

### 33 ТЕХНОЛОГИЯ LINQ TO SQL

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:

```
public partial class Form1 : Form
{
    static private string _dbPath = @"G:\Колледж\Git\C-
practice\33\33\z1\Data\DBTur_firm.mdf";
    static private string _connectionString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename={_dbPath}";
    private DataContext _dB = new DataContext(_connectionString);
    public Form1()
    {
        InitializeComponent();
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        this.туристыTableAdapter.Fill(this.dBTur_firmDataSet.Туристы);
    }
    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)
            {
```

					УП 2-40 01 01.31 ТП.2471.22.33		
Изм.	Лист	Поддокум	Подп.	Дата			
Разраб.	Мушинский М.С				ТЕХНОЛОГИЯ LINQ TO SQL	Лист	Лист
Пров.	Толочко П.С.						
Н.контр.						Гродненский ГКТТид	
Утв.							

```

        _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());
    }
}

```

					УП 2-40 01 01.31 ТП.2471.22.33	Лист
Изм.	Лист	№ докум.	Подп.	Дата		

```

    }
}
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 33.1 – Входные и выходные данные

Входные данные	Выходные данные
surname, name, patronymic	DataGridView1, DataGridView2

Анализ результатов:

The screenshot displays a software interface for managing tourist data. On the left, a table lists existing tourists with columns for 'Код\_туриста' (Tourist Code), 'Фамилия' (Surname), and 'Имя' (Name). The first row shows a tourist with code '1', surname '1', and name '123'. Below the table are buttons for 'Обновить' (Update) and 'Удалить' (Delete). On the right, there are input fields for 'Имя' (Name), 'Фамилия' (Surname), and 'Отчество' (Patronymic), followed by a 'Добавить' (Add) button. Below these fields is another table, likely for the newly added or selected tourist, with columns 'Id', 'Surname', 'Name', and 'Patronymic'. This second table shows the same data as the first table: Id '1', Surname '1', Name '123', and Patronymic '3'.

Рисунок 33.1 – Результат работы программы  
Источник: собственная разработка