**33 Технология LINQ TO SQL.**

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

Листинг программы:

private void Delete()

{

try

{

var currentCeil = TouristDataGridView.SelectedCells;

int touristId = (int)TouristDataGridView[0, currentCeil[0].RowIndex].Value;

var deleteTourists = from tourists in \_dB.GetTable<Turist>()

where tourists.Id == touristId

select tourists;

MessageBox.Show("Запись успешно удалена", "Успех");

foreach (var tourist in deleteTourists)

{

\_dB.GetTable<Turist>().DeleteOnSubmit(tourist);

}

\_dB.SubmitChanges();

}

catch

{

MessageBox.Show("Операция удаления не была завершена", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

Delete();

}

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

{

try

{

if ((TouristSurnameTextBox.Text == string.Empty) && (TouristNameTextBox.Text == string.Empty) &&

(TouristPatronymicTextBox.Text == string.Empty))

{

MessageBox.Show("Заполните все поля", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

Turist tourist = new Turist

{

Surname = TouristSurnameTextBox.Text,

Name = TouristNameTextBox.Text,

Patronymic = TouristPatronymicTextBox.Text

};

\_dB.GetTable<Turist>().InsertOnSubmit(tourist);

\_dB.SubmitChanges();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

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

{

var values = from tourists in \_dB.GetTable<Turist>()

select tourists;

var grid = new DataGrid()

{

DataSource = values,

Width = 400,

Height = 200,

};

panel1.Controls.Add(grid)

Таблица 1.1 – Выходные и входные данные

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

Анализ результатов:

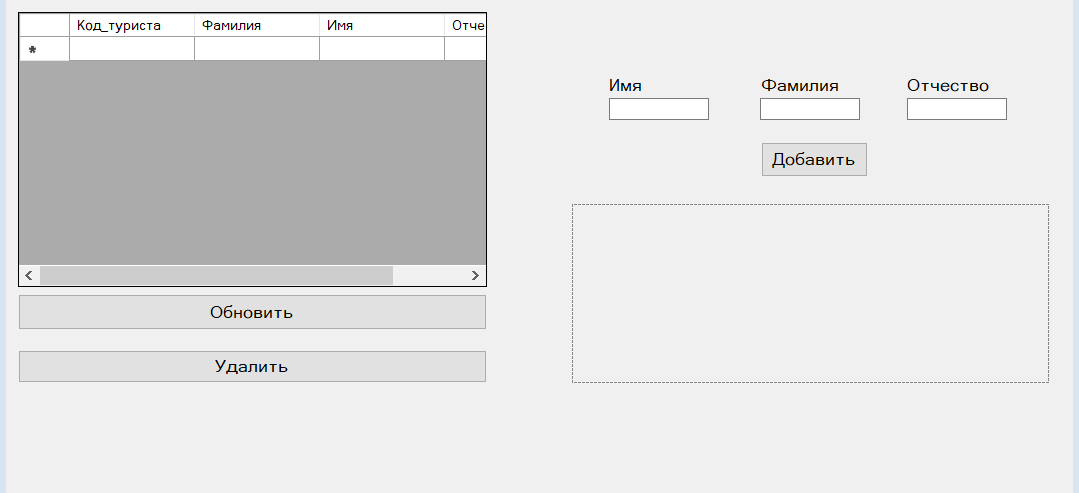


Рисунок 1.1 – Результат работы программы

Источник: собственная разработка