**32 Работа с SqlDataAdapter и DataSet**

Задание №1. Изучите главу 4 пункт 4.4, используя приложение из задания №1/ Свяжите данные с помощью DataSet, предварительно создайте DataTable для каждой из таблиц БД.

Листинг программы:

using System;

using System.Data;

using System.Data.SQLite;

using System.Linq;

using System.Windows.Forms;

namespace Task

{

public partial class Form1 : Form

{

private const string ConnStr = "Data Source=C:\\sqlite\\DBTur\_firm.sqlite;Version=3;";

private readonly DataSet dataSet = new DataSet();

private readonly DataTable tours = new DataTable("Туры"), tourists = new DataTable("Туристы");

public Form1()

{

InitializeComponent();

InitDataTables();

InitControls();

}

private void InitDataTables()

{

tours.Columns.AddRange(new[] {

new DataColumn("Код\_тура", typeof(int)),

new DataColumn("Название", typeof(string)),

new DataColumn("Дата\_начала", typeof(DateTime)),

new DataColumn("Дата\_окончания", typeof(DateTime)),

});

tourists.Columns.AddRange(new[] {

new DataColumn("Код\_туриста", typeof(int)),

new DataColumn("Фамилия", typeof(string)),

new DataColumn("Имя", typeof(string)),

new DataColumn("Код\_тура", typeof(int))

});

tours.PrimaryKey = new[] { tours.Columns["Код\_тура"] };

tourists.PrimaryKey = new[] { tourists.Columns["Код\_туриста"] };

dataSet.Tables.AddRange(new[] { tours, tourists });

}

private void InitControls()

{

var dgvTours = new DataGridView { Left = 200, Top = 10, Width = 620, Height = 300, DataSource = tours };

var dgvTourists = new DataGridView { Left = 200, Top = 320, Width = 620, Height = 150, DataSource = tourists };

string[] btnTexts = { "Загрузить туры", "Удалить тур", "Добавить туриста",

"Изменить туриста", "Загрузить туристов", "Удалить туриста", "Сохранить изменения" };

int[] tops = { 10, 50, 90, 130, 370, 410, 450 };

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

{

var btn = new Button { Text = btnTexts[i], Left = 10, Top = tops[i], Width = 170, Height = 30 };

btn.Click += (s, e) => HandleButtonClick(btn.Text);

}

string[] placeholders = { "введите id тура", "введите фамилия", "введите имя",

"введите отчество", "введите id туриста" };

int[] textBoxTops = { 170, 210, 250, 290, 330 };

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

{

var tb = new TextBox { Text = placeholders[i], Left = 10, Top = textBoxTops[i], Width = 170, Height = 30 };

tb.GotFocus += (s, e) => { if (tb.Text == placeholders[i]) tb.Text = ""; };

tb.LostFocus += (s, e) => { if (string.IsNullOrEmpty(tb.Text)) tb.Text = placeholders[i]; };

Controls.Add(tb);

}

Controls.AddRange(new Control[] { dgvTours, dgvTourists });

}

private void HandleButtonClick(string action)

{

var tbs = Controls.OfType<TextBox>().ToArray();

try

{

using (var conn = new SQLiteConnection(ConnStr))

{

conn.Open();

switch (action)

{

case "Загрузить туры":

new SQLiteDataAdapter("SELECT \* FROM Туры", conn).Fill(tours); break;

case "Загрузить туристов":

new SQLiteDataAdapter("SELECT \* FROM Туристы", conn).Fill(tourists); break;

case "Удалить тур":

if (int.TryParse(tbs[0].Text, out int tourId))

new SQLiteCommand("DELETE FROM Туры WHERE Код\_тура = @id", conn)

.AddParam("@id", tourId).ExecuteNonQuery(); break;

case "Удалить туриста":

if (int.TryParse(tbs[4].Text, out int touristId))

new SQLiteCommand("DELETE FROM Туристы WHERE Код\_туриста = @id", conn)

.AddParam("@id", touristId).ExecuteNonQuery(); break;

case "Добавить туриста":

if (int.TryParse(tbs[0].Text, out int tId))

new SQLiteCommand("INSERT INTO Туристы (Фамилия, Имя, Отчество, Код\_тура) VALUES (@l,@f,@m,@t)", conn)

.AddParams(tbs[1].Text, tbs[2].Text, tbs[3].Text, tId).ExecuteNonQuery(); break;

case "Изменить туриста":

if (int.TryParse(tbs[4].Text, out int tid) && int.TryParse(tbs[0].Text, out int ttid))

new SQLiteCommand("UPDATE Туристы SET Фамилия=@l,Имя=@f,Отчество=@m,Код\_тура=@t WHERE Код\_туриста=@id", conn)

.AddParams(tbs[1].Text, tbs[2].Text, tbs[3].Text, ttid).AddParam("@id", tid).ExecuteNonQuery();

break;

case "Сохранить изменения":

new SQLiteDataAdapter("SELECT \* FROM Туры", conn).Update(tours);

new SQLiteDataAdapter("SELECT \* FROM Туристы", conn).Update(tourists);

MessageBox.Show("Данные сохранены!");

break;

}

}

if (action.StartsWith("Удалить") || action.StartsWith("Добавить") || action.StartsWith("Изменить"))

HandleButtonClick(action.Contains("тур") ? "Загрузить туры" : "Загрузить туристов");

}

static class Extensions

{

public static SQLiteCommand AddParam(this SQLiteCommand cmd, string name, object value)

{

cmd.Parameters.AddWithValue(name, value);

return cmd;

}

public static SQLiteCommand AddParams(this SQLiteCommand cmd, params object[] values)

{

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

cmd.Parameters.AddWithValue($"@{Convert.ToChar(108 + i)}", values[i]);

return cmd;

}

}

}

Анализ результатов:

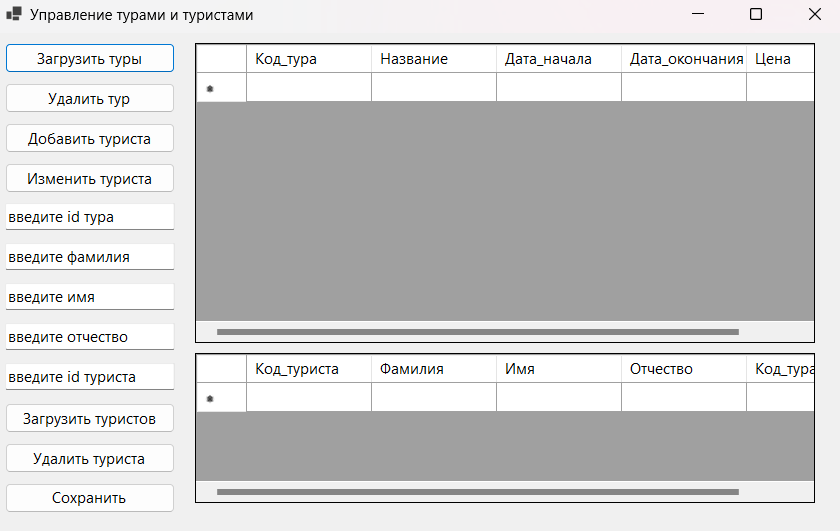


Рисунок 32.1 – Результаты работы программы