# Программа

using System;

using DataBaseWorkingProj06;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Configuration;

using System.Windows.Forms;

namespace DataBaseWorkingProj06

{

public partial class Form1 : Form

{

Ext Class = new Ext();

public Form1()

{

InitializeComponent();

sw\_action.Click += (s, e) => { pnl\_Action.BringToFront(); };

sw\_lookat.Click += (s, e) => { pnl\_lookAt.BringToFront(); };

}

void Message(string line)

{

MessageBox.Show(line, "Уведомление!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

void Refresh(DataGridView DataGrid, string line)

{

DataTable Data = Class.Select(line);

DataGrid.DataSource = Data;

}

void ClearRows()

{

txt\_id.Text = "";

txt\_name.Text = "";

txt\_lastname.Text = "";

txt\_YaerOfBirth.Text = "";

cmb\_Gender.Text = "";

txt\_Contact.Text = "";

textBox1.Text = "";

textBox2.Text = "";

}

void ComboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

switch (comboBox1.Text)

{

case "Директор":

textBox1.Enabled = false;

textBox2.Enabled = false;

Refresh(dataGridView1, "Директор");

break;

case "Кладовщик":

textBox1.Enabled = true;

textBox2.Enabled = true;

Refresh(dataGridView1, "Кладовщик");

break;

}

}

bool Success = false;

void Btn\_Add\_Click(object sender, EventArgs e)

{

try

{

switch (comboBox1.Text)

{

case "Директор":

Class.Имя = txt\_name.Text;

Class.Фамилия = txt\_lastname.Text;

Class.Год\_Рождения = txt\_YaerOfBirth.Text;

Class.Пол = cmb\_Gender.Text;

Class.Контакты = txt\_Contact.Text;

Success = Class.Insert(Class, "Директор");

Message(Success ? "Добавление прошло успешно!" : "При добалении произошла ошибка!");

Refresh(dataGridView1, "Директор");

ClearRows();

break;

case "Кладовщик":

Class.Имя = txt\_name.Text;

Class.Фамилия = txt\_lastname.Text;

Class.Год\_Рождения = txt\_YaerOfBirth.Text;

Class.Пол = cmb\_Gender.Text;

Class.Контакты = txt\_Contact.Text;

// Class.Опыт\_работы = textBox1.Text;

//Class.Желаема\_Зарплата = textBox2.Text;

Success = Class.Insert(Class, "Кладовщик");

Message(Success ? "Добавление прошло успешно!" : "При добалении произошла ошибка!");

Refresh(dataGridView1, "Кладовщик");

ClearRows();

break;

}

}

catch(Exception ex) { MessageBox.Show(ex.Message, ex.Source, MessageBoxButtons.OK, MessageBoxIcon.Error); }

}

void Btn\_Edit\_Click(object sender, EventArgs e)

{

try

{

switch (comboBox1.Text)

{

case "Директор":

Class.id = int.Parse(txt\_id.Text);

Class.Имя = txt\_name.Text;

Class.Фамилия = txt\_lastname.Text;

Class.Год\_Рождения = txt\_YaerOfBirth.Text;

Class.Пол = cmb\_Gender.Text;

Class.Контакты = txt\_Contact.Text;

Success = Class.Update(Class, "Директор");

Message(Success ? "Обновление данных прошла успешно!" : "Обновление данных прошла не удачно!");

Refresh(dataGridView1, "Директор");

ClearRows();

break;

case "Кладовщик":

Class.id = int.Parse(txt\_id.Text);

Class.Имя = txt\_name.Text;

Class.Фамилия = txt\_lastname.Text;

Class.Год\_Рождения = txt\_YaerOfBirth.Text;

Class.Пол = cmb\_Gender.Text;

Class.Контакты = txt\_Contact.Text;

Success = Class.Update(Class, "Кладовщик");

Message(Success ? "Обновление данных прошла успешно!" : "Обновление данных прошла не удачно!");

Refresh(dataGridView1, "Кладовщик");

ClearRows();

break;

}

}

catch(Exception ex) { MessageBox.Show(ex.Message, ex.Source, MessageBoxButtons.OK, MessageBoxIcon.Error); }

}

void DataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

int rowIndex = e.RowIndex;

txt\_id.Text = dataGridView1.Rows[rowIndex].Cells[0].Value.ToString();

txt\_name.Text = dataGridView1.Rows[rowIndex].Cells[1].Value.ToString();

txt\_lastname.Text = dataGridView1.Rows[rowIndex].Cells[2].Value.ToString();

txt\_YaerOfBirth.Text = dataGridView1.Rows[rowIndex].Cells[3].Value.ToString();

cmb\_Gender.Text = dataGridView1.Rows[rowIndex].Cells[4].Value.ToString();

txt\_Contact.Text = dataGridView1.Rows[rowIndex].Cells[5].Value.ToString();

}

void Btn\_Delete\_Click(object sender, EventArgs e)

{

switch (comboBox1.Text)

{

case "Директор":

Class.id = int.Parse(txt\_id.Text);

Success = Class.Delete(Class, "Директор");

Message(Success ? "Удаление прошло успешно!" : "Удаление прошло не удачно!");

Refresh(dataGridView1, "Директор");

ClearRows();

break;

case "Кладовщик":

Class.id = int.Parse(txt\_id.Text);

Success = Class.Delete(Class, "Кладовщик");

Message(Success ? "Удаление прошло успешно!" : "Удаление прошло не удачно!");

Refresh(dataGridView1, "Кладовщик");

ClearRows();

break;

}

}

static string MyConnection = ConfigurationManager.ConnectionStrings["connect"].ConnectionString;

void Txt\_search\_TextChanged(object sender, EventArgs e)

{

string keyword = txt\_search.Text;

SqlConnection SqlConn = new SqlConnection(MyConnection);

try

{

switch (comboBox1.Text)

{

case "Директор":

SqlDataAdapter Sda = new SqlDataAdapter("SELECT \* FROM Директор WHERE Имя LIKE '%" + keyword + "%' OR Фамилия LIKE '%" + keyword + "%' OR Пол LIKE '%" + keyword + "%'", SqlConn);

DataTable dt = new DataTable();

Sda.Fill(dt);

dataGridView1.DataSource = dt;

break;

case "Кладовщик":

SqlDataAdapter Sda1 = new SqlDataAdapter("SELECT \* FROM Кладовщик WHERE Имя LIKE '%" + keyword + "%' OR Фамилия LIKE '%" + keyword + "%' OR Пол LIKE '%" + keyword + "%'", SqlConn);

DataTable dt1 = new DataTable();

Sda1.Fill(dt1);

dataGridView1.DataSource = dt1;

break;

}

}

catch(Exception ex) { MessageBox.Show(ex.Message, ex.Source, MessageBoxButtons.OK, MessageBoxIcon.Error); }

}

}

}

# Класс

using System;

using System.Collections.Generic;

using System.Linq;

using System.Data;

using System.Data.SqlClient;

using System.Text;

using System.Configuration;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DataBaseWorkingProj06

{

class Ext

{

public int id { get; set; }

public string Имя { get; set; }

public string Фамилия { get; set; }

public string Год\_Рождения { get; set; }

public string Пол { get; set; }

public string Контакты { get; set; }

//public string Опыт\_работы { get; set; }

// public string Желаема\_Зарплата { get; set; }

static string MyConnection = ConfigurationManager.ConnectionStrings["connect"].ConnectionString;

SqlConnection SqlConn = new SqlConnection(MyConnection);

public DataTable Select(string line)

{

DataTable dt = new DataTable();

try

{

string query = $"SELECT \* FROM {line}";

SqlCommand Command = new SqlCommand(query, SqlConn);

SqlDataAdapter Adapter = new SqlDataAdapter(Command);

Adapter.Fill(dt);

SqlConn.Open();

}

catch(Exception ex) { MessageBox.Show(ex.Message, ex.Source, MessageBoxButtons.OK, MessageBoxIcon.Error); }

finally { SqlConn.Close(); }

return dt;

}

bool IsSuccess = false;

public bool Insert (Ext Class, string line)

{

try

{

string query = $"INSERT INTO {line} (Имя, Фамилия, Год\_Рождения, Пол, Контакты)VALUES(@Имя, @Фамилия, @Год\_Рождения, @Пол, @Контакты)";

SqlCommand Command = new SqlCommand(query, SqlConn);

Command.Parameters.AddWithValue("@Имя", Class.Имя);

Command.Parameters.AddWithValue("@Фамилия", Class.Фамилия);

Command.Parameters.AddWithValue("@Год\_Рождения", Class.Год\_Рождения);

Command.Parameters.AddWithValue("@Пол", Class.Пол);

Command.Parameters.AddWithValue("@Контакты", Class.Контакты);

// Command.Parameters.AddWithValue("@Опыт\_работы", Class.Опыт\_работы);

//Command.Parameters.AddWithValue("@Желаема\_Зарплата", Class.Желаема\_Зарплата);

SqlConn.Open();

int rows = Command.ExecuteNonQuery();

IsSuccess = rows > 0 ? true : false;

}

catch(Exception ex) { MessageBox.Show(ex.Message, ex.Source, MessageBoxButtons.OK, MessageBoxIcon.Error); }

finally { SqlConn.Close(); }

return IsSuccess;

}

public bool Update(Ext Class, string line)

{

try

{

string query = $"UPDATE {line} SET Имя=@Имя, Фамилия=@Фамилия, Год\_Рождения=@Год\_Рождения, Пол=@Пол, Контакты=@Контакты WHERE id=@id";

SqlCommand Command = new SqlCommand(query, SqlConn);

Command.Parameters.AddWithValue("@id", Class.id);

Command.Parameters.AddWithValue("@Имя", Class.Имя);

Command.Parameters.AddWithValue("@Фамилия", Class.Фамилия);

Command.Parameters.AddWithValue("@Год\_Рождения", Class.Год\_Рождения);

Command.Parameters.AddWithValue("@Пол", Class.Пол);

Command.Parameters.AddWithValue("@Контакты", Class.Контакты);

SqlConn.Open();

int rows = Command.ExecuteNonQuery();

IsSuccess = rows > 0 ? true : false;

}

catch (Exception ex) { MessageBox.Show(ex.Message, ex.Source, MessageBoxButtons.OK, MessageBoxIcon.Error); }

finally { SqlConn.Close(); }

return IsSuccess;

}

public bool Delete(Ext Class, string line)

{

try

{

string query = $"DELETE FROM {line} WHERE id=@id";

SqlCommand Command = new SqlCommand(query, SqlConn);

Command.Parameters.AddWithValue("@id", Class.id);

SqlConn.Open();

int rows = Command.ExecuteNonQuery();

IsSuccess = rows > 0 ? true : false;

}

catch(Exception ex) { MessageBox.Show(ex.Message, ex.Source, MessageBoxButtons.OK, MessageBoxIcon.Error); }

finally { SqlConn.Close(); }

return IsSuccess;

}

}

}