Министерство цифрового развития, связи и массовых коммуникаций РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

**Лабораторная работа №2**

**По дисциплине: Программирование и обработка графического интерфейса**

*«Работа с базами данных»*

**Выполнил:** Нагаев К.А.

**Группа:** ДИМ-32

**Вариант:** №2

**Проверил:** Голованчиков С.А.

Новосибирск, 2023 г.

**Лабораторная работа №2**

**«Работа с базами данных»**

**Целью** данной работы является получение навыков работы с баами данных на языке C# и знакомство с принципом построения SQL-запросов.

**Задание:** Необходимо разработать WPF-приложение с графическим интерфейсом и реализовать следующие функции:

1) ввод данных о студентах: уникальный номер, ФИО, оценка по физике, оценка по математике;

2) добавление данных в базу данных SQLite (далее - БД) через интерфейс приложения;

3) чтение данных из БД и отображение их в окне приложения;

4) редактирование данных в БД через интерфейс приложения;

5) удаление данных из таблиц.

**Задание 1:** Ввод данных о студентах: уникальный номер, ФИО, оценка по физике, оценка по математике

**Выполнение:**

Результат работы программы представлен на рисунке 1.1.

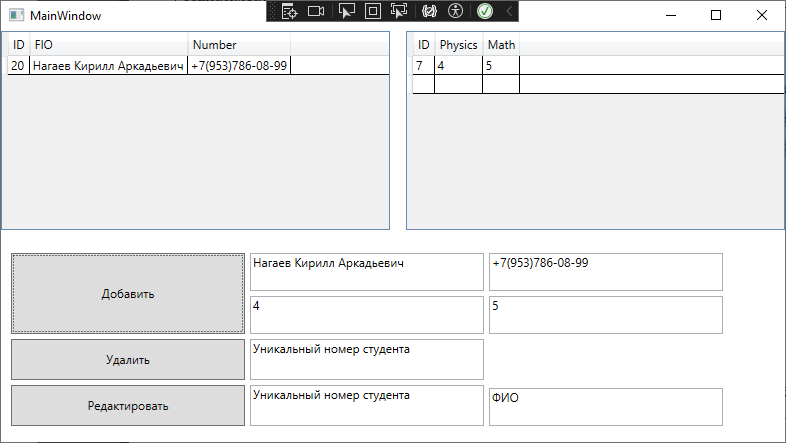


Рисунок 1.1 – Результат работы программы при нажатии на кнопку «Добавить»

Листинг 1.1

C# код

private void Button\_ClickADD(object sender, RoutedEventArgs e)

{

string connectionString = "Data Source=C:\\Prog\\FIO.db;Version=3;";

try

{

using (var connection = new SQLiteConnection(connectionString))

{

connection.Open();

string insertQuery = "INSERT INTO Students (FIO, Number) VALUES (@Value1, @Value2)";

string insertQueryO = "INSERT INTO Ocenki (Physics, Math) VALUES (@Value3, @Value4)";

using (SQLiteCommand command = new SQLiteCommand(insertQuery, connection))

{

command.Parameters.AddWithValue("@Value1", FIO.Text);

command.Parameters.AddWithValue("@Value2", Number.Text);

command.ExecuteNonQuery();

}

using (SQLiteCommand command = new SQLiteCommand(insertQueryO, connection))

{

command.Parameters.AddWithValue("@Value3", Phi.Text);

command.Parameters.AddWithValue("@Value4", Math.Text);

command.ExecuteNonQuery();

}

}

// После выполнения операции вставки, обновите данные в вашем DataGrid

string selectQuery = "SELECT \* FROM Students"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQuery, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

myDataGrid.ItemsSource = dataTable.DefaultView;

}

string selectQueryO = "SELECT \* FROM Ocenki"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQueryO, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

oDataGrid.ItemsSource = dataTable.DefaultView;

}

}

catch (Exception ex)

{

// Обработка исключения, например, вывод сообщения об ошибке

MessageBox.Show("Произошла ошибка: " + ex.Message);

}

}

XAML разметка

<Button Content="Добавить" HorizontalAlignment="Left" Height="81" Margin="10,222,0,0" VerticalAlignment="Top" Width="234" Click="Button\_ClickADD"/>  
<TextBox x:Name="FIO"

HorizontalAlignment="Left" Height="38" Margin="249,222,0,0" TextWrapping="Wrap" Text="ФИО" VerticalAlignment="Top" Width="234"/>

<TextBox x:Name="Number"

HorizontalAlignment="Left" Height="38" Margin="488,222,0,0" TextWrapping="Wrap" Text="Номер телефона" VerticalAlignment="Top" Width="234"/>

<TextBox x:Name="Phi"

HorizontalAlignment="Left" Height="38" Margin="249,265,0,0" TextWrapping="Wrap" Text="Физика" VerticalAlignment="Top" Width="234" />

<TextBox x:Name="Math"

HorizontalAlignment="Left" Height="38" Margin="488,265,0,0" TextWrapping="Wrap" Text="Математика" VerticalAlignment="Top" Width="234"/>

**Описание**

Приведенный код является частью приложения с графическим интерфейсом, написанного на платформе .NET с использованием WPF (Windows Presentation Foundation).

Основным классом в коде является `Button\_ClickADD`, который представляет функцию.

Данная функция представляет собой обработчик события нажатия кнопки. Она выполняет следующие действия:

* Устанавливает соединение с базой данных SQLite, используя строку подключения "Data Source=C:\Prog\FIO.db;Version=3;".
* Вставляет данные в таблицу "Students" с использованием введенных значений имени (FIO) и номера (Number).
* Вставляет данные в таблицу "Ocenki" с использованием введенных значений по физике (Physics) и математике (Math).
* После выполнения операций вставки, обновляет данные в DataGrid, отображая информацию из таблицы "Students".
* Также обновляет данные в другом DataGrid, отображая информацию из таблицы "Ocenki".
* Обрабатывает исключения, выводя сообщение об ошибке в случае возникновения исключения.

Эта функция предназначена для обработки нажатия кнопки "ADD" в пользовательском интерфейсе, и она выполняет важные операции по вставке данных в базу данных и обновлению отображаемых данных в DataGrid.

**Задание 2-3:** Реализуйте добавление данных в базу данных SQLite (далее - БД) через интерфейс приложения:

**Выполнение:**

Результат работы программы представлен на рисунке 2.1. Код программы представлен в листинге 2.1.

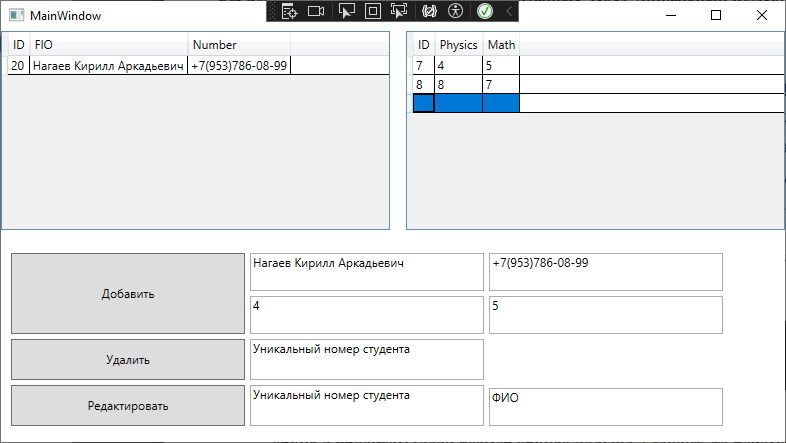


Рисунок 2.1 – Результат работы программы после добавления 8-й строки в оценки

Листинг 2.1

C# код

public MainWindow()

{

InitializeComponent();

var connectionString = "Data Source=C:\\Prog\\FIO.db;Version=3;";

var connection = new SQLiteConnection(connectionString);

connection.Open();

// \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

string sql = "SELECT \* FROM Students";

DataTable dataTable = new DataTable();

string sqlo = "SELECT \* FROM Ocenki";

DataTable dataTableo = new DataTable();

try

{

connection = new SQLiteConnection(connectionString);

SQLiteCommand command = new SQLiteCommand(sql, connection);

SQLiteDataAdapter adapter = new SQLiteDataAdapter(command);

SQLiteCommand commando = new SQLiteCommand(sqlo, connection);

SQLiteDataAdapter adaptero = new SQLiteDataAdapter(commando);

connection.Open();

adapter.Fill(dataTable);

myDataGrid.ItemsSource = dataTable.DefaultView;

//connection.Open();

adaptero.Fill(dataTableo);

oDataGrid.ItemsSource = dataTableo.DefaultView;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

if (connection != null) connection.Close();

}

//---------------------------

myDataGrid.Items.Refresh();

oDataGrid.Items.Refresh();

// \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

connection.Close();

}

XAML разметка

<DataGrid

x:Name="myDataGrid"

Margin="0,0,395,212"

AutoGenerateColumns="True"

CanUserAddRows="False" CanUserDeleteRows="False"

CanUserResizeRows="False" CanUserResizeColumns="False" CanUserReorderColumns="False"

IsReadOnly="False"

/>

<DataGrid

x:Name="oDataGrid"

Margin="405,0,0,212"

AutoGenerateColumns="True"

/>

**Описание**

Данный код представляет собой окна для отображения таблиц базы данных в интерфейсе WPF-приложения.

Данный код выполняет следующие действия:

* Инициализирует компоненты окна, вызывая метод InitializeComponent() в конструкторе класса MainWindow. Этот метод обычно генерируется автоматически и содержит логику для инициализации компонентов пользовательского интерфейса, определенных в XAML.
* Устанавливает соединение с базой данных SQLite, используя строку подключения "Data Source=C:\Prog\FIO.db;Version=3;".
* Выполняет запросы к базе данных для выборки данных из таблиц "Students" и "Ocenki".
* Заполняет объекты DataTable данными из выполненных запросов.
* Отображает данные из DataTable в DataGrid элементах пользовательского интерфейса.
* Обрабатывает исключения, выводя сообщение об ошибке в случае возникновения исключения.Закрывает соединение с базой данных после выполнения операций.

Этот код представляет собой инициализацию окна приложения, установку соединения с базой данных SQLite, выборку данных из таблиц и отображение их в пользовательском интерфейсе.

**Задание 4:** Разработайте и реализуйте редактирование данных в БД через интерфейс приложения:

**Выполнение:**

Результат работы программы представлен на рисунке 4.1. Код программы представлен в листинге 4.1.

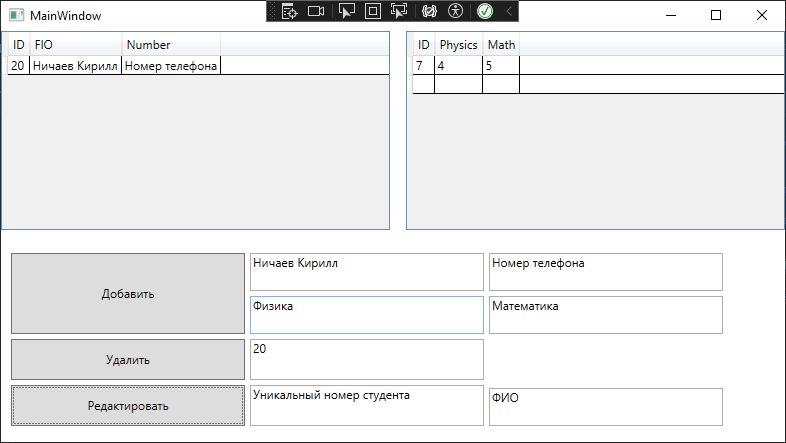


Рисунок 4.1 – Результат работы программы после нажатия кнопки редактировать

Листинг 4.1

C# код

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

{

string connectionString = "Data Source=C:\\Prog\\FIO.db;Version=3;";

myDataGrid.Items.Refresh();

oDataGrid.Items.Refresh();

try

{

using (var connection = new SQLiteConnection(connectionString))

{

connection.Open();

// Здесь вы можете выполнить SQL-запрос для обновления таблицы

string updateQuery = "UPDATE Students SET FIO = @Value1, Number = @Value2 WHERE ID = @StudentID";

using (SQLiteCommand command = new SQLiteCommand(updateQuery, connection))

{

command.Parameters.AddWithValue("@StudentID", ID.Text);

command.Parameters.AddWithValue("@Value1", FIO.Text);

command.Parameters.AddWithValue("@Value2", Number.Text);

command.ExecuteNonQuery();

}

string updateQueryO = "UPDATE Ocenki SET Physics = @Value3, Math = @Value4 WHERE ID = @StudentID";

using (SQLiteCommand command = new SQLiteCommand(updateQueryO, connection))

{

command.Parameters.AddWithValue("@StudentID", ID.Text);

command.Parameters.AddWithValue("@Value3", Phi.Text);

command.Parameters.AddWithValue("@Value4", Math.Text);

command.ExecuteNonQuery();

}

}

myDataGrid.Items.Refresh();

oDataGrid.Items.Refresh();

string selectQuery = "SELECT \* FROM Students"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQuery, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

myDataGrid.ItemsSource = dataTable.DefaultView;

}

string selectQueryO = "SELECT \* FROM Ocenki"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQueryO, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

oDataGrid.ItemsSource = dataTable.DefaultView;

}

}

catch (Exception ex)

{

// Обработка исключения, например, вывод сообщения об ошибке

MessageBox.Show("Произошла ошибка: " + ex.Message);

}

}

XAML разметка

<Button

Content="Редактировать" HorizontalAlignment="Left" Height="41" Margin="10,354,0,0" VerticalAlignment="Top" Width="234" Click="EditButton\_Click"/>

**Описание**

Данный код представляет метод EditButton\_Click, который выполняет следующие действия:

* Обновляет отображение данных в DataGrid элементах пользовательского интерфейса myDataGrid и oDataGrid с помощью метода Items.Refresh().
* Устанавливает соединение с базой данных SQLite, используя строку подключения "Data Source=C:\Prog\FIO.db;Version=3;".
* Выполняет SQL-запросы для обновления данных в таблицах "Students" и "Ocenki" на основе введенных значений.
* Обновляет отображение данных в DataGrid элементах пользовательского интерфейса myDataGrid и oDataGrid после выполнения операций обновления.
* Обрабатывает исключения, выводя сообщение об ошибке в случае возникновения исключения.

Этот метод предназначен для обработки нажатия кнопки "Edit" в пользовательском интерфейсе, и он выполняет важные операции по обновлению данных в базе данных и обновлению отображаемых данных в DataGrid.

**Задание 5:** Разработайте и реализуйте удаление данных из таблиц.

**Выполнение:**

Результат работы программы представлен на рисунке 5.1. Код программы представлен в листинге 5.1.

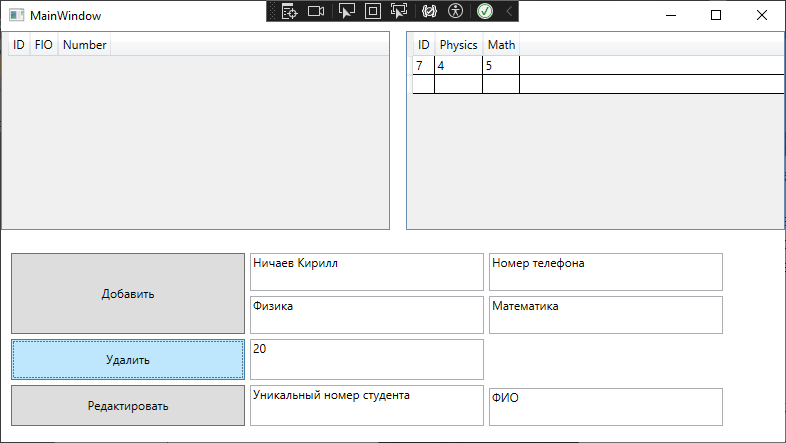


Рисунок 5.1 – Результат работы программы после ввода строки (Test5.txt)

Листинг 5.1

C# код

private void Button\_ClickDel(object sender, RoutedEventArgs e)

{

string connectionString = "Data Source=C:\\Prog\\FIO.db;Version=3;";

try

{

using (var connection = new SQLiteConnection(connectionString))

{

connection.Open();

string deleteQuery = "DELETE FROM Students WHERE ID = @StudentID";

string deleteQueryO = "DELETE FROM Ocenki WHERE ID = @StudentID";

using (SQLiteCommand command = new SQLiteCommand(deleteQuery, connection))

{

command.Parameters.AddWithValue("@StudentID", ID.Text);

command.ExecuteNonQuery();

}

using (SQLiteCommand command = new SQLiteCommand(deleteQueryO, connection))

{

command.Parameters.AddWithValue("@StudentID", ID.Text);

command.ExecuteNonQuery();

}

}

// После удаления данных можно обновить отображаемую таблицу

// Например, вызовите метод для обновления данных в вашем DataGrid

myDataGrid.Items.Refresh();

oDataGrid.Items.Refresh();

string selectQuery = "SELECT \* FROM Students"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQuery, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

myDataGrid.ItemsSource = dataTable.DefaultView;

}

string selectQueryO = "SELECT \* FROM Ocenki"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQueryO, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

oDataGrid.ItemsSource = dataTable.DefaultView;

}

}

catch (Exception ex)

{

// Обработка исключения, например, вывод сообщения об ошибке

MessageBox.Show("Произошла ошибка: " + ex.Message);

}

} XAML разметка

<Button

Content="Удалить" HorizontalAlignment="Left" Height="41" Margin="10,308,0,0" VerticalAlignment="Top" Width="234" Click="Button\_ClickDel"/>

**Описание**

Этот метод Button\_ClickDel представляет обработчик события нажатия кнопки "Delete" в пользовательском интерфейсе. Он выполняет следующие действия:

* Устанавливает соединение с базой данных SQLite, используя строку подключения "Data Source=C:\Prog\FIO.db;Version=3;".
* Выполняет SQL-запросы для удаления данных из таблиц "Students" и "Ocenki" на основе введенного идентификатора (ID).Обновляет отображение данных в DataGrid элементах пользовательского интерфейса myDataGrid и oDataGrid после выполнения операций удаления.
* Обрабатывает исключения, выводя сообщение об ошибке в случае возникновения исключения.

Этот метод предназначен для обработки нажатия кнопки "Delete" в пользовательском интерфейсе, и он выполняет важные операции по удалению данных из базы данных и обновлению отображаемых данных в DataGrid.

**Полный код WPF-приложения**

C# код

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;

using System.Data.SQLite;

using System.Data;

using System.Windows.Documents.DocumentStructures;

using System.Data.SqlClient;

using System.Data.Entity.ModelConfiguration.Configuration;

using System.Security.Permissions;

using System.Data.Common;

namespace LAB\_2

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

var connectionString = "Data Source=C:\\Prog\\FIO.db;Version=3;";

var connection = new SQLiteConnection(connectionString);

connection.Open();

// \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

string sql = "SELECT \* FROM Students";

DataTable dataTable = new DataTable();

string sqlo = "SELECT \* FROM Ocenki";

DataTable dataTableo = new DataTable();

try

{

connection = new SQLiteConnection(connectionString);

SQLiteCommand command = new SQLiteCommand(sql, connection);

SQLiteDataAdapter adapter = new SQLiteDataAdapter(command);

SQLiteCommand commando = new SQLiteCommand(sqlo, connection);

SQLiteDataAdapter adaptero = new SQLiteDataAdapter(commando);

connection.Open();

adapter.Fill(dataTable);

myDataGrid.ItemsSource = dataTable.DefaultView;

//connection.Open();

adaptero.Fill(dataTableo);

oDataGrid.ItemsSource = dataTableo.DefaultView;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

if (connection != null) connection.Close();

}

//---------------------------

myDataGrid.Items.Refresh();

oDataGrid.Items.Refresh();

// \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

connection.Close();

}

// --------------------------------------------------------------------------------------------

// Добавление

// --------------------------------------------------------------------------------------------

private void Button\_ClickADD(object sender, RoutedEventArgs e)

{

string connectionString = "Data Source=C:\\Prog\\FIO.db;Version=3;";

try

{

using (var connection = new SQLiteConnection(connectionString))

{

connection.Open();

string insertQuery = "INSERT INTO Students (FIO, Number) VALUES (@Value1, @Value2)";

string insertQueryO = "INSERT INTO Ocenki (Physics, Math) VALUES (@Value3, @Value4)";

using (SQLiteCommand command = new SQLiteCommand(insertQuery, connection))

{

command.Parameters.AddWithValue("@Value1", FIO.Text);

command.Parameters.AddWithValue("@Value2", Number.Text);

command.ExecuteNonQuery();

}

using (SQLiteCommand command = new SQLiteCommand(insertQueryO, connection))

{

command.Parameters.AddWithValue("@Value3", Phi.Text);

command.Parameters.AddWithValue("@Value4", Math.Text);

command.ExecuteNonQuery();

}

}

// После выполнения операции вставки, обновите данные в вашем DataGrid

string selectQuery = "SELECT \* FROM Students"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQuery, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

myDataGrid.ItemsSource = dataTable.DefaultView;

}

string selectQueryO = "SELECT \* FROM Ocenki"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQueryO, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

oDataGrid.ItemsSource = dataTable.DefaultView;

}

}

catch (Exception ex)

{

// Обработка исключения, например, вывод сообщения об ошибке

MessageBox.Show("Произошла ошибка: " + ex.Message);

}

}

// --------------------------------------------------------------------------------------------

// Удаление

// --------------------------------------------------------------------------------------------

private void Button\_ClickDel(object sender, RoutedEventArgs e)

{

string connectionString = "Data Source=C:\\Prog\\FIO.db;Version=3;";

try

{

using (var connection = new SQLiteConnection(connectionString))

{

connection.Open();

string deleteQuery = "DELETE FROM Students WHERE ID = @StudentID";

string deleteQueryO = "DELETE FROM Ocenki WHERE ID = @StudentID";

using (SQLiteCommand command = new SQLiteCommand(deleteQuery, connection))

{

command.Parameters.AddWithValue("@StudentID", ID.Text);

command.ExecuteNonQuery();

}

using (SQLiteCommand command = new SQLiteCommand(deleteQueryO, connection))

{

command.Parameters.AddWithValue("@StudentID", ID.Text);

command.ExecuteNonQuery();

}

}

// После удаления данных можно обновить отображаемую таблицу

// Например, вызовите метод для обновления данных в вашем DataGrid

myDataGrid.Items.Refresh();

oDataGrid.Items.Refresh();

string selectQuery = "SELECT \* FROM Students"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQuery, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

myDataGrid.ItemsSource = dataTable.DefaultView;

}

string selectQueryO = "SELECT \* FROM Ocenki"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQueryO, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

oDataGrid.ItemsSource = dataTable.DefaultView;

}

}

catch (Exception ex)

{

// Обработка исключения, например, вывод сообщения об ошибке

MessageBox.Show("Произошла ошибка: " + ex.Message);

}

}

// --------------------------------------------------------------------------------------------

// Изменение

// --------------------------------------------------------------------------------------------

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

{

string connectionString = "Data Source=C:\\Prog\\FIO.db;Version=3;";

myDataGrid.Items.Refresh();

oDataGrid.Items.Refresh();

try

{

using (var connection = new SQLiteConnection(connectionString))

{

connection.Open();

// Здесь вы можете выполнить SQL-запрос для обновления таблицы

string updateQuery = "UPDATE Students SET FIO = @Value1, Number = @Value2 WHERE ID = @StudentID";

using (SQLiteCommand command = new SQLiteCommand(updateQuery, connection))

{

command.Parameters.AddWithValue("@StudentID", ID.Text);

command.Parameters.AddWithValue("@Value1", FIO.Text);

command.Parameters.AddWithValue("@Value2", Number.Text);

command.ExecuteNonQuery();

}

string updateQueryO = "UPDATE Ocenki SET Physics = @Value3, Math = @Value4 WHERE ID = @StudentID";

using (SQLiteCommand command = new SQLiteCommand(updateQueryO, connection))

{

command.Parameters.AddWithValue("@StudentID", ID.Text);

command.Parameters.AddWithValue("@Value3", Phi.Text);

command.Parameters.AddWithValue("@Value4", Math.Text);

command.ExecuteNonQuery();

}

}

myDataGrid.Items.Refresh();

oDataGrid.Items.Refresh();

string selectQuery = "SELECT \* FROM Students"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQuery, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

myDataGrid.ItemsSource = dataTable.DefaultView;

}

string selectQueryO = "SELECT \* FROM Ocenki"; // Замените на ваш запрос выборки

using (SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(selectQueryO, connectionString))

{

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

oDataGrid.ItemsSource = dataTable.DefaultView;

}

}

catch (Exception ex)

{

// Обработка исключения, например, вывод сообщения об ошибке

MessageBox.Show("Произошла ошибка: " + ex.Message);

}

}

}

}

XAML разметка

<Window x:Class="LAB\_2.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:LAB\_2"

mc:Ignorable="d"

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

<Grid>

<Button

Content="Добавить" HorizontalAlignment="Left" Height="81" Margin="10,222,0,0" VerticalAlignment="Top" Width="234" Click="Button\_ClickADD"/>

<DataGrid

x:Name="myDataGrid"

Margin="0,0,395,212"

AutoGenerateColumns="True"

CanUserAddRows="False" CanUserDeleteRows="False"

CanUserResizeRows="False" CanUserResizeColumns="False" CanUserReorderColumns="False"

IsReadOnly="False"

/>

<DataGrid

x:Name="oDataGrid"

Margin="405,0,0,212"

AutoGenerateColumns="True"

/>

<TextBox x:Name="FIO"

HorizontalAlignment="Left" Height="38" Margin="249,222,0,0" TextWrapping="Wrap" Text="ФИО" VerticalAlignment="Top" Width="234"/>

<TextBox x:Name="Number"

HorizontalAlignment="Left" Height="38" Margin="488,222,0,0" TextWrapping="Wrap" Text="Номер телефона" VerticalAlignment="Top" Width="234"/>

<TextBox x:Name="Phi"

HorizontalAlignment="Left" Height="38" Margin="249,265,0,0" TextWrapping="Wrap" Text="Физика" VerticalAlignment="Top" Width="234" />

<TextBox x:Name="Math"

HorizontalAlignment="Left" Height="38" Margin="488,265,0,0" TextWrapping="Wrap" Text="Математика" VerticalAlignment="Top" Width="234"/>

<Button

Content="Удалить" HorizontalAlignment="Left" Height="41" Margin="10,308,0,0" VerticalAlignment="Top" Width="234" Click="Button\_ClickDel"/>

<TextBox x:Name="ID"

HorizontalAlignment="Left" Height="41" Margin="249,308,0,0" TextWrapping="Wrap" Text="Уникальный номер студента" VerticalAlignment="Top" Width="234" />

<Button

Content="Редактировать" HorizontalAlignment="Left" Height="41" Margin="10,354,0,0" VerticalAlignment="Top" Width="234" Click="EditButton\_Click"/>

<TextBox x:Name="IDR"

HorizontalAlignment="Left" Height="41" Margin="249,354,0,0" TextWrapping="Wrap" Text="Уникальный номер студента" VerticalAlignment="Top" Width="234" />

<TextBox x:Name="FIOR"

HorizontalAlignment="Left" Height="38" Margin="488,357,0,0" TextWrapping="Wrap" Text="ФИО" VerticalAlignment="Top" Width="234"/>

</Grid>

</Window>