**31 C# и MS SQL Server**

Задание 1. Выполните этапы создания приложения используя главу 3). Выполните подключения к базе данных использую DataAdapter для вашей СУБД.

Листинг программы:

using System;

using System.Windows.Forms;

namespace Task1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "dBTur\_firmDataSet.Информация\_о\_туристах". При необходимости она может быть перемещена или удалена.

this.информация\_о\_туристахTableAdapter.Fill(this.dBTur\_firmDataSet.Информация\_о\_туристах);

// TODO: данная строка кода позволяет загрузить данные в таблицу "dBTur\_firmDataSet.Туристы". При необходимости она может быть перемещена или удалена.

this.туристыTableAdapter.Fill(this.dBTur\_firmDataSet.Туристы);

}

}

}

Таблица 31.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

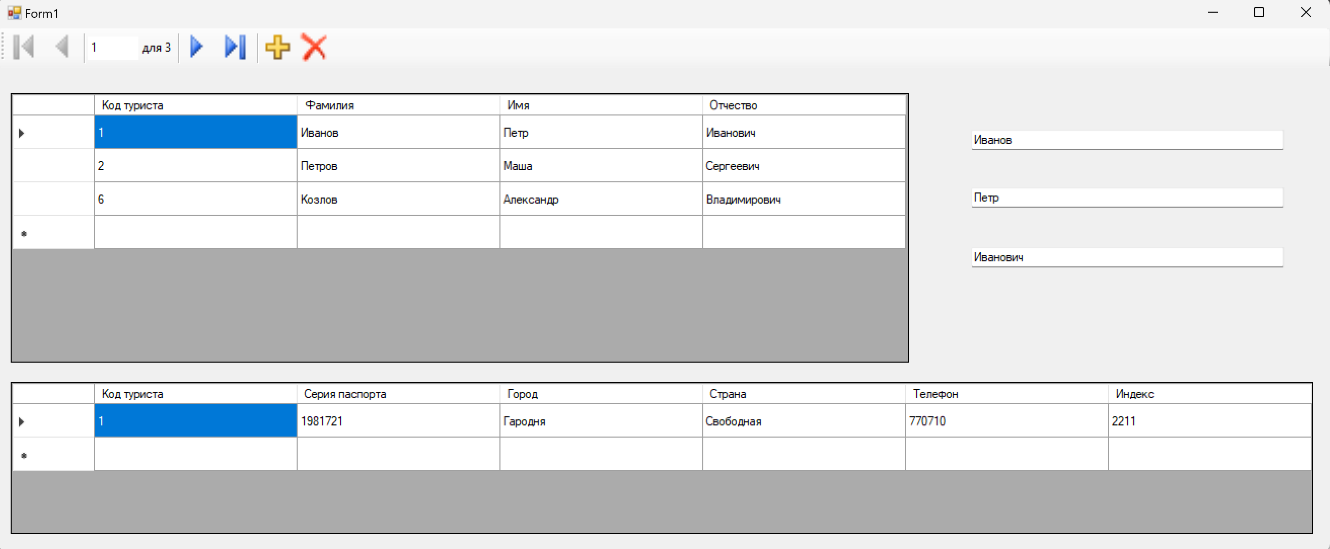


Рисунок 31.1 – Результат работы программы

Задание 2. Изучите главу 4 пункт 4.3, используя приложение из задания №1 создайте и выполните SQL запросы: выберите все туры; удалите тур; добавьте туриста; измените туриста.

Листинг программы:

using System;

using System.Data;

using System.Windows.Forms;

using System.Data.SQLite;

namespace Task2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

LoadTourists();

}

public SQLiteConnection CreateConnection()

{

string connectionString = "Data Source=../../../DBTur\_firm.db;Version=3;";

SQLiteConnection conn = new SQLiteConnection(connectionString);

return conn;

}

private void LoadTourists()

{

using (SQLiteConnection conn = CreateConnection())

{

conn.Open();

string query = "SELECT \* FROM Туристы";

SQLiteDataAdapter da = new SQLiteDataAdapter(query, conn);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

textBox1.DataBindings.Clear();

textBox2.DataBindings.Clear();

textBox3.DataBindings.Clear();

textBox1.DataBindings.Add("Text", dt, "Имя");

textBox2.DataBindings.Add("Text", dt, "Фамилия");

textBox3.DataBindings.Add("Text", dt, "Отчество");

}

}

private void button1\_Click(object sender, EventArgs e)

{

using (SQLiteConnection conn = CreateConnection())

{

conn.Open();

string query = "SELECT \* FROM Туры";

SQLiteDataAdapter da = new SQLiteDataAdapter(query, conn);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (dataGridView2.DataSource == null)

{

MessageBox.Show("Сначала добавьте данные во второй DataGridView");

return;

}

using (DeleteTourForm deleteForm = new DeleteTourForm())

{

if (deleteForm.ShowDialog() == DialogResult.OK)

{

string tourCode = deleteForm.TourCode;

if (string.IsNullOrWhiteSpace(tourCode))

{

MessageBox.Show("Код тура не может быть пустым");

return;

}

using (SQLiteConnection conn = CreateConnection())

{

conn.Open();

string query = "DELETE FROM Туры WHERE Код\_тура = @Код\_тура";

using (SQLiteCommand cmd = new SQLiteCommand(query, conn))

{

cmd.Parameters.AddWithValue("@Код\_тура", tourCode);

int rowsAffected = cmd.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Тур успешно удален");

}

else

{

MessageBox.Show("Тур с указанным кодом не найден");

}

}

}

button1\_Click(sender, e);

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

using (AddTouristForm addForm = new AddTouristForm())

{

if (addForm.ShowDialog() == DialogResult.OK)

{

string firstName = addForm.FirstName;

string lastName = addForm.LastName;

string middleName = addForm.MiddleName;

if (string.IsNullOrWhiteSpace(firstName) || string.IsNullOrWhiteSpace(lastName) || string.IsNullOrWhiteSpace(middleName))

{

MessageBox.Show("Все поля должны быть заполнены");

return;

}

using (SQLiteConnection conn = CreateConnection())

{

conn.Open();

string query = "INSERT INTO Туристы (Имя, Фамилия, Отчество) VALUES (@Имя, @Фамилия, @Отчество)";

using (SQLiteCommand cmd = new SQLiteCommand(query, conn))

{

cmd.Parameters.AddWithValue("@Имя", firstName);

cmd.Parameters.AddWithValue("@Фамилия", lastName);

cmd.Parameters.AddWithValue("@Отчество", middleName);

cmd.ExecuteNonQuery();

}

}

LoadTourists();

}

}

}

private void button4\_Click(object sender, EventArgs e)

{

using (EditTouristForm updateForm = new EditTouristForm())

{

if (updateForm.ShowDialog() == DialogResult.OK)

{

string touristCode = updateForm.TouristCode;

string firstName = updateForm.FirstName;

string lastName = updateForm.LastName;

string middleName = updateForm.MiddleName;

if (string.IsNullOrWhiteSpace(touristCode) || string.IsNullOrWhiteSpace(firstName) || string.IsNullOrWhiteSpace(lastName) || string.IsNullOrWhiteSpace(middleName))

{

MessageBox.Show("Все поля должны быть заполнены");

return;

}

using (SQLiteConnection conn = CreateConnection())

{

conn.Open();

string query = "UPDATE Туристы SET Имя = @Имя, Фамилия = @Фамилия, Отчество = @Отчество WHERE Код\_туриста = @Код\_туриста";

using (SQLiteCommand cmd = new SQLiteCommand(query, conn))

{

cmd.Parameters.AddWithValue("@Код\_туриста", touristCode);

cmd.Parameters.AddWithValue("@Имя", firstName);

cmd.Parameters.AddWithValue("@Фамилия", lastName);

cmd.Parameters.AddWithValue("@Отчество", middleName);

int rowsAffected = cmd.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Данные туриста успешно обновлены");

}

else

{

MessageBox.Show("Турист с указанным кодом не найден");

}

}

}

LoadTourists();

}

}

}

}

}

Таблица 31.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

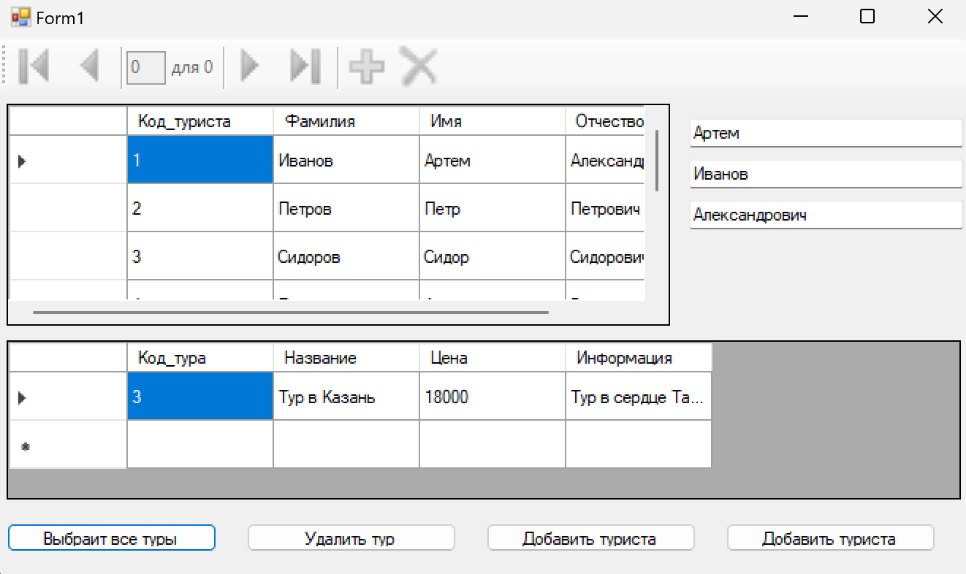


Рисунок 31.2 – Результат работы программы