

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА



Институт радиоэлектроники и информационных технологий  
Кафедра вычислительные системы и технологии

Практическая работа № 1  
Строка подключения

## ОТЧЕТ

по практической работе  
по дисциплине  
Базы данных

РУКОВОДИТЕЛЬ:

\_\_\_\_\_

Мисевич П.В.

СТУДЕНТ:

\_\_\_\_\_

Сапожников В.О.  
19-В-1

Работа защищена «\_\_» \_\_\_\_\_

С оценкой \_\_\_\_\_

Нижний Новгород 2021

## 1. Создание объекта SqlConnection

Для создания подключения к источнику данных в ADO.NET существует специальный объект **Connection**.

Поскольку мы работаем с MS SQL Server то нам необходимо воспользоваться объектом **SqlConnection** из пространства имен **System.Data.SqlClient**

Для работы с объектом **SqlConnection** ему нужно предоставить строку соединения, которая указывает каким образом нужно подключиться к источнику данных.

Строка соединения – строка, состоящая из пар имя – значение, содержащая сведения об инициализации, передаваемые в виде параметра от предложения к источнику данных.

Основные параметры строки подключения к MS SQL Server БД:

- **Data Source** – указывает имя экземпляра SQL Server, к которому нужно подключиться
- **Initial Catalog** – параметр, указывающий имя БД уна сервере, к которой необходимо подключиться
- **Integrated Security** – позволяет использоваться для подключения к серверу данные учетные записи Windows или имя входа SQL Server.
- **User Id** – позволяет указать имя входа SQL Server для подключения к серверу
- **Password** – пароль имени входа SQL Server

Создадим строку подключения SqlConnection передав в конструктор параметры в виде строки (скриншот 1).

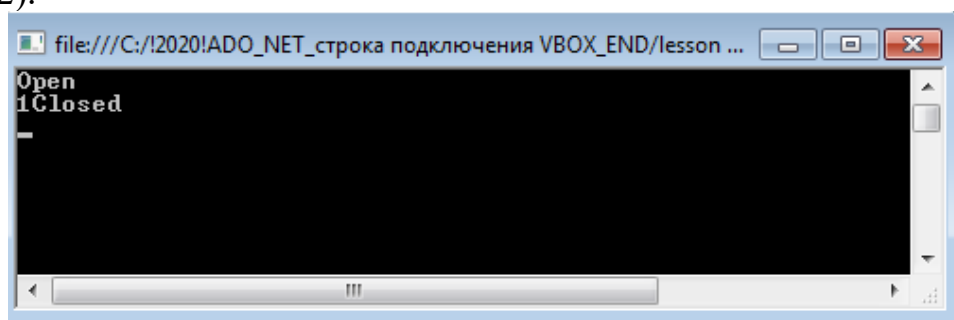
```
string conStr = @"Data Source=.\SQLEXPRESS;Initial Catalog=ShopDB; Integrated Security=True";           //строка подключения
// string conStr = @"Data Source=(local)\SQLEXPRESS;Initial Catalog=ShopDB; Integrated Security=True";
// string conStr = @"Data Source=localhost\SQLEXPRESS;Initial Catalog=ShopDB; Integrated Security=True";

// Следующим инструкциям строки подключения следует найти на локальном компьютере экземпляр SQL Server с именем SQLEXPRESS,
// поискать каталог ShopDB и попытаться получить доступ к источнику данных через доверительное подключение,
// используя для этого вашу учетную запись Microsoft Windows

SqlConnection connection = new SqlConnection(conStr);
```

Скриншот 1.

Запустив программу, убеждаемся, что подключение успешно создано (скриншот 2).



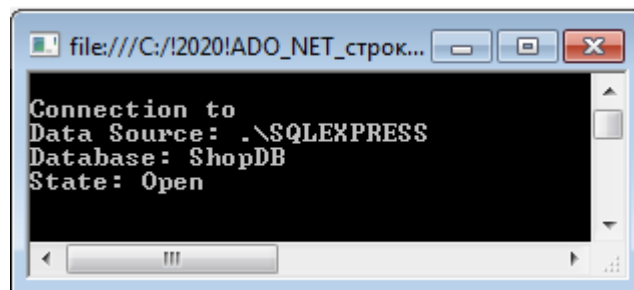
Скриншот 2.

## 2. Обращение к полям объекта SqlConnection

Мы можем напрямую общаться к параметрам строк подключения (скриншоты 3, 4)

```
static void connection_StateChange(object sender, System.Data.StateChangeEventArgs e)
{
    SqlConnection connection = sender as SqlConnection;
    Console.WriteLine();
    Console.WriteLine           //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
    );
}
```

Скриншот 3



Скриншот 4

## 3. Передача параметров подключения через формы напрямую

Мы можем использовать прямой ввод параметров строки подключения в формах (скриншоты 5, 6)

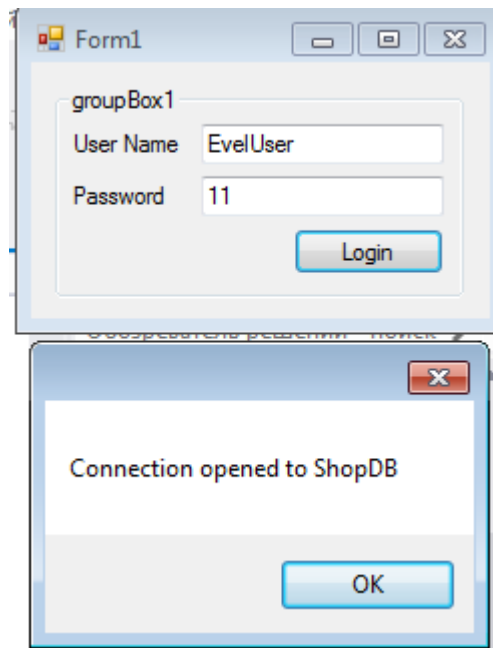
```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void LoginButton_Click(object sender, EventArgs e)
    {
        string conStr = @"Data Source=.\SQLEXPRESS;Initial Catalog=ShopDB;" + // построение строки подключения
            "User ID=" + userNameTextBox.Text + ";" +
            "Password=" + passwordTextBox.Text + ";";

        using (SqlConnection connection = new SqlConnection(conStr))
        {
            try
            {
                connection.Open();
                MessageBox.Show("Connection opened to " + connection.Database); // вывод на экран информации о подключении к базе данных
            }
            catch (Exception exception)
            {
                MessageBox.Show(exception.Message);
            }
        }
    }

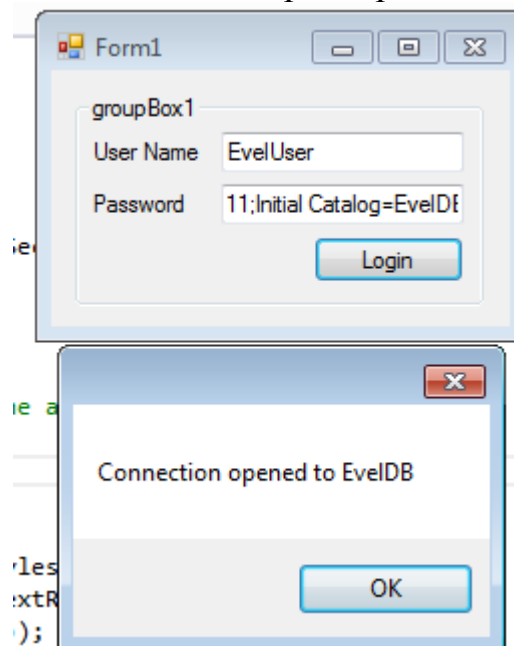
    private void groupBox1_Enter(object sender, EventArgs e)
    {
    }
}
```

Скриншот 5



Скриншот 6

Однако данный способ передачи параметров строки подключения не является безопасным, т.к. при помощи конкатенации строк пользователь может изменить прописанные в коде параметры подключения (скриншот 7).



Скриншот 7

#### 4. Передача параметров подключения через формы при помощи **ConnectionStringBuilder**

Для избегания ситуаций из примера выше необходимо использовать **ConnectionStringBuilder**. Данный класс позволяет вводить параметры пароля с экранной формы без применения конкатенации в коде программы и повысить безопасность (скриншоты 8, 9).

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void LoginButton_Click(object sender, EventArgs e)
    {
        SqlConnectionStringBuilder connectionStringBuilder = new SqlConnectionStringBuilder(); // создание

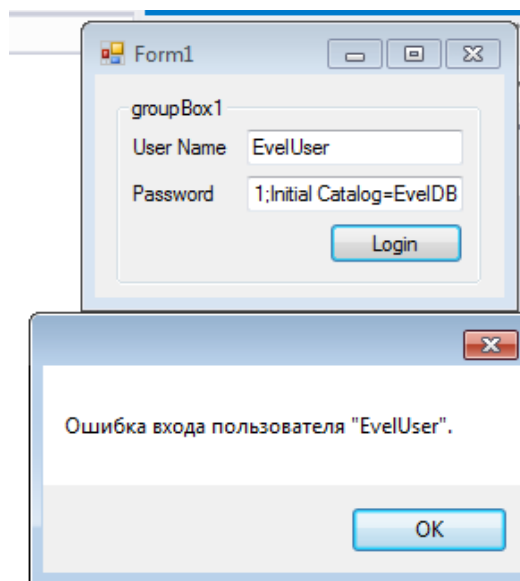
#if SetProperty

        connectionStringBuilder.DataSource = @".\SQLEXPRESS"; // используйте конструктор строк подкл
        connectionStringBuilder.InitialCatalog = "ShopDB"; // предотвращения изменения пользовател
        connectionStringBuilder.UserID = userNameTextBox.Text;
        connectionStringBuilder.Password = passwordTextBox.Text;

#else
        ...
#endif

        using (SqlConnection connection = new SqlConnection(connectionStringBuilder.ConnectionString))
        {
            try
            {
                connection.Open();
                MessageBox.Show("Connection opened to " + connection.Database);
            }
            catch (Exception exception)
            {
                MessageBox.Show(exception.Message);
            }
        }
    }
}
```

Скриншот 8



Скриншот 9

## **Вывод**

Необходимо использовать инструментarii обеспечения безопасности проекта. Только так можно обеспечить информационную безопасность проект в течении всего жизненного цикла.