

## ТЕХНОЛОГИИ ДОСТУПА К ДАННЫМ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

ADO.NET — это набор классов, предоставляющих службы доступа к данным программистам, которые используют платформу .NET Framework. ADO.NET удовлетворяет различные потребности разработчиков, включая создание клиентских приложений баз данных, а также бизнес-объектов среднего уровня, используемых приложениями, средствами, языками и браузерами (<https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/>).

### Практическое занятие 2. Объект соединения

Объект *Connection* служит для установления соединения с заданными источниками данных. Соединение можно настроить, редактируя нужным образом значения параметров строки подключения - свойство *ConnectionString*.

Базовый формат строки подключения содержит ряд пар ключевых слов и значений, разделенных точкой с запятой. В следующем примере показана типичная строка подключения ([https://msdn.microsoft.com/ru-ru/library/system.data.sqlclient.sqlconnection.connectionstring\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.data.sqlclient.sqlconnection.connectionstring(v=vs.110).aspx)).

```
cnn.ConnectionString = "Persist Security Info=False; Integrated Security=true; Initial Catalog=Northwind; server=(local)";
```

Ключевые слова не учитывают регистр, порядок параметров не имеет значения. Полный список параметров строки соединения и особенности их описания, например, указания сетевого адреса экземпляра SQL Server к которому выполняется подключение, приводятся в материалах для разработчиков от компании Microsoft [msdn.microsoft.com](https://msdn.microsoft.com).

Объект *Connection* имеет методы *Open()* и *Close()*, которые открывают/закрывают подключение к базе данных со значениями свойств, определяемыми свойством *ConnectionString*.

Объект *Connection* имеет набор свойств, например, *ServerVersion*, *State*, которые отображают состояние объекта во время последней сетевой операции, выполненной при подключении.

Создайте *Windows-проект*, организуйте *MDI-приложение*. В форме проекта поместите элементы для отображения данных, извлекаемых из базы данных, и командные кнопки для выполнения действий. Пример такого приложения показан на рисунке 2. Командный интерфейс можно формировать по мере необходимости – при изучении объектов технологии ADO .NET.

Рассмотрим возможности объекта *Connection*. С помощью командной кнопки *Соединение 1* и элемента *richTextBox* обеспечьте выполнение следующих действий:

- Создание экземпляра объекта *Connection*.
- Отображение текущего состояния подключения.
- Открытие подключения.
- Отображение состояния подключения.
- Закрытие подключения.
- Отображение состояния подключения.

На рисунке 1 приведен пример выполнения перечисленных действий. Для описания строки сообщения обязательно используйте метод *Format()* класса *String*. На рисунке 2 видно как при щелчке по кнопке Соединение 1 в элемент *richTextBox* выводятся значения свойств объекта *Connection*.

```
//Создание объекта соединения
SqlConnection cnn = new SqlConnection();

//Вариант 1 создания объекта соединения
private void button1_Click(object sender, EventArgs e)
{
    cnn = new SqlConnection(@"Data Source=ASUS;Initial Catalog=Employees;Integrated Security=True");
    // Открыть соединение
    cnn.Open();
    richTextBox1.Text = String.Format("Версия сервера:{0} \n", cnn.ServerVersion);
    richTextBox1.Text += String.Format("Состояние соединения: {0}\n",cnn.State.ToString());
    // Закреть соединение
    cnn.Close();
    richTextBox1.Text += String.Format("Состояние соединения:{0} ", cnn.State.ToString());
}
```

Рисунок 1 – Пример формирования объекта соединения

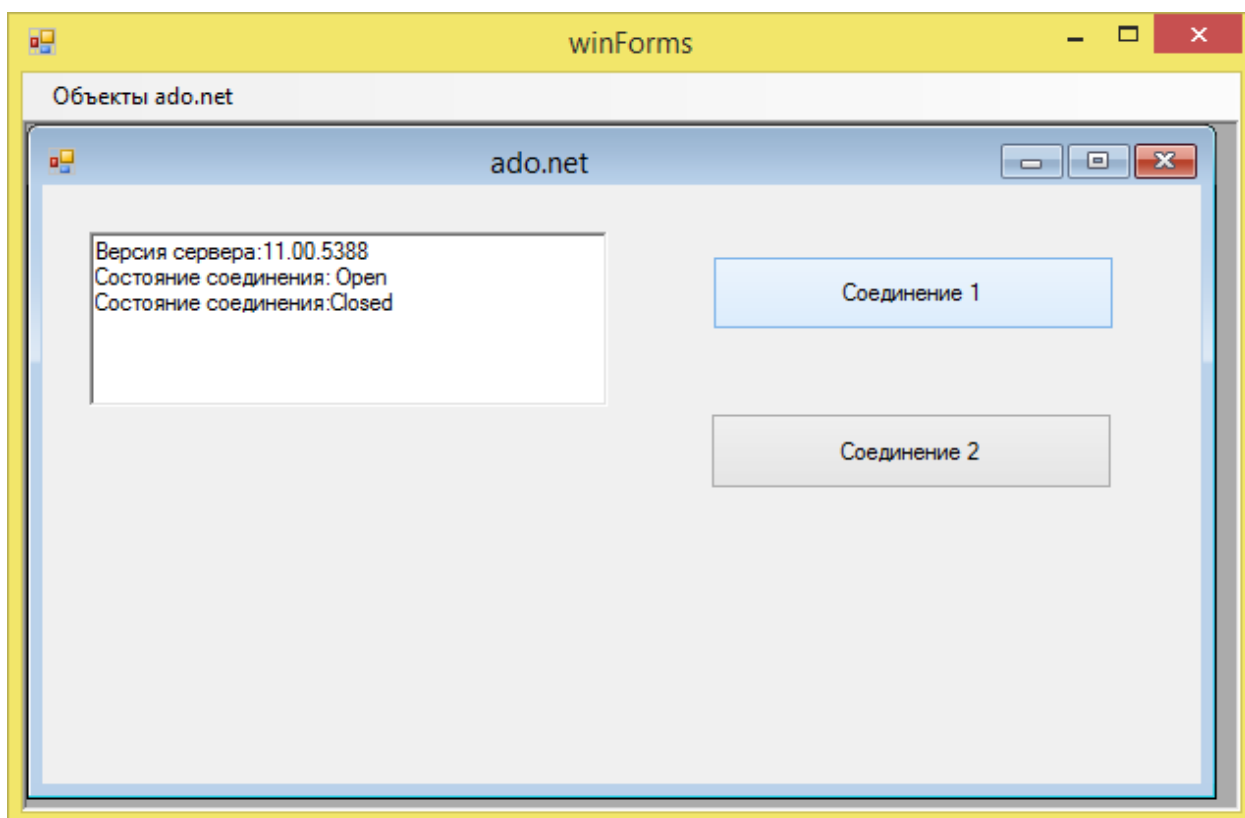


Рисунок 2 - Запуск windows-приложения с подключением к базе данных

### Управление строками соединений в windowsForms

При использовании в windows-приложении различных поставщиков баз данных для определения строк соединения удобно использовать раздел `<connectionStrings>` конфигурационного файла `App.config`. На рисунке 3 приведен пример строки подключений к источнику данных, размещенной в области `connectionStrings`. Таких строк подключения к различным источникам может быть несколько; каждая строка должна иметь наименование, например, `name="Employees"` как на рисунке 3. Имя может быть произвольным.

Еще одним параметром строки является `providerName`. В нашем случае `providerName="System.Data.SqlClient"`. Если вы используете базу данных, отличную от SQL Server, то необходимо познакомиться с документацией поставщика данных (или справочником по библиотеке классов .NET Framework), чтобы узнать о поддерживаемых значениях строки соединения.

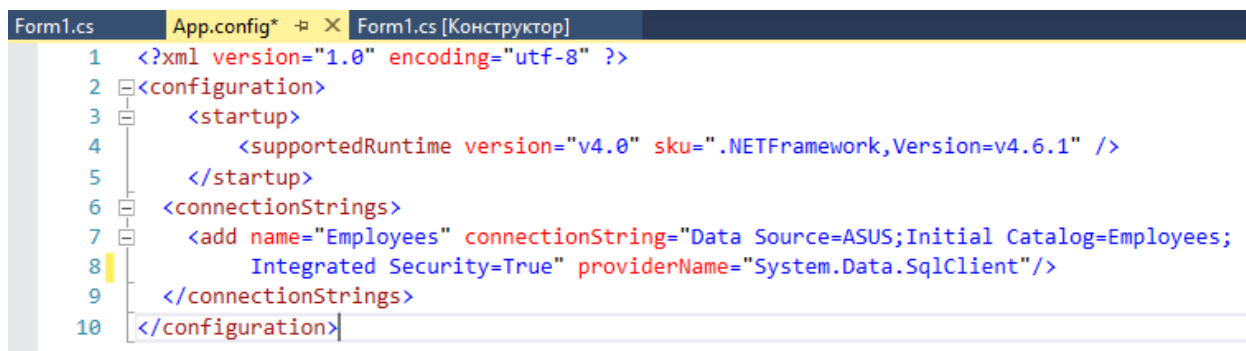


Рисунок 3 – Пример определения строки соединения в конфигураторе winForms

После того, как информация подключения к базе данных определена в конфигурационном файле, ее можно использовать в приложении. На рисунке 4 приведен программный код варианта создания объекта соединения, параметры которого прочитаны из конфигурационного файла при обращении к именованной области *Employees*.

Не забудьте добавить ссылки на сборку *System.Configuration*, чтобы обеспечить доступность класса *ConfigurationManager*, который предоставляет доступ к файлам конфигурации клиентских приложений (<https://docs.microsoft.com/ru-ru/dotnet/api/system.configuration.configurationmanager?view=netcore-2.2>).

```
//Вариант 2 создания объекта соединения
private void button2_Click(object sender, EventArgs e)
{ // Получить параметры объекта Connection
  string connectionString = ConfigurationManager.ConnectionStrings["Employees"].ConnectionString;
  cnn = new SqlConnection(connectionString);
  // Открыть соединение
  cnn.Open();
  richTextBox1.Text += String.Format("Версия сервера:{0} \n", cnn.ServerVersion);
  richTextBox1.Text += String.Format("Состояние соединения: {0}\n", cnn.State.ToString());
  // Закрыть соединение
  cnn.Close();
  richTextBox1.Text += String.Format("Состояние соединения:{0} ", cnn.State.ToString());
}
```

Рисунок 4 – Получение параметров объекта соединения из именованной области App.config

На рисунке 5 показаны варианты соединений с базой данных: без использования управляемых строк соединений (щелчок по кнопке Соединение 1) и с управляемыми строками (щелчок по кнопке Соединение 2).

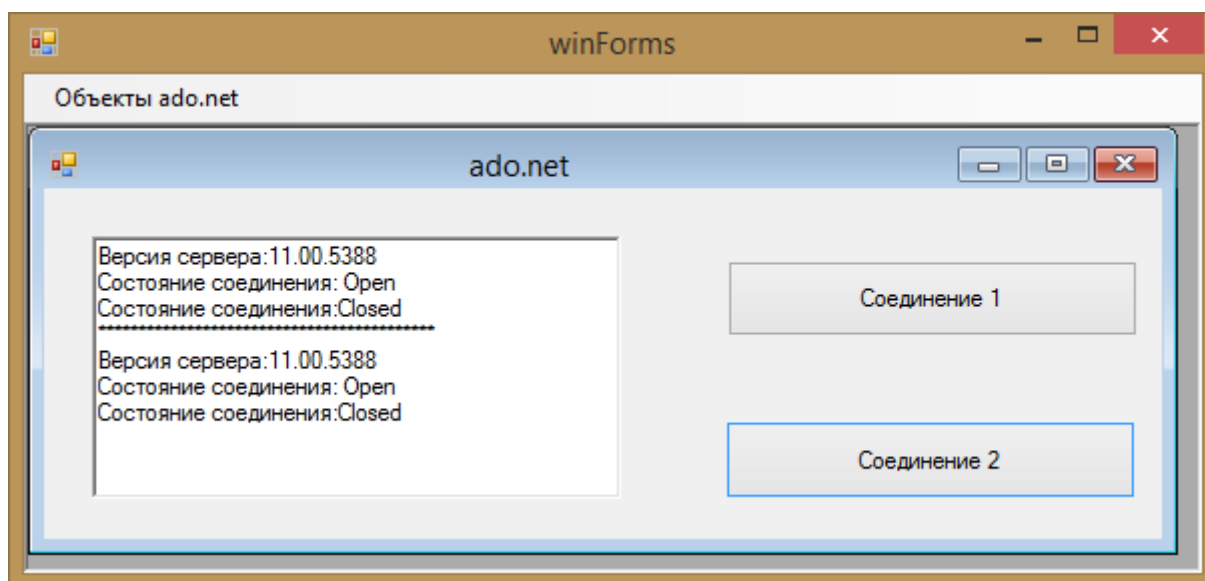


Рисунок 5 – Состояние windows-приложения с вариантами соединений

## Управление строками соединений в webForms

Технология ADO.NET позволяет писать более-менее одинаковый код для доступа к данным — как в веб-приложениях, в клиент-серверных настольных приложениях, и даже в однопользовательских приложениях, подключаемых к локальной базе данных, например, MS Access.

Создайте веб-приложение ASP.NET (.NET Framework). Для изучения архитектуры ADO.NET в контексте веб-приложений ASP.NET будем использовать самый простой шаблон веб-проекта как показано на рисунках 6 и 7.

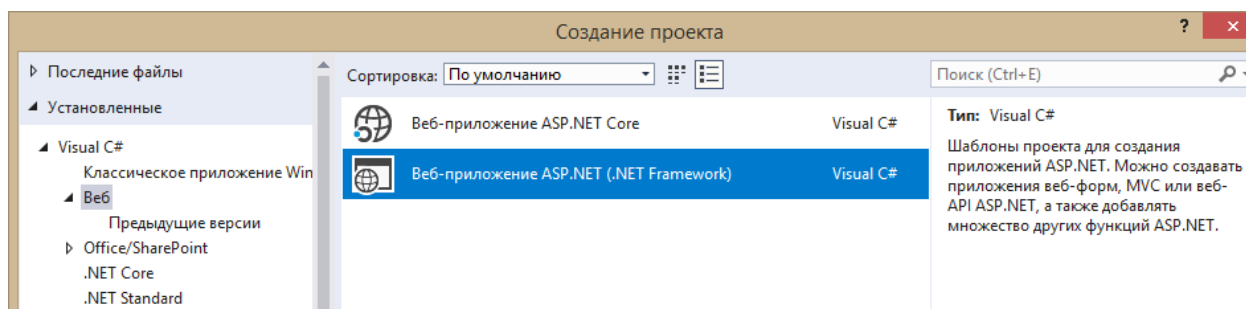


Рисунок 6 – Выбор шаблона для веб-проекта

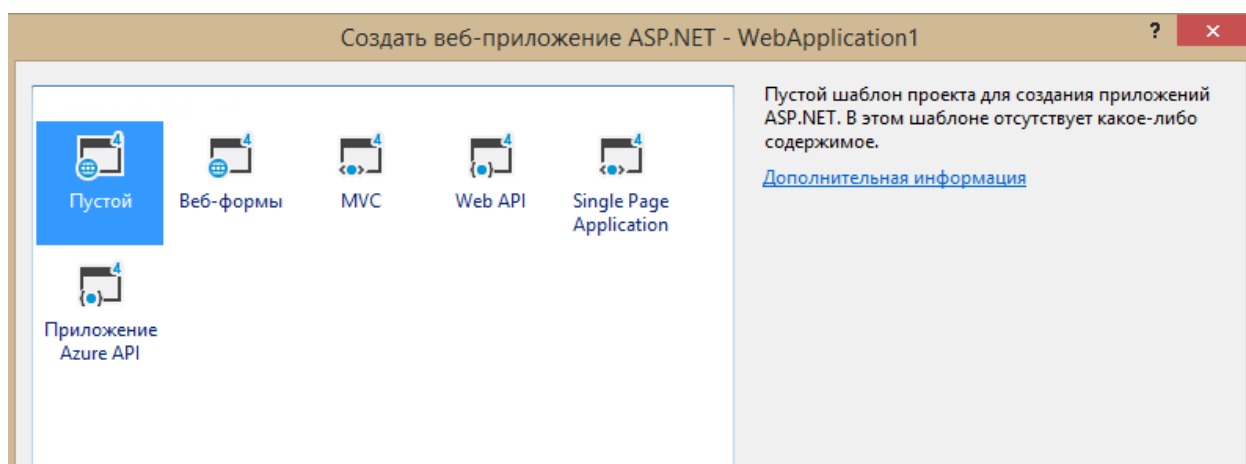


Рисунок 7 – Выбор пустого шаблона для исследования

Добавьте в проект Web-форму, разместите на странице элементы управления *TextBox* и *Button*. У элемента *TextBox* можно установить свойство *TextMode* в значение *MultiLine*. Рисунок 8 показывает структуру страницы с элементами интерфейса.

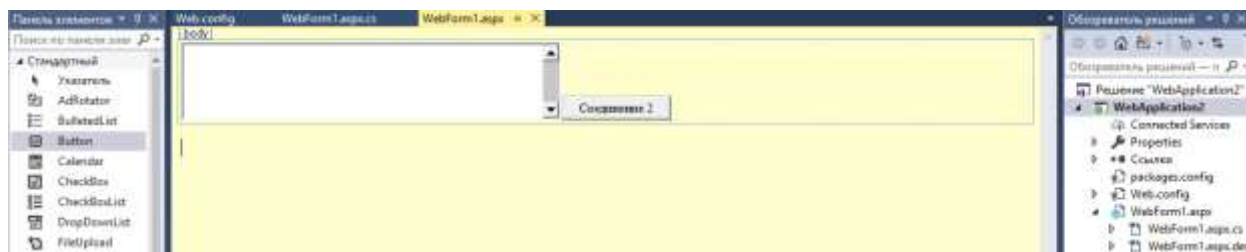


Рисунок 8 – Формирование интерфейса web-страницы

Теперь можно скопировать раздел `connectionString` в конфигурационный файл веб-приложения как на рисунке 9.

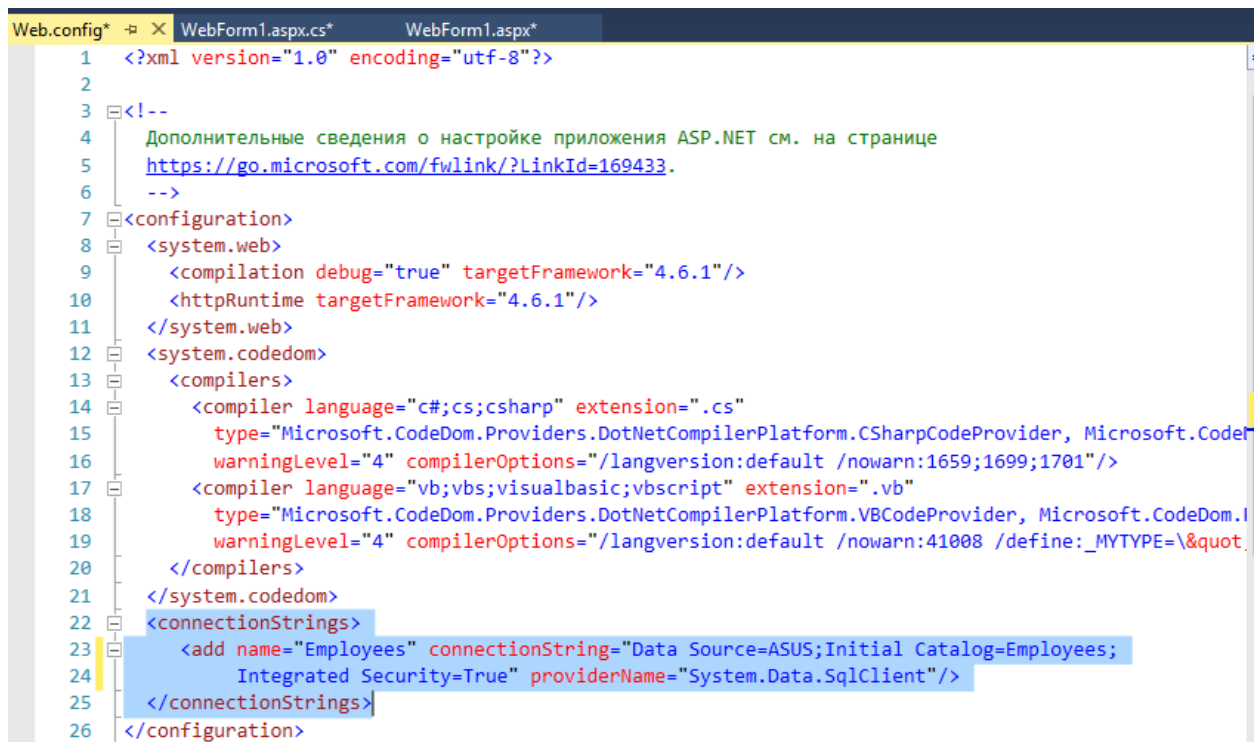


Рисунок 9 – Пример определения строки соединения в webForms

Затем эту строку соединения легко извлечь по имени из коллекции *WebConfigurationManager.ConnectionStrings*. При условии, что импортировано пространство имен *System.Web.Configuration* как показано на рисунке 10.

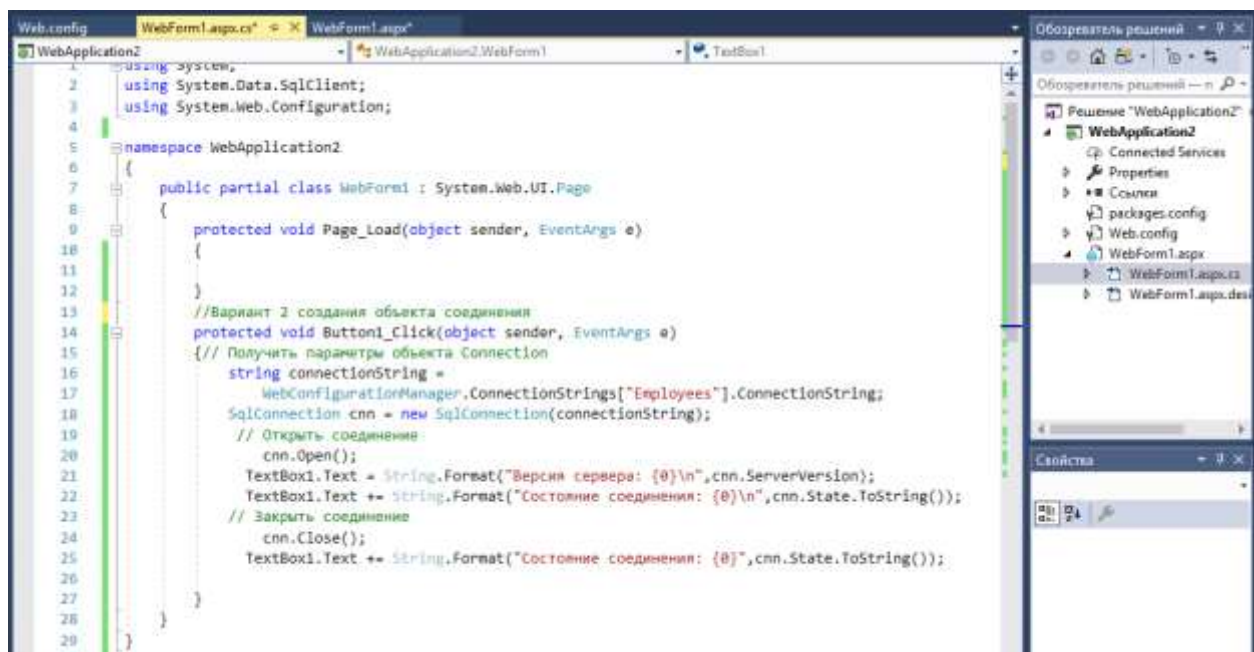


Рисунок 10 – Получение параметров объекта соединения из именованной области Web.config

Выполните тестирование веб-страницы. На рисунке 11 показан пример выполнения соединения.

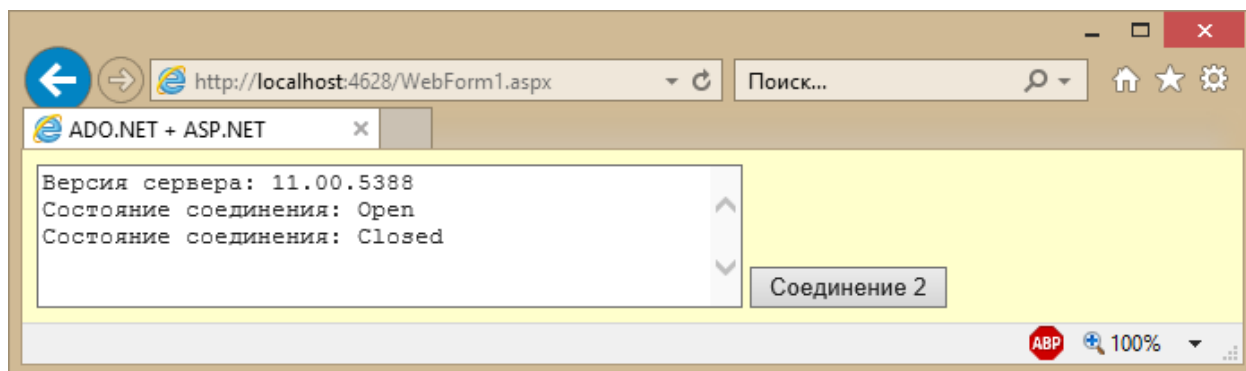


Рисунок 11 – Состояние web-приложения с вариантом соединения

### Задание для самостоятельной работы

У вас должна сохраниться база данных под управлением СУБД MS Access. Выполните подключение к базе данных из формы и страницы ваших приложений. Не забывайте о смене провайдера данных и импортировании необходимых пространств имен. На рисунке 12 показано использование панели *Обозреватель серверов* для получения параметров строки соединения. Параметры строки соединения можно скопировать из панели свойств объекта соединения в поле Строка подключения.

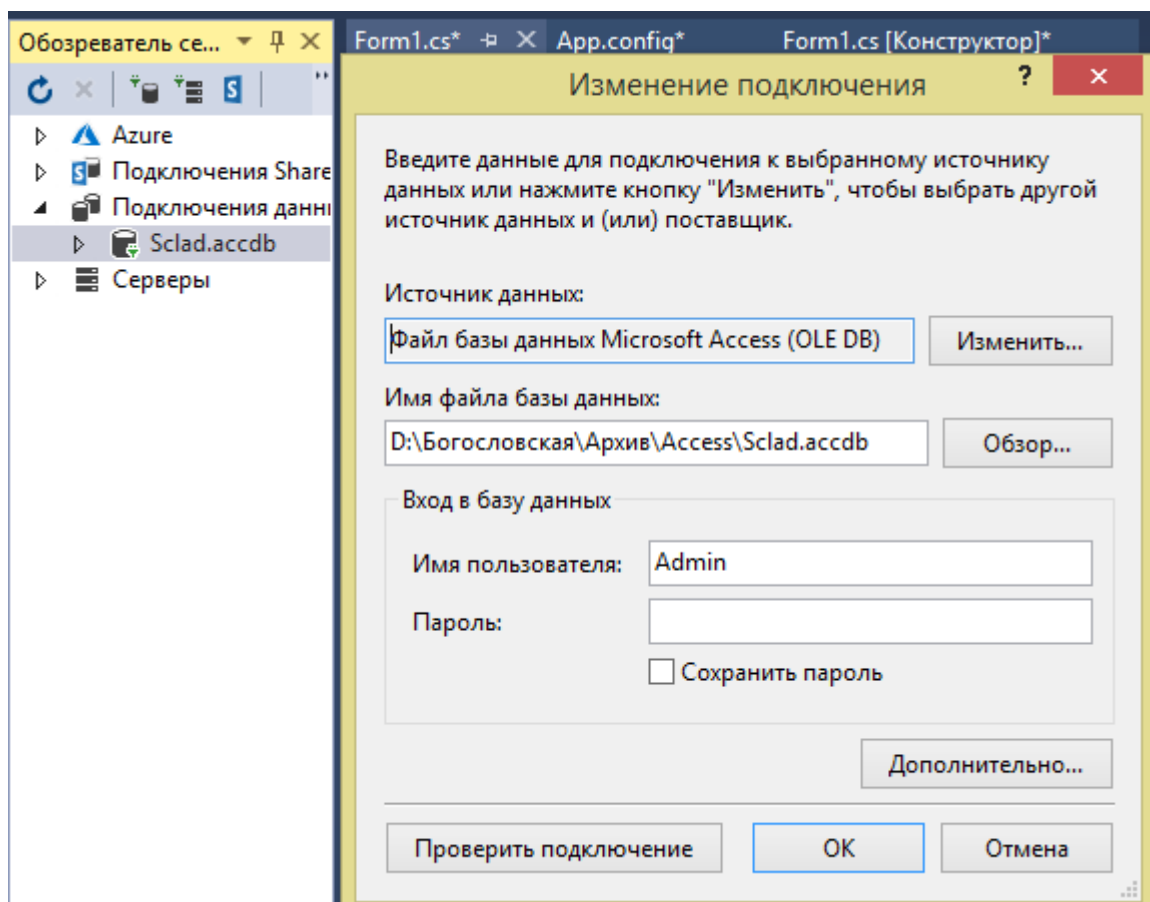


Рисунок 12 – Подключение к базе данных с помощью панели Обозреватель серверов

На рисунке 13 демонстрируется результат подключений к базе данных Employees на MS Sql Server и базе данных Dogovor на MS Access.

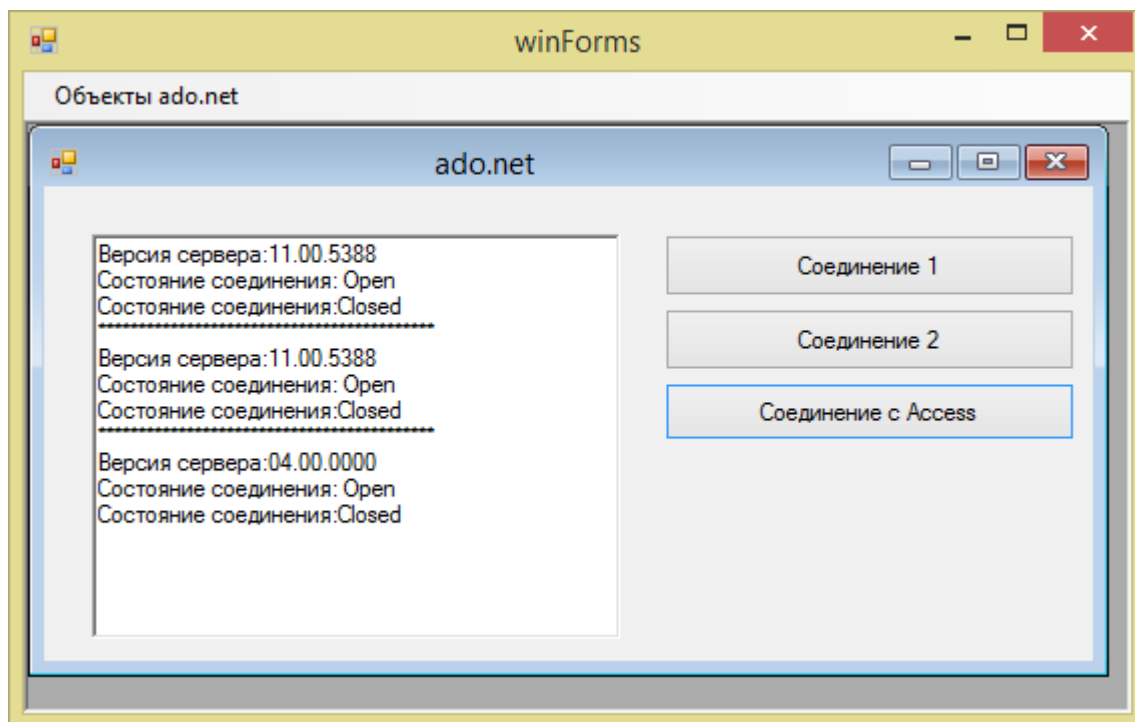


Рисунок 13 – Результат подключения к базе данных Dogovor в MS Access

В отчете к практической работе 2 необходимо показать все программные коды для реализации подключений с подробными комментариями к ним. Подключения должны быть выполнены из win-формы и web-формы. Комментарии к коду, приведенные в практикуме, являются примерами очень краткого описания, ваши комментарии должны быть расширенными.

Результаты исследования возможностей объекта *Connection* обязательно продемонстрируйте преподавателю.