Федеральное государственное образовательное бюджетное учреждение высшего образования

«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ

РОССИЙСКОЙ ФЕДЕРАЦИИ»

**Колледж информатики и программирования**

Отчёт по заданию

«Разработка WPF-приложения для управления базой данных студентов»

Выполнила:

студентка группы 3ИСИП-422

Никифорова Ю. Р.

Проверил:

преподаватель

Тарасов И. Ю.

«\_\_» \_\_\_\_\_\_\_\_ 2025 г.

Москва

2025

СОДЕРЖАНИЕ

[ПРАКТИЧЕСКАЯ ЧАСТЬ 3](#_Toc178729302)

# ПРАКТИЧЕСКАЯ ЧАСТЬ

Цель работы: разработать простое приложение с графическим пользовательским интерфейсом (GUI) на C#. Приложение должно демонстрировать основные элементы управления и их взаимодействие.

Описание задания:

Разработайте WPF-приложение, которое позволяет пользователю управлять базой данных студентов. Приложение должно поддерживать операции добавления, редактирования, удаления и фильтрации данных о студентах и их среднем балле. Для взаимодействия с базой данных используйте Entity Framework.

Внизу представлены 4 листинга кода, а также рисунки работоспособности приложения.

Листинг 1 – MainWindow.xaml

|  |
| --- |
| <Window x:Class="StudentsApp.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:StudentsApp"  mc:Ignorable="d"  Title="Студенты" Height="450" Width="800">  <Grid>  <Grid.ColumnDefinitions>  <ColumnDefinition></ColumnDefinition>  <ColumnDefinition></ColumnDefinition>  </Grid.ColumnDefinitions>  <DataGrid x:Name="DataGrid" AutoGenerateColumns="True" Grid.Column="0"/>  <Button x:Name="Filter" Grid.Column="1" Content="Фильтрация" HorizontalAlignment="Left" Margin="268,49,0,0" VerticalAlignment="Top" Height="34" Width="122" Click="Filter\_Click"/>  <TextBox x:Name="FilterTextBox" Grid.Column="1" HorizontalAlignment="Left" Margin="77,49,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="191" Height="34"/>  <Label Grid.Column="1" Content="Фамилия:" HorizontalAlignment="Left" Margin="77,23,0,0" VerticalAlignment="Top"/>  <Button x:Name="Delete" Grid.Column="1" Content="Удалить" HorizontalAlignment="Left" Margin="268,130,0,0" VerticalAlignment="Top" Height="34" Width="122" Click="Delete\_Click"/>  <Button x:Name="Add" Grid.Column="1" Content="Добавить" HorizontalAlignment="Left" Margin="268,0,0,0" VerticalAlignment="Center" Height="34" Width="122" Click="Add\_Click" />  <Button x:Name="Edit" Grid.Column="1" Content="Редактировать" HorizontalAlignment="Left" Margin="268,272,0,0" VerticalAlignment="Top" Height="34" Width="122" Click="Edit\_Click" />  </Grid>  </Window> |

Листинг 2 – MainWindow.xaml.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  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 StudentsApp  {  /// <summary>  /// Логика взаимодействия для MainWindow.xaml  /// </summary>  public partial class MainWindow : Window  {  public MainWindow()  {  InitializeComponent();  using (var context = new Entities())  {  var res = context.student.ToList();  DataGrid.ItemsSource = res;  }  }  private void Filter\_Click(object sender, RoutedEventArgs e)  {  if(!string.IsNullOrWhiteSpace(FilterTextBox.Text))  {  using (var context = new Entities())  {  var res = context.student.Where(s => s.LastName.Contains(FilterTextBox.Text)).ToList();  DataGrid.ItemsSource = res;  }  }  else  {  MessageBox.Show("Введите значение");  using (var context = new Entities())  {  var res = context.student.ToList();  DataGrid.ItemsSource = res;  }  }  }  private void Delete\_Click(object sender, RoutedEventArgs e)  {  if (DataGrid.SelectedItem is student selectedStudent)  {  var result = MessageBox.Show($"Удалить студента {selectedStudent.LastName}?", "Подтверждение удаления", MessageBoxButton.YesNo, MessageBoxImage.Question);  if (result == MessageBoxResult.Yes)  {  using (var context = new Entities())  {  var studentToDelete = context.student.Find(selectedStudent.Id);  if (studentToDelete != null)  {  context.student.Remove(studentToDelete);  context.SaveChanges();  DataGrid.ItemsSource = context.student.ToList();  }  }  }  }  else  {  MessageBox.Show("Выберите студента для удаления", "Ошибка",  MessageBoxButton.OK, MessageBoxImage.Warning);  }  }  private void Add\_Click(object sender, RoutedEventArgs e)  {  var editWindow = new EditStudentWindow();  if (editWindow.ShowDialog() == true)  {  using (var context = new Entities())  {  var student = new student()  {  FirstName = editWindow.FirstName,  LastName = editWindow.LastName,  AverageScore = editWindow.AverageScore  };  context.student.Add(student);  context.SaveChanges();  DataGrid.ItemsSource = context.student.ToList();  }  }  }  private void Edit\_Click(object sender, RoutedEventArgs e)  {  if (DataGrid.SelectedItem is student selectedStudent)  {  var editWindow = new EditStudentWindow(selectedStudent);  if (editWindow.ShowDialog() == true)  {  try  {  using (var context = new Entities())  {  var studentToUpdate = context.student.Find(selectedStudent.Id);  if (studentToUpdate == null)  {  MessageBox.Show("Студент не найден в базе данных", "Ошибка",  MessageBoxButton.OK, MessageBoxImage.Error);  return;  }  studentToUpdate.FirstName = editWindow.FirstName;  studentToUpdate.LastName = editWindow.LastName;  studentToUpdate.AverageScore = editWindow.AverageScore;  int changes = context.SaveChanges();  if (changes > 0)  {  DataGrid.ItemsSource = context.student.ToList();  }  }  }  catch (Exception ex)  {  MessageBox.Show($"Ошибка при обновлении студента: {ex.Message}", "Ошибка",  MessageBoxButton.OK, MessageBoxImage.Error);  }  }  }  else  {  MessageBox.Show("Выберите студента для редактирования", "Ошибка",  MessageBoxButton.OK, MessageBoxImage.Warning);  }  }  }  } |

Листинг 3 – EditStudentWindow.xaml

|  |
| --- |
| <Window x:Class="StudentsApp.EditStudentWindow"  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:StudentsApp"  mc:Ignorable="d"  Title="Редактирование студента" Height="250" Width="300">  <Grid Margin="10">  <Grid.RowDefinitions>  <RowDefinition Height="Auto"/>  <RowDefinition Height="Auto"/>  <RowDefinition Height="Auto"/>  <RowDefinition Height="Auto"/>  <RowDefinition Height="Auto"/>  <RowDefinition Height="Auto"/>  <RowDefinition Height="Auto"/>  </Grid.RowDefinitions>  <Label Grid.Row="0" Content="Имя:"/>  <TextBox x:Name="FirstNameTextBox" Grid.Row="1" Margin="0,0,0,10"/>  <Label Grid.Row="2" Content="Фамилия:"/>  <TextBox x:Name="LastNameTextBox" Grid.Row="3" Margin="0,0,0,10"/>  <Label Grid.Row="4" Content="Средний балл:"/>  <TextBox x:Name="AverageScoreTextBox" Grid.Row="5"/>  <StackPanel Grid.Row="6" Orientation="Horizontal" HorizontalAlignment="Right" Margin="0,10,0,0">  <Button x:Name="SaveButton" Content="Сохранить" Width="80" Margin="5,0" Click="SaveButton\_Click"/>  <Button x:Name="CancelButton" Content="Отмена" Width="80" Margin="5,0" Click="CancelButton\_Click"/>  </StackPanel>  </Grid>  </Window> |

Листинг 4 – EditStudentWindow.xaml.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  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.Shapes;  namespace StudentsApp  {  /// <summary>  /// Логика взаимодействия для EditStudentWindow.xaml  /// </summary>  public partial class EditStudentWindow : Window  {  public string FirstName => FirstNameTextBox.Text;  public string LastName => LastNameTextBox.Text;  public decimal AverageScore => decimal.Parse(AverageScoreTextBox.Text);  public EditStudentWindow()  {  InitializeComponent();  }  public EditStudentWindow(student student) : this()  {  FirstNameTextBox.Text = student.FirstName;  LastNameTextBox.Text = student.LastName;  AverageScoreTextBox.Text = student.AverageScore.ToString();  }  private void SaveButton\_Click(object sender, RoutedEventArgs e)  {  if (ValidateInput())  {  DialogResult = true;  Close();  }  }  private void CancelButton\_Click(object sender, RoutedEventArgs e)  {  DialogResult = false;  Close();  }  private bool ValidateInput()  {  if (string.IsNullOrWhiteSpace(FirstName))  {  MessageBox.Show("Введите имя студента", "Ошибка",  MessageBoxButton.OK, MessageBoxImage.Error);  return false;  }  if (string.IsNullOrWhiteSpace(LastName))  {  MessageBox.Show("Введите фамилию студента", "Ошибка",  MessageBoxButton.OK, MessageBoxImage.Error);  return false;  }  if (!decimal.TryParse(AverageScoreTextBox.Text, out decimal score) || score < 0 || score > 5)  {  MessageBox.Show("Введите корректный средний балл (0-5)", "Ошибка",  MessageBoxButton.OK, MessageBoxImage.Error);  return false;  }  return true;  }  }  } |

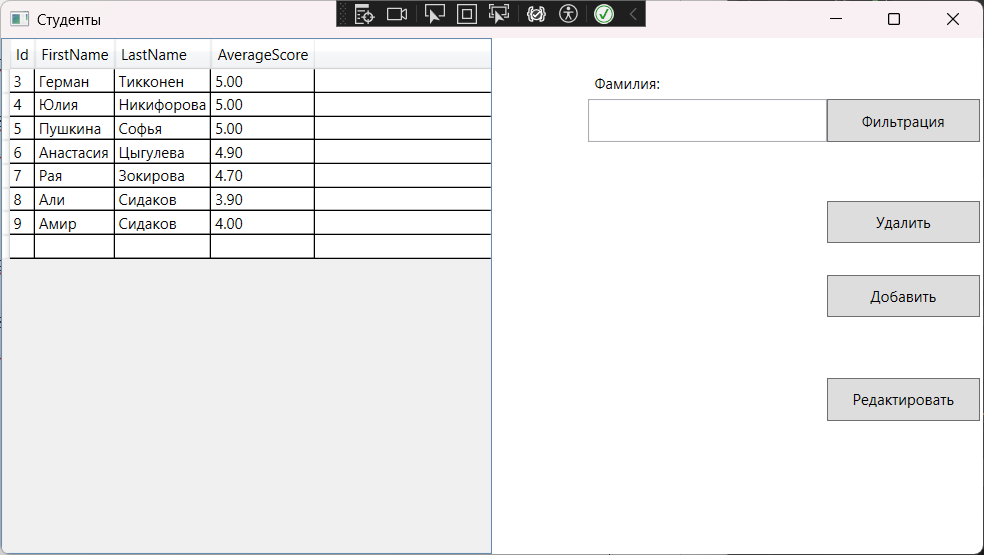


Рисунок 1. Главная страница

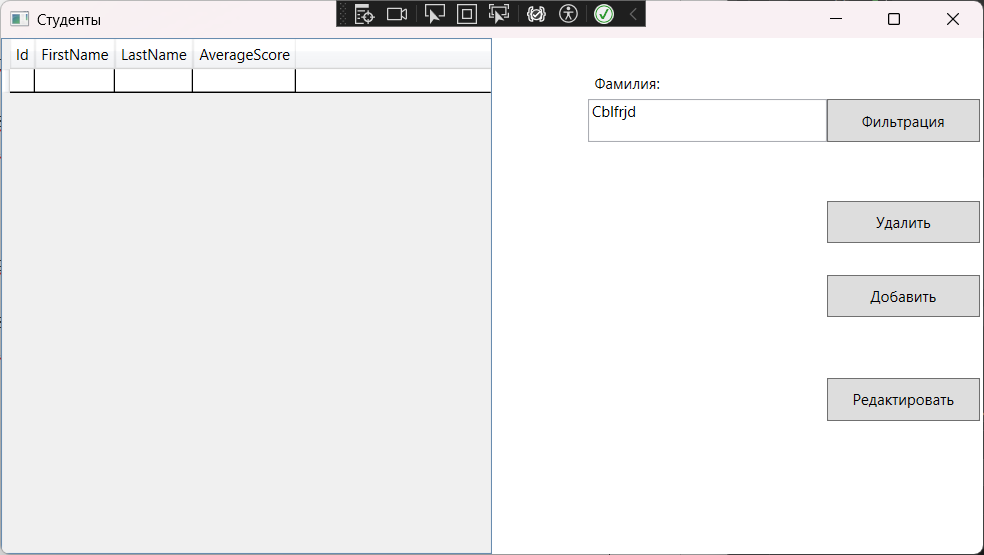


Рисунок 2. Фильтрация

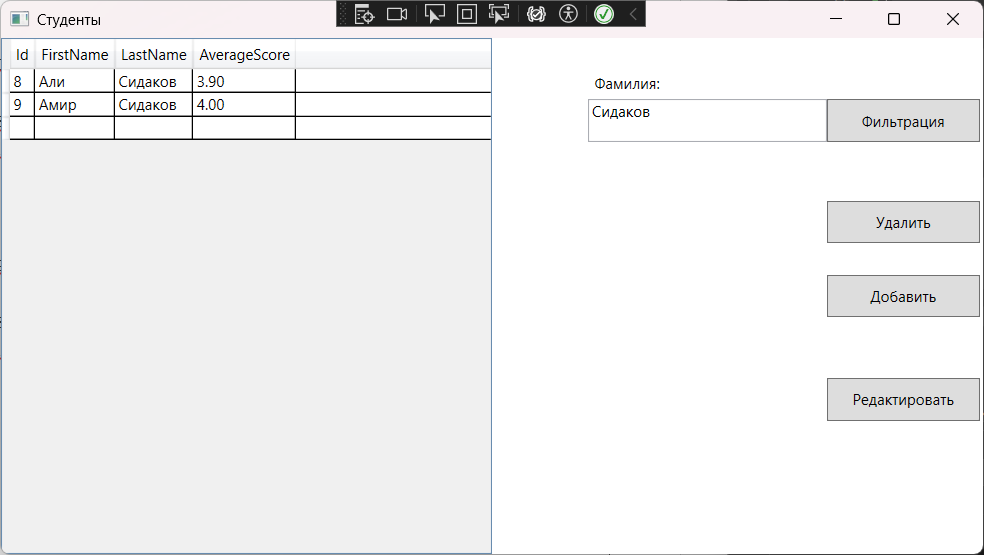


Рисунок 3. Фильтрация

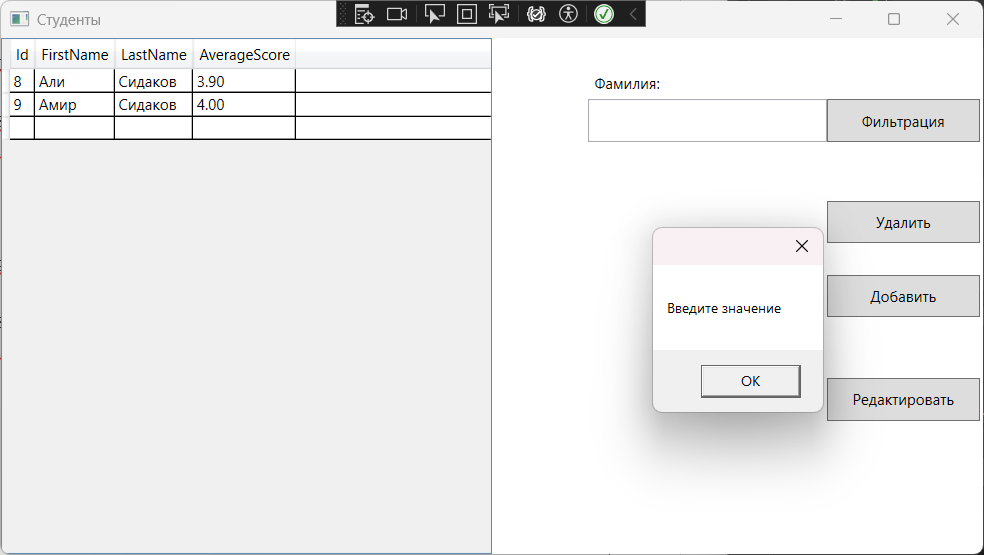


Рисунок 4. Фильтрация

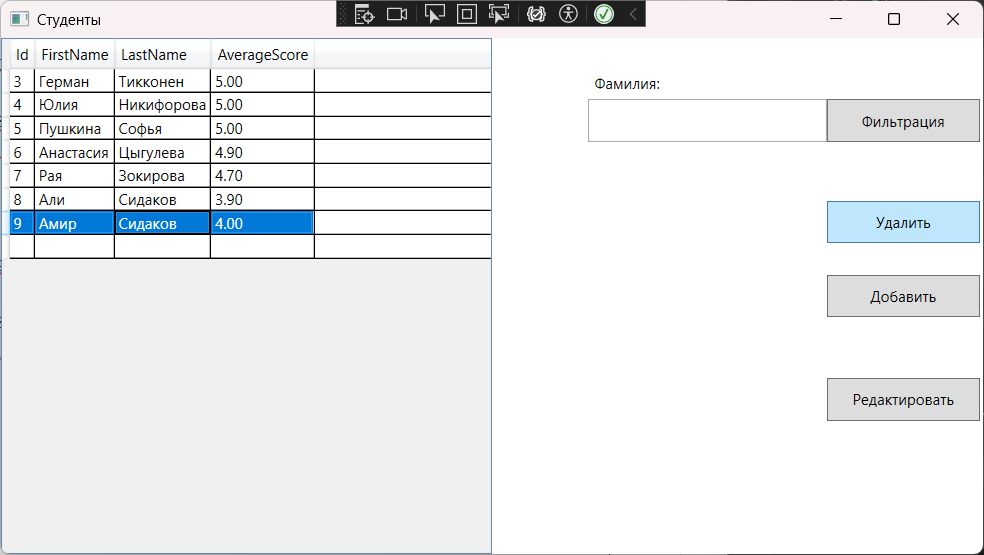


Рисунок 5. Удаление

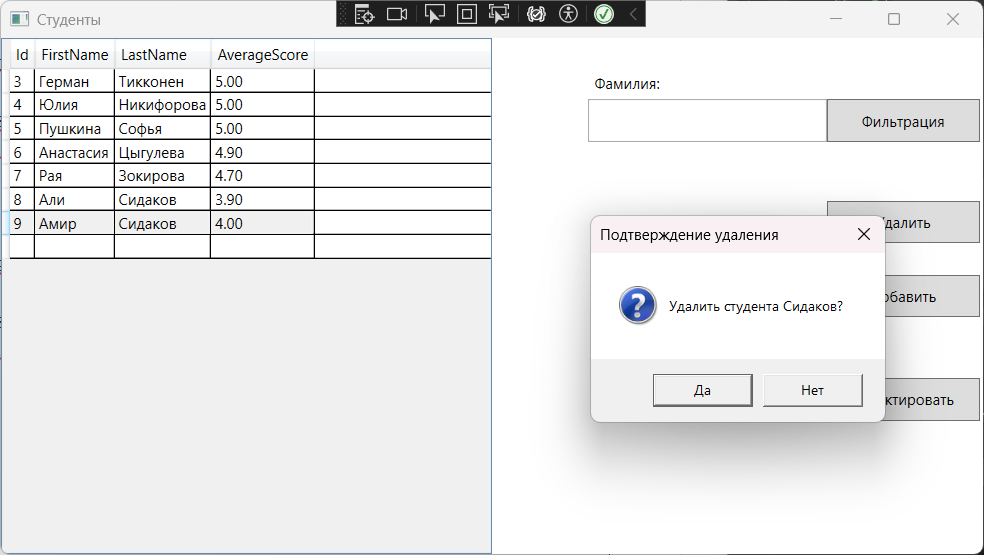


Рисунок 6. Подтверждение удаления

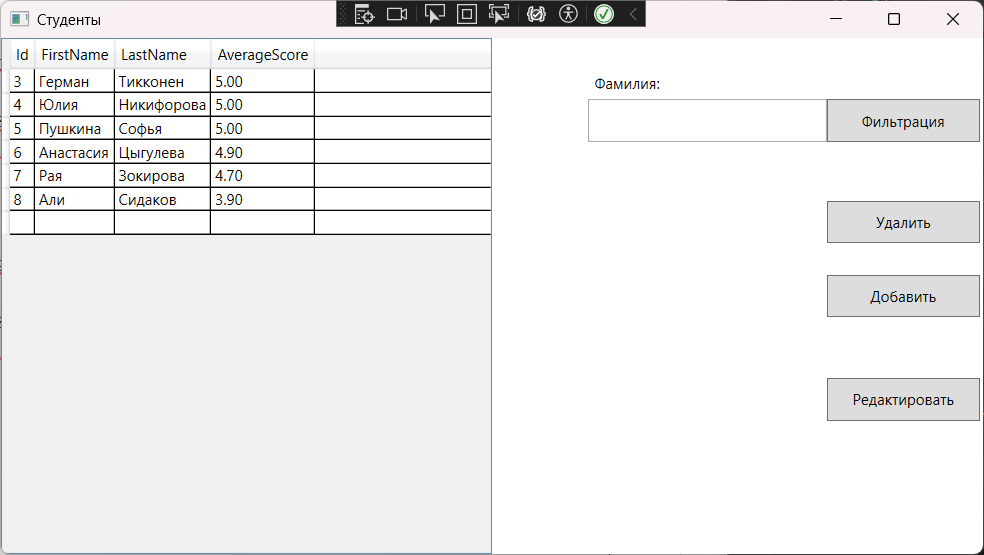


Рисунок 7. Успешное удаление

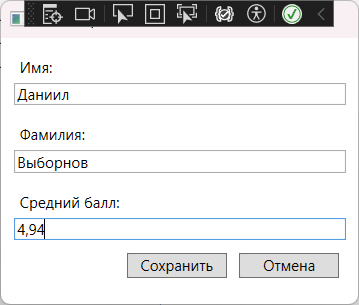


Рисунок 8. Добавление

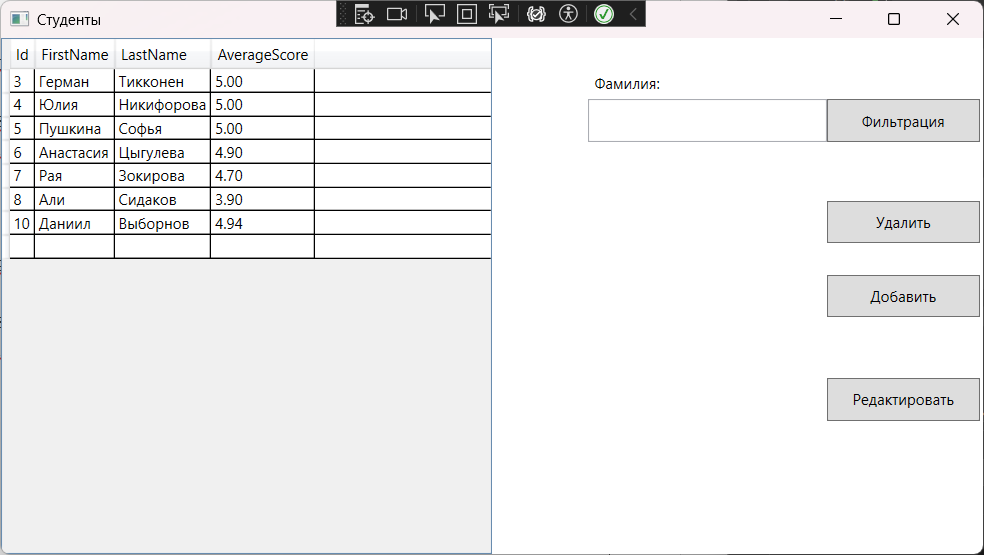


Рисунок 9. Успешное добавление

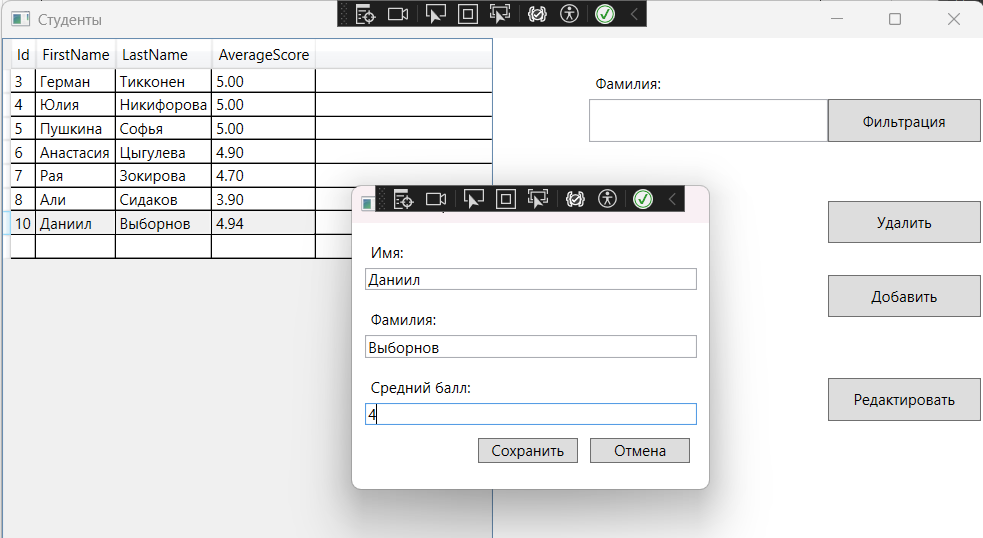


Рисунок 10. Редактирование

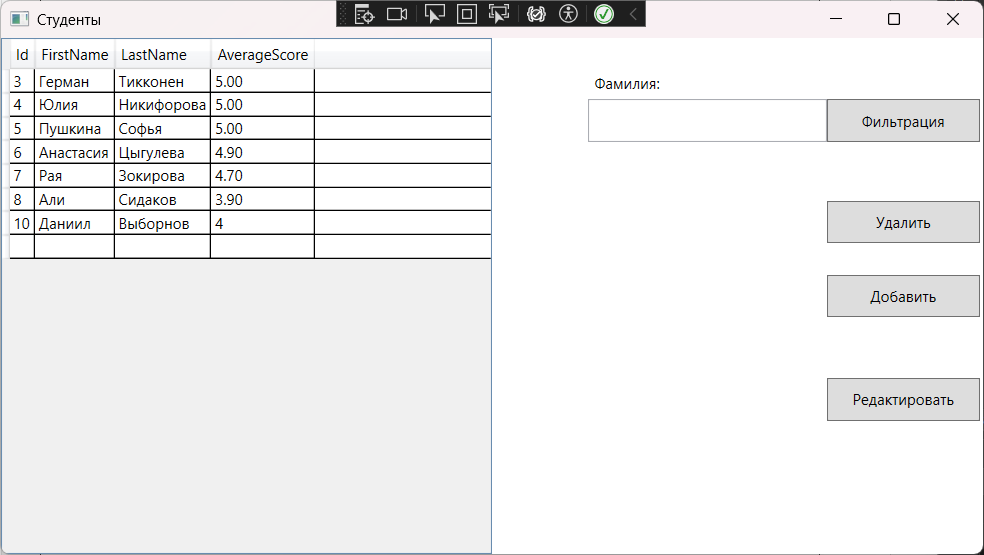


Рисунок 11. Успешное редактирование

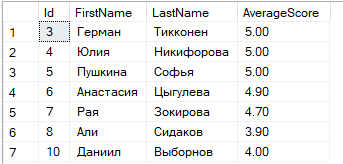


Рисунок 12. Таблица в базе данных

Вывод: было реализовано простое приложение со связкой базы данных. Всё работает успешно.