dodaj(object sender, RoutedEventArgs e)

{

if (ProveriSve() == true)

{

string ime = txtIme.Text;

string brojIndexa = txtIndex.Text;

int oopOcena = int.Parse(txtOOPocena.Text);

int aspOcena = int.Parse(txtASPocena.Text);

double prosek = 0;

Student noviStudent = new Student();

noviStudent.Ime = ime;

noviStudent.BrojIndexa = brojIndexa;

noviStudent.OOPocena = oopOcena;

noviStudent.ASPocena = aspOcena;

studenti.Add(noviStudent);

// lstPodaci.ItemsSource = studenti;

lstPodaci.Items.Refresh();

// prosek=(oopOcena+aspOcena)/2;

OcistiSveTextBoxove();

}

}

private void OcistiSveTextBoxove()

{

txtASPocena.Clear();

txtOOPocena.Clear();

txtIme.Clear();

txtIndex.Clear();

}

private void BrisiSve(object sender, RoutedEventArgs e)

{

studenti.Clear();

lstPodaci.Items.Refresh();

}

private void BrisiSelektovan(object sender, RoutedEventArgs e)

{

if (lstPodaci.SelectedIndex < 0)

{

MessageBox.Show("Odaberite samo jednu stavku!");

}

else

{

studenti.RemoveAt(lstPodaci.SelectedIndex);

lstPodaci.Items.Refresh();

}

}

private bool ProveriOcenu(string ocenaText)

{

bool provera = true;

if ( int.TryParse(ocenaText, out int grade))

{

if(grade>=5 && grade <= 10)

{

MessageBox.Show("Unos je ispravan. Ocena: " + grade);

}

else

{

MessageBox.Show("Greska! Ocena mora da bude izmedju 5 i 10");

provera = false;

}

}

else

{

MessageBox.Show("Unos nije validan, treba da bude broj!");

provera = false;

}

return provera;

}

private bool ProveriIndex(string brojIndeksa)

{

if (brojIndeksa.Length == duzinaIndexa)

{

MessageBox.Show("Unos je ispravan za indeks");

return true;

}

else

{

MessageBox.Show("Proverite duzinu broja indeksa!");

return false;

}

}

private bool ProveriIme(string ime)

{

if(ime.Contains(" "))

{

MessageBox.Show("Ime nije ispravno, sadrzi razmak!");

return false;

}

else

{

MessageBox.Show("Ime je ispravno uneto.");

return true;

}

}

private bool ProveriSve()

{

//da li su ocene OK

if (!ProveriOcenu(txtOOPocena.Text) || !ProveriOcenu(txtASPocena.Text))

{

return false;

}

if(!ProveriIndex(txtIndex.Text))

{

return false;

}

if(!ProveriIme(txtIme.Text))

{

return false;

}

return true;

}

private void Provera(object sender, RoutedEventArgs e)

{

}

private void Prosek(object sender, RoutedEventArgs e)

{

double oopProsek = 0;

double aspProsek = 0;

if(studenti.Count > 0)

{

foreach(Student s in studenti)

{

oopProsek += s.OOPocena;

aspProsek+= s.ASPocena;

}

oopProsek/=studenti.Count;

aspProsek/=studenti.Count;

}

MessageBox.Show("Prosek ocena je: " + oopProsek +" " +aspProsek, "Prosek ocena", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}