Klasa Student

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SamostalanRad

{

public class Student

{

public string FirstName { get; set; }

public string LastName { get; set; }

public string IndexNumber { get; set; }

public override string ToString()

{

return $"{FirstName} {LastName} ({IndexNumber})";

}

}

}

XAML

<Window x:Class="SamostalanRad.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:SamostalanRad"

mc:Ignorable="d"

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

<Grid Margin="10">

<!-- ListView to display students -->

<ListView x:Name="StudentListView"

ItemsSource="{Binding Students}"

SelectedItem="{Binding SelectedStudent}"

Margin="0,0,0,50">

<ListView.View>

<GridView>

<GridViewColumn Header="First Name" DisplayMemberBinding="{Binding FirstName}" Width="150" />

<GridViewColumn Header="Last Name" DisplayMemberBinding="{Binding LastName}" Width="150" />

<GridViewColumn Header="Index Number" DisplayMemberBinding="{Binding IndexNumber}" Width="100" />

</GridView>

</ListView.View>

</ListView>

<!-- Buttons -->

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Bottom" Margin="0,10,0,0">

<Button Content="Read Data" Width="100" Margin="5" Click="ReadData\_Click" />

<Button Content="Clear List" Width="100" Margin="5" Click="ClearList\_Click" />

<Button Content="Delete Selected" Width="120" Margin="5" Click="DeleteSelected\_Click" />

<Button Content="Unique Names" Width="120" Margin="5" Click="UniqueNames" />

</StackPanel>

</Grid>

</Window>

using System.IO;

using System.Text;

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 SamostalanRad

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

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

public MainWindow()

{

InitializeComponent();

}

private void ReadData\_Click(object sender, RoutedEventArgs e)

{

try

{

var filePath = @"c:\tmp\studenti1.txt"; // Path to the text file

if (!File.Exists(filePath))

{

MessageBox.Show("File not found.", "Error", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

var lines = File.ReadAllLines(filePath);

foreach (var line in lines)

{

var parts = line.Split(' ');

if (parts.Length == 3)

{

var student = new Student

{

LastName = parts[0].Trim(),

FirstName = parts[1].Trim(),

IndexNumber = parts[2].Trim()

};

if (!students.Contains(student))

{

students.Add(student);

}

}

}

RefreshStudentListView();

}

catch (Exception ex)

{

MessageBox.Show($"An error occurred: {ex.Message}", "Error", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void ClearList\_Click(object sender, RoutedEventArgs e)

{

students.Clear();

RefreshStudentListView();

}

private void DeleteSelected\_Click(object sender, RoutedEventArgs e)

{

if (StudentListView.SelectedItem is Student selectedStudent)

{

students.Remove(selectedStudent);

RefreshStudentListView();

}

else

{

MessageBox.Show("No student selected.", "Warning", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

private void RefreshStudentListView()

{

StudentListView.ItemsSource = null;

StudentListView.ItemsSource = students;

}

private void UniqueNames(object sender, RoutedEventArgs e)

{

var uniqueNames = new Dictionary<string, bool>();

foreach (var student in students)

{

if (!uniqueNames.ContainsKey(student.FirstName))

{

uniqueNames[student.FirstName] = true;

}

}

// Kreiranje liste studenata sa jedinstvenim FirstName vrednostima

var uniqueNameList = new List<Student>();

// Iteriranje kroz sve jedinstvene FirstName vrednosti

foreach (var name in uniqueNames.Keys)

{

// Kreiramo novi objekat Student

Student newStudent = new Student();

// Dodeljujemo vrednost FirstName

newStudent.FirstName = name;

// Dodajemo novog studenta u listu

uniqueNameList.Add(newStudent);

}

// Postavljanje izvora podataka za ListView

StudentListView.ItemsSource = uniqueNameList;

}

}

}