МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського

“Харківський авіаційний інститут”

Факультет СУЛА

Кафедра математичного моделювання та штучного інтелекту

**Лабораторна робота №1**

з Організація баз даних

на тему: «Архітектури технології ADO.NET.»

Виконав: студент 3 курсу групи №335a

напряму підготовки(спеціальності):

122 Комп’ютерні науки

(шифр і назва напряму підготовки(спеціальності))

Гринюк М.О.

(прізвище й ініціали студента)

Прийняв: ст. викл., к.т.н. Коробчинський К.П.

(посада, науковий ступінь, прізвище й ініціали)

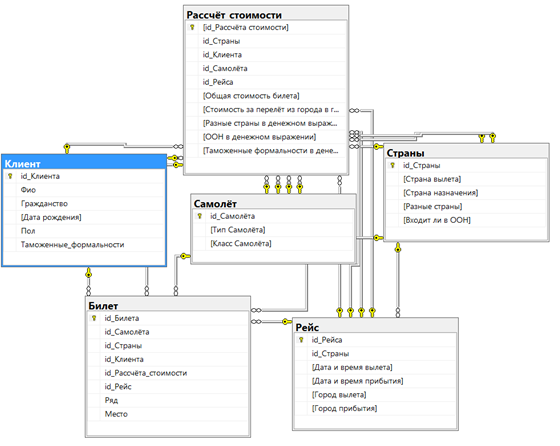
Харків – 2020

**Цель работы:** Изучить принципы построения пользовательского интерфейса, рассмотреть вопросы использования основных серверных элементов управления, применяемых для создания Web-приложений, рассмотреть вопросы организации ввода данных от пользователя.

**Ход работы:**

1. Пройти тест;
2. Добавить логическую модель предметной обрасти;
3. Создать добавить к своему проекту класс логирования;
4. Создать проект C#. Выполнить рассмотренные на лекции со строками соединения и результат вывести в лог файл:
   1. Провайдеры для получения данных из БД. Провайдер SQL Server .NET Data Provider.
   2. Строки подключение. Создание и работа со строками подключения. Объект SQLConnection.
   3. Хранение строк подключений в файле конфигураций.
   4. Безопасность строк подключения. Объект SqlConnection String Builder.
   5. Понятие пула соединений. Использование пула соединений
5. Выбрать один из вариантов соединения с БД для своего варианта.
6. **Зайдите на сайт MSDN.**Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на занятии, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. *Сохраните ссылки и дайте им короткое описание.*
7. Для каждого пункта показать программный код и пример работы(скрин)
8. Оформить отчёт и сохранить на севере STM
9. Уточнить сколько пар выполняется работа и когда необходимо её сдать.
10. Знать ответы на контрольные вопросы.

**Логическая модель БД**



**Выполнение работы**

1. Создать добавить к своему проекту класс логирования;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using NLog;

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

NLog.Logger logger = LogManager.GetCurrentClassLogger();

logger.Trace("trace message");

logger.Debug("debug message");

logger.Info("info message");

logger.Warn("warn message");

logger.Error("error message");

logger.Fatal("fatal message");

string conStr = @"Data Source=COMPUTER2\MSSQLSERVER02;

Initial Catalog=Agency;

Integrated Security=True";

using (SqlConnection conn = new SqlConnection(conStr))

{

try

{

conn.StateChange += connection\_StateChange;

conn.Open();

MessageBox.Show("Connection Open");

conn.Close();

MessageBox.Show("Connection closed");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

static void connection\_StateChange(object sender, System.Data.StateChangeEventArgs e)

{

SqlConnection connection = sender as SqlConnection;

MessageBox.Show //вывод информации о соединении и его состоянии

(

"Connection to" + Environment.NewLine +

"Data Source: " + connection.DataSource + Environment.NewLine +

"Database: " + connection.Database + Environment.NewLine +

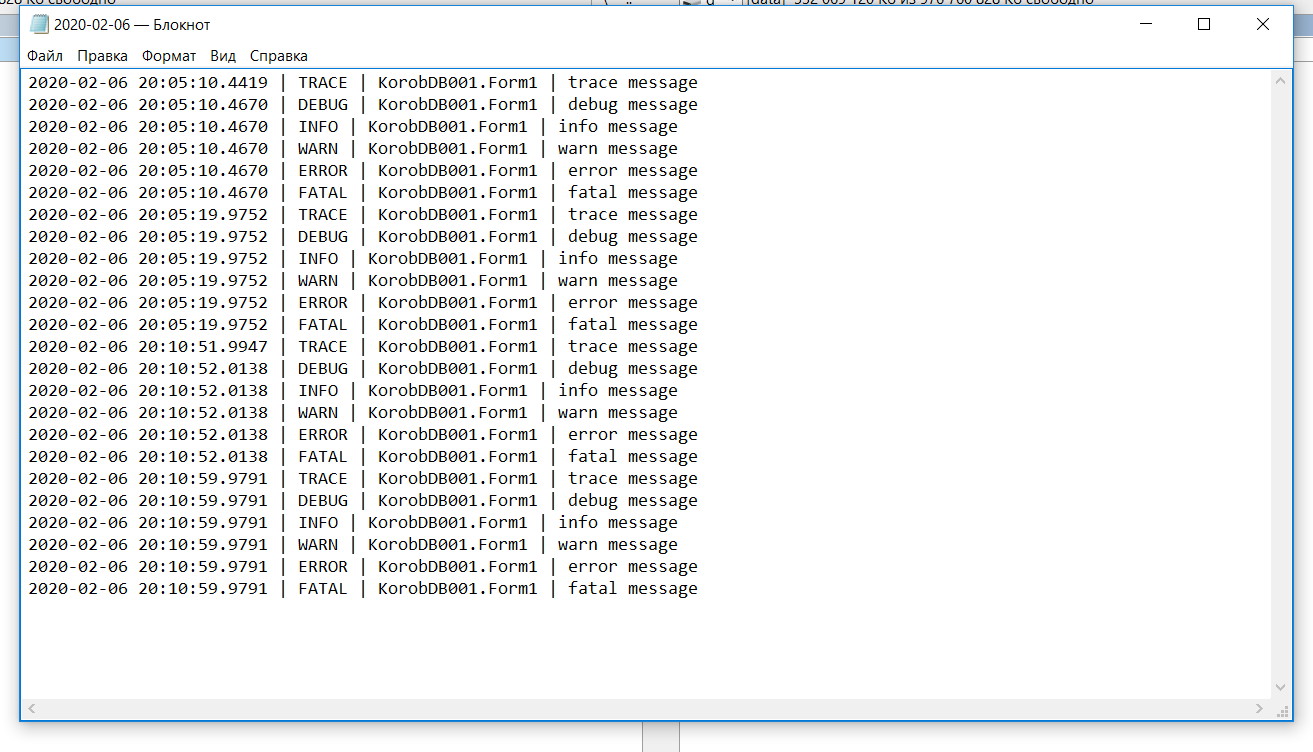
"State: " + connection.State

);

}

}

**Скриншоты работы:**



1. **Безопасность строк подключения. Объект SqlConnection String Builder**

Предоставляет простой способ создания и управления содержимым строк подключения, используемых классом SqlConnection. Код использует класс SqlConnectionStringBuilder для создания строки подключения, а затем передает свойство ConnectionString экземпляра SqlConnectionStringBuilder в конструктор класса подключения

String Builder подключения позволяет разработчикам программно создавать синтаксически правильные строки подключения, а также анализировать и перестраивать существующие строки подключения, используя свойства и методы класса.

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

SqlConnectionStringBuilder stringBuilder = new SqlConnectionStringBuilder();

stringBuilder.DataSource = @"COMPUTER2\MSSQLSERVER02";

stringBuilder.InitialCatalog = "Agency";

stringBuilder.IntegratedSecurity = true;

stringBuilder.UserID = textBox2.Text;

stringBuilder.Password = textBox1.Text;

using (SqlConnection conn = new SqlConnection(stringBuilder.ConnectionString))

{

MessageBox.Show(conn.ConnectionString);

try

{

conn.StateChange += connection\_StateChange;

conn.Open();

MessageBox.Show("Connection Open");

conn.Close();

MessageBox.Show("Connection closed");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

static void connection\_StateChange(object sender, System.Data.StateChangeEventArgs e)

{

SqlConnection connection = sender as SqlConnection;

MessageBox.Show //вывод информации о соединении и его состоянии

(

"Connection to" + Environment.NewLine +

"Data Source: " + connection.DataSource + Environment.NewLine +

"Database: " + connection.Database + Environment.NewLine +

"State: " + connection.State

);

}

1. **Понятие пула соединений. Использование пула соединений**

Подключение к источнику данных может занять много времени. Чтобы минимизировать стоимость открытия соединений, ADO.NET использует метод оптимизации, называемый пул соединений, который минимизирует стоимость многократного открытия и закрытия соединений.

Когда соединение открывается впервые, пул соединений создается на основе алгоритма точного соответствия, который связывает пул со строкой соединения в соединении. Каждый пул соединений связан с отдельной строкой соединения. При открытии нового соединения, если строка соединения не точно соответствует существующему пулу, создается новый пул.

Строки подключения также должны точно совпадать; Ключевые слова, представленные в другом порядке для одного и того же соединения, будут объединены отдельно.

using (SqlConnection connection = new SqlConnection(

"Integrated Security=SSPI;Initial Catalog=Northwind"))

{

connection.Open();

// Pool A is created.

}

using (SqlConnection connection = new SqlConnection(

"Integrated Security=SSPI;Initial Catalog=pubs"))

{

connection.Open();

// Pool B is created because the connection strings differ.

}

using (SqlConnection connection = new SqlConnection(

"Integrated Security=SSPI;Initial Catalog=Northwind"))

{

connection.Open();

// The connection string matches pool A.

}

первая и вторая строки подключения отличаются значением, назначенным для Начального каталога.

**Программный код:**

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string str = @"Data Source = COMPUTER2\MSSQLSERVER02; Initial Catalog=Agency; Integrated Security = true; Pooling = false;";

DateTime start = DateTime.Now;

for(int i=0; i< 1000; i++)

{

SqlConnection connsql = new SqlConnection(str);

connsql.Open();

connsql.Close();

}

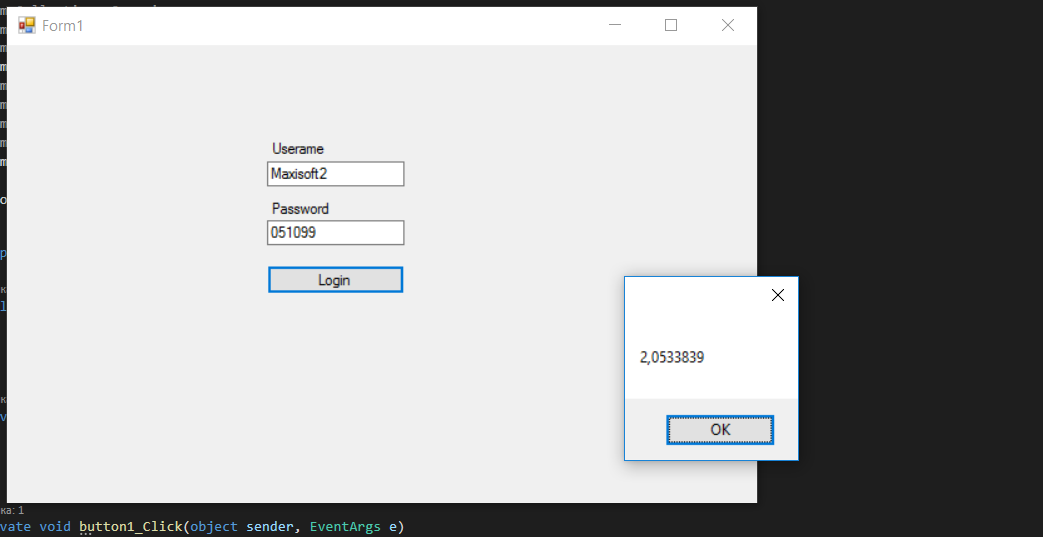
TimeSpan stop = DateTime.Now - start;

MessageBox.Show(stop.TotalSeconds.ToString());

}

}

**Скриншоты работы:**



1. **Хранение строк подключений в файле конфигураций.**

Внешние файлы конфигурации представляют собой отдельные файлы, каждый из которых содержит фрагмент файла конфигурации, состоящий из одного раздела. В таком случае основной файл конфигурации ссылается на внешний файл конфигурации.

Для хранения строк подключения во внешнем файле конфигурации создайте отдельный файл, содержащий единственный раздел **connectionStrings**. Не следует включать какие-либо дополнительные элементы, разделы или атрибуты. для работы с файлами конфигурации на локальном компьютере используется класс ConfigurationManager служит для работы с файлами конфигурации ASP.NET.

**Программный код:**

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

var settings = new ConnectionStringSettings

{

Name = "MyConnectionString1",

ConnectionString = @"Data Source = COMPUTER2\MSSQLSERVER02; Initial Catalog=Agency; Integrated Security = true;"

};

Configuration config;

config = ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);

config.ConnectionStrings.ConnectionStrings.Add(settings);

config.Save();

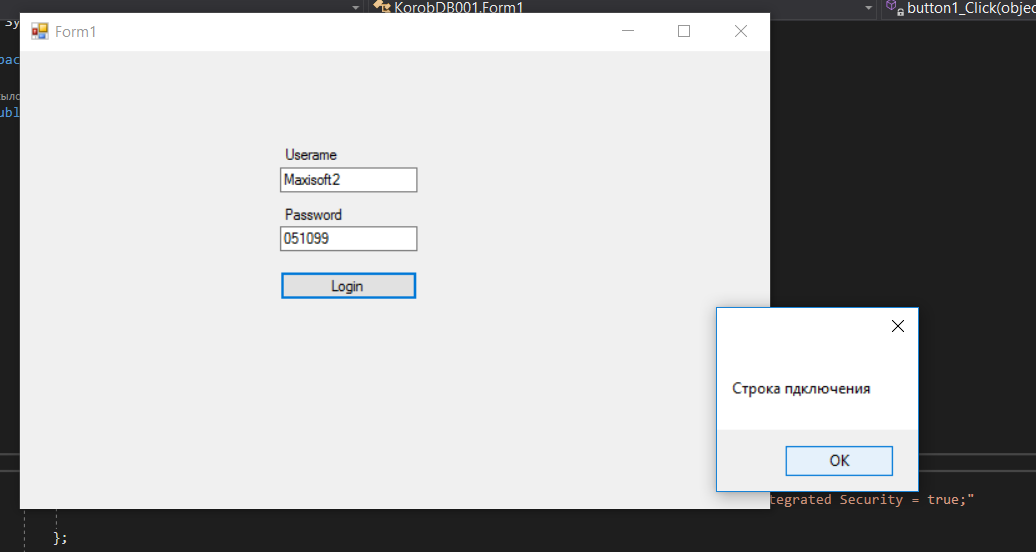
MessageBox.Show("Строка пдключения");

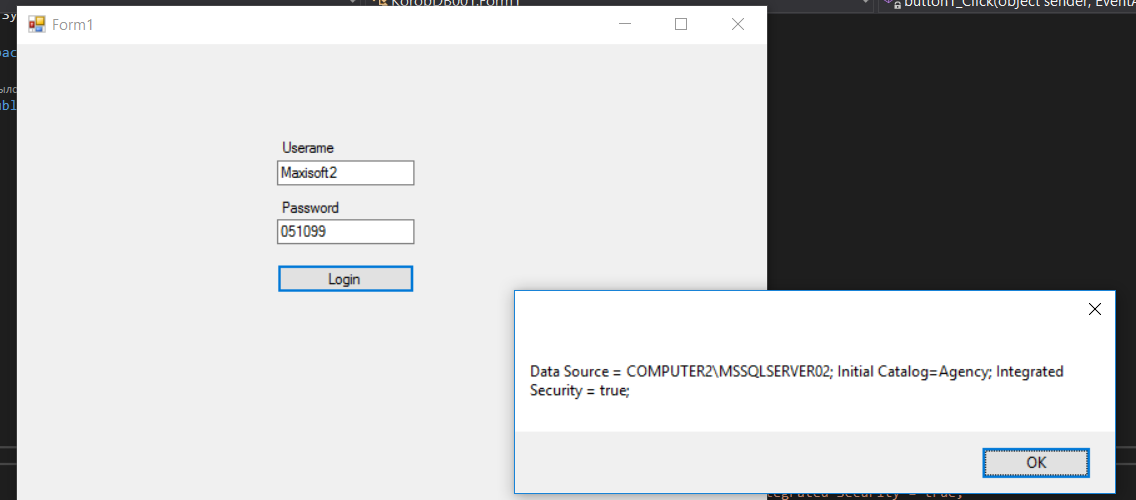
MessageBox.Show(ConfigurationManager.ConnectionStrings["MyConnectionString1"].ConnectionString);

}

}

**Скриншоты работы:**





1. **AppConfigCrypt**

**Программный код:**

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Configuration config = ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);

config.ConnectionStrings.ConnectionStrings.Add(

new ConnectionStringSettings("ConnectionStr1", "SomeConnectionString")

);

config.Save();

ConnectionStringsSection section = config.GetSection("connectionStrings") as ConnectionStringsSection;

if (section.SectionInformation.IsProtected)

{

section.SectionInformation.UnprotectSection();

}

else

{

section.SectionInformation.ProtectSection("DataProtectionConfigurationProvider");

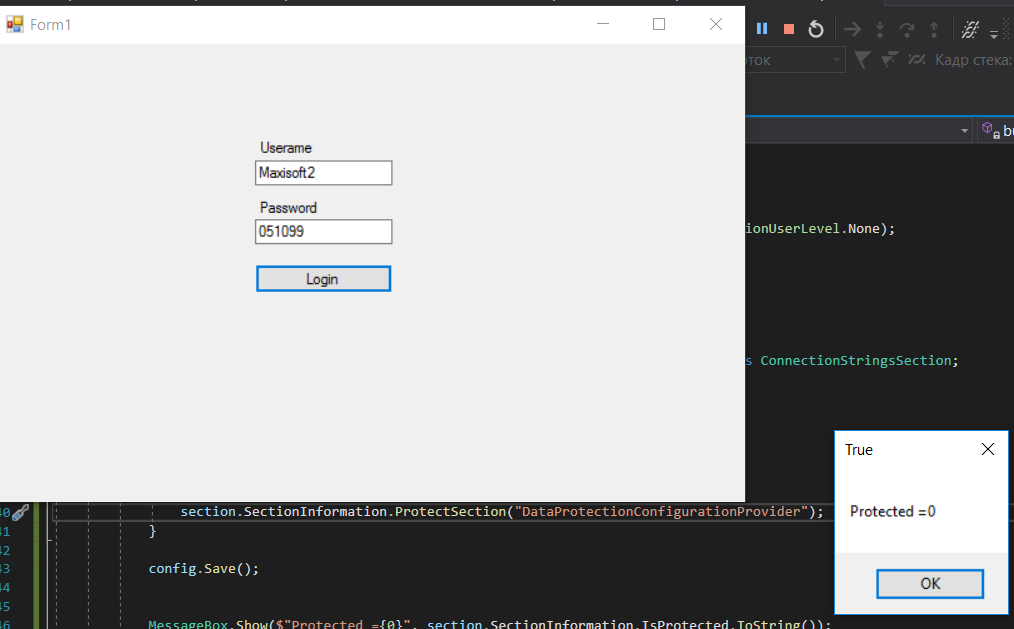
}

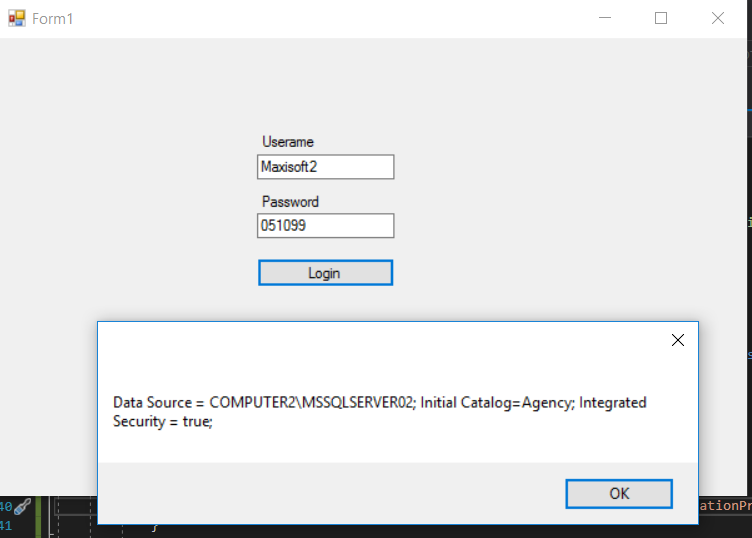
config.Save();

MessageBox.Show($"Protected ={0}", section.SectionInformation.IsProtected.ToString());

MessageBox.Show(ConfigurationManager.ConnectionStrings["MyConnectionString1"].ConnectionString);

}}





1. Создать добавить к своему проекту класс логирования

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using NLog;

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

NLog.Logger logger = LogManager.GetCurrentClassLogger();

logger.Trace("trace message");

logger.Debug("debug message");

logger.Info("info message");

logger.Warn("warn message");

logger.Error("error message");

logger.Fatal("fatal message");

string conStr = @"Data Source=COMPUTER2\MSSQLSERVER02;

Initial Catalog=Agency;

Integrated Security=True";

using (SqlConnection conn = new SqlConnection(conStr))

{

try

{

conn.StateChange += connection\_StateChange;

conn.Open();

MessageBox.Show("Connection Open");

conn.Close();

MessageBox.Show("Connection closed");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

static void connection\_StateChange(object sender, System.Data.StateChangeEventArgs e)

{

SqlConnection connection = sender as SqlConnection;

MessageBox.Show //вывод информации о соединении и его состоянии

(

"Connection to" + Environment.NewLine +

"Data Source: " + connection.DataSource + Environment.NewLine +

"Database: " + connection.Database + Environment.NewLine +

"State: " + connection.State

);

}

}

