**Описание класса AccessDataBase-для подключения к базе MyDB в Access**

using System;

using System.Data.OleDb;

using System.Windows.Forms;

namespace ACCESS

{

class AccessDataBase

{

private OleDbConnection myConnection

public static string connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=акссес достопримечательности.mdb";

public bool Connect()

{

try

{

myConnection = new OleDbConnection(connectString);

myConnection.Open();

return true;

}

catch (Exception e)

{

MessageBox.Show(e.Message);

return false;

}

finally

{

myConnection.Close();

}

}

public void execQueryReader(String[] \_cellsNames, String \_que, DataGridView \_dgv)

{

int j = -1;

int countCells;

\_dgv.RowCount = 1;

\_dgv.ColumnCount = 0;

try

{

OleDbCommand command = new OleDbCommand(\_que, myConnection);

myConnection.Open();

OleDbDataReader dr = command.ExecuteReader();

countCells = dr.FieldCount;

\_dgv.ColumnCount = countCells;

for (int i = 0; i < countCells; i++)

\_dgv.Columns[i].HeaderText = \_cellsNames[i];

while (dr.Read())

{

j += 1;

\_dgv.RowCount++;

for (int i = 0; i < countCells; i++)

\_dgv.Rows[j].Cells[i].Value = dr[i];

}

}

catch (OleDbException e)

{

MessageBox.Show(e.Message);

}

finally

{

myConnection.Close();

}

}

public void execQueryReader(String \_que, DataGridView \_dgv)

{

int j = -1;

int countCells;

\_dgv.RowCount = 1;

\_dgv.ColumnCount = 0;

try

{

OleDbCommand command = new OleDbCommand(\_que, myConnection);

OleDbDataReader dr = command.ExecuteReader();

countCells = dr.FieldCount;

\_dgv.ColumnCount = countCells;

while (dr.Read())

{

j += 1;

\_dgv.RowCount++;

for (int i = 0; i < countCells; i++)

\_dgv.Rows[j].Cells[i].Value = dr[i];

}

}

catch (OleDbException e)

{

MessageBox.Show(e.Message);

}

finally

{

myConnection.Close();

}

}

public void disCon()

{

if (myConnection != null)

myConnection.Close();

}

}

}

Описание класса Zapros

using System;

using System.Windows.Forms;

namespace ACCESS

{

public partial class Zapros : Form

{

AccessDataBase db = new AccessDataBase();

public bool conStat;

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

{

this.Мемориальные\_доскиTableAdapter.Fill(this.акссес\_достопримечательностиDataSet. Мемориальные\_доски);

this.ДостопримечательностьTableAdapter.Fill(this.акссес\_достопримечательностиDataSet. Достопримечательность);

}

public Zapros()

{

InitializeComponent();

conStat = db.Connect();

}

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

{

if (!conStat)

{

Application.Exit();

}

}

private void Main\_FormClosing(object sender, FormClosingEventArgs e)

{

if (!conStat)

db.disCon();

}

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

{

db.execQueryReader(tbZapros.Text, dgZapros);

}

private void tsMenuZapros1\_Click(object sender, EventArgs e)

{

db.execQueryReader(new string[] { "Достопримечательность", "Табличка", "Год открытия" }, "SELECT [Достопримечательность], [Табличка], [Год открытия] FROM [Мемориальные доски] WHERE ((([Мемориальные доски].[Год открытия])<'1700'));", dgZapros);

}

private void tsMenuZapros2\_Click(object sender, EventArgs e)

{

db.execQueryReader(new string[] { "Достопримечательность", "Табличка" }, "SELECT [Достопримечательность], [Табличка] FROM [Мемориальные доски] WHERE ((([Мемориальные доски].[Местарасположение])='Смоленск'));", dgZapros);

}

private void tsMenuZapros3\_Click(object sender, EventArgs e)

{

db.execQueryReader(new string[] { "Фото", "Месторасположение", "Описание" }, "SELECT [Фото], [Месторасположение], [Описание] FROM [Табличка] WHERE (((Год открытия)<'125'));", dgZapros);

}

private void tsMenuZapros4\_Click(object sender, EventArgs e)

{

db.execQueryReader(new string[] { "Фото", "Местарасположение", "Описание" }, "SELECT [Фото], [Месторасположение], [Описание] FROM [Табличка] WHERE (((Табличка.Фото)='Name'));", dgZapros);

}

private void tsMenuZapros5\_Click(object sender, EventArgs e)

{

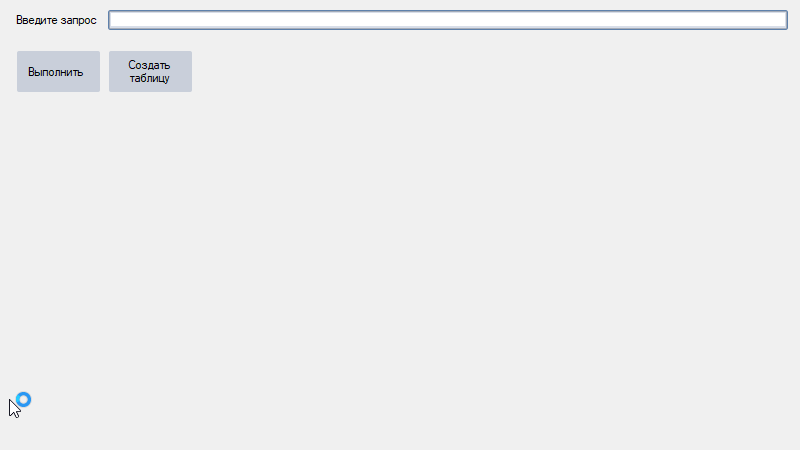
db.execQueryReader(new string[] { "","Описание и фото", " Местоположение", "Год открытия" }, "SELECT [Достопримечательность], [Описание и фото],[Местоположение], [Год открытия] FROM [Мемориальные таблички] WHERE ((([Мемориальные таблички].[Год открытия])>'1900'));", dgZapros);

}

}

}

Вид формы



**НОВЫЙ ЗАПРОС НА ВЫБОРКУ:** SELECT [Код], [Достопримечательность], [Город(посёлок)], [Год постройки] FROM [Храмы и церкви смоленской области] WHERE ((([Храмы и церкви смоленской области]. [Город(посёлок)])="Смоленск") AND (([Храмы и церкви смоленской области].[Год постройки])<'1700'));

**НОВЫЙ ЗАПРОС НА ВЫБОРКУ:** SELECT [Код], [Достопримечательность], [Описание и фото], [Город(посёлок)], [Точное местоположение], [Год постройки] FROM [Храмы и церкви смоленской области] WHERE ((([Храмы и церкви смоленской области].Достопримечательность) Like "церковь\*"));

**НОВЫЙ ПЕРЕКРЕСТНЫЙ ЗАПРОС:** TRANSFORM Max([Год постройки]) AS [Count-Год постройки] SELECT Достопримечательность, Max([Год постройки]) AS [Итоговое значение Год постройки] FROM [Храмы и церкви смоленской области] WHERE ((([Город(посёлок)]) Like 'с\*')) GROUP BY Достопримечательность PIVOT [Город(посёлок)];

