**30 Работа с базами данных. Технология ADO.NET**

Задание №1. Создайте .Net приложение (консольное WinForm, WPF любое на выбор) и выполните подключение к разработанной базе данных, для этого изучите материал главы 4, пункт 4.1. Создайте и выполните SQL запросы: Выберите все туры, удалите тур, добавьте туриста, измените туриста.

Листинг программы:

using System;

using System.Data;

using System.Data.SQLite;

using System.Windows.Forms;

namespace Task1

{

public partial class Form1 : Form

{

private string connectionString = @"D:\!!!!!!!ПРАКТИКА\Виноградова\_день31\Практическая часть31\DBTur\_firm.sqlite";

private DataGridView dataGridView;

private TextBox textBoxTourId, textBoxTouristId, textBoxLastName, textBoxFirstName, textBoxMiddleName;

public Form1()

{

InitializeComponent();

InitializeUI();

}

private void InitializeUI()

{

string[] buttonNames = { "Загрузить туры", "Удалить тур", "Добавить туриста", "Изменить туриста", "Загрузить туристов", "Удалить туриста" };

EventHandler[] handlers = { LoadTours, DeleteTour, AddTourist, UpdateTourist, LoadTourists, DeleteTourist };

for (int i = 0; i < buttonNames.Length; i++)

{

Button btn = new Button { Text = buttonNames[i], Left = 10, Top = 10 + i \* 40, Width = 170, Height = 30 };

btn.Click += handlers[i];

this.Controls.Add(btn);

}

dataGridView = new DataGridView { Left = 200, Top = 10, Width = 600, Height = 430 };

this.Controls.Add(dataGridView);

string[] textBoxHints = { "введите id тура", "введите id туриста", "введите фамилия", "введите имя", "введите отчество" };

TextBox[] textBoxes = { textBoxTourId, textBoxTouristId, textBoxLastName, textBoxFirstName, textBoxMiddleName };

for (int i = 0; i < textBoxHints.Length; i++)

{

textBoxes[i] = new TextBox { Left = 10, Top = 250 + i \* 40, Width = 170, Height = 30, Text = textBoxHints[i] };

textBoxes[i].GotFocus += (s, e) => { if (((TextBox)s).Text == textBoxHints[i]) ((TextBox)s).Text = ""; };

textBoxes[i].LostFocus += (s, e) => { if (((TextBox)s).Text == "") ((TextBox)s).Text = textBoxHints[i]; };

this.Controls.Add(textBoxes[i]);

}

}

private void ExecuteQuery(string query, params SQLiteParameter[] parameters)

{

using (SQLiteConnection conn = new SQLiteConnection(connectionString))

{

SQLiteCommand cmd = new SQLiteCommand(query, conn);

cmd.Parameters.AddRange(parameters);

conn.Open();

cmd.ExecuteNonQuery();

}

}

private void LoadTours(object sender, EventArgs e)

{

LoadData("SELECT \* FROM Туры");

}

private void LoadTourists(object sender, EventArgs e)

{

LoadData("SELECT \* FROM Туристы");

}

private void LoadData(string query)

{

using (SQLiteConnection conn = new SQLiteConnection(connectionString))

{

SQLiteDataAdapter adapter = new SQLiteDataAdapter(query, conn);

DataTable dt = new DataTable();

adapter.Fill(dt);

dataGridView.DataSource = dt;

}

}

private void DeleteTour(object sender, EventArgs e)

{

if (int.TryParse(textBoxTourId.Text, out int id)) ExecuteQuery("DELETE FROM Туры WHERE Код\_тура = @id", new SQLiteParameter("@id", id));

}

private void DeleteTourist(object sender, EventArgs e)

{

if (int.TryParse(textBoxTouristId.Text, out int id)) ExecuteQuery("DELETE FROM Туристы WHERE Код\_туриста = @id", new SQLiteParameter("@id", id));

}

private void AddTourist(object sender, EventArgs e)

{

ExecuteQuery("INSERT INTO Туристы (Фамилия, Имя, Отчество) VALUES (@ln, @fn, @mn)",

new SQLiteParameter("@ln", textBoxLastName.Text), new SQLiteParameter("@fn", textBoxFirstName.Text), new SQLiteParameter("@mn", textBoxMiddleName.Text));

}

private void UpdateTourist(object sender, EventArgs e)

{

if (int.TryParse(textBoxTouristId.Text, out int id))

{

ExecuteQuery("UPDATE Туристы SET Фамилия = @ln, Имя = @fn, Отчество = @mn WHERE Код\_туриста = @id",

new SQLiteParameter("@id", id), new SQLiteParameter("@ln", textBoxLastName.Text), new SQLiteParameter("@fn", textBoxFirstName.Text), new SQLiteParameter("@mn", textBoxMiddleName.Text));

}

}

}

}

Анализ результатов:

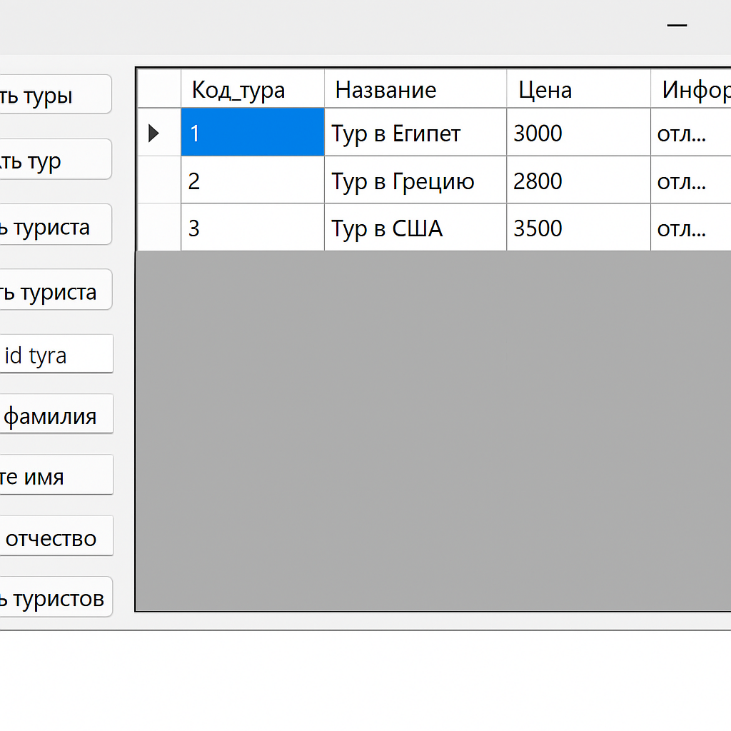


Рисунок 30.1 – Результаты работы программы