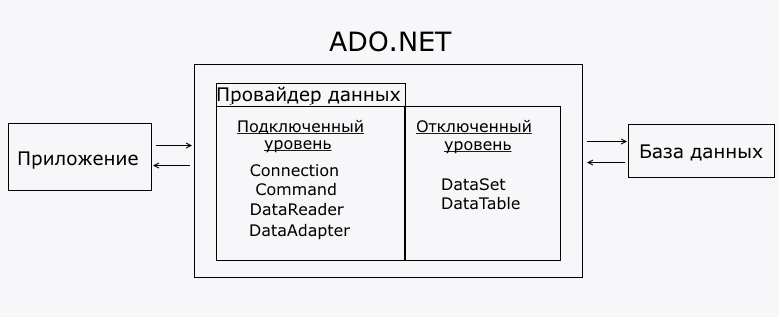
1. **Какова архитектура компонентов ADO.NET?**



1. **Как производится подключение к базе данных?**

class Program

{

    static void Main(string[] args)

    {

        string connectionString = @"Data Source=.\SQLEXPRESS;Initial Catalog=usersdb;Integrated Security=True";

    }

}

Строка подключения представляет набор параметров в виде пар ключ=значение. В данном случае для подключения к ранее созданной базе данных usersdb мы определяем строку подключения из трех параметров:

* Data Source: указывает на название сервера. По умолчанию это ".\SQLEXPRESS". Поскольку в строке используется слеш, то в начале строки ставится символ @. Если имя сервера базы данных отличается, то соответственно его и надо использовать.
* Initial Catalog: указывает на название базы данных на сервере
* Integrated Security: устанавливает проверку подлинности

1. **Охарактеризуйте используемые в лабораторной работе объекты доступа к данным.**

В конструктор объекту **SqlConnection** передается строка подключения, которая инициализирует объект. Чтобы использовать этот объект и подключаться к базе данных, мы должны выполнить его метод Open(), а после завершения работы с базой данных нам надо вызвать метод Close() для закрытия подключения. В случае возникновения ошибок закрытие подключения происходит в блоке finally.

После установки подключения мы можем выполнить к базе данных какие-либо команды, например, добавить в базу данных объект, удалить, изменить его или просто извлечь. Команды представлены объектом интерфейса System.Data.IDbCommand. Провайдер для MS SQL предоставляет его реализацию в виде класса **SqlCommand**. Этот класс инкапсулирует sql-выражение, которое должно быть выполнено.

Чтобы выполнить команду, необходимо применить один из методов SqlCommand:

* **ExecuteNonQuery**: просто выполняет sql-выражение и возвращает количество измененных записей. Подходит для sql-выражений INSERT, UPDATE, DELETE.
* **ExecuteReader**: выполняет sql-выражение и возвращает строки из таблицы. Подходит для sql-выражения SELECT.
* **ExecuteScalar**: выполняет sql-выражение и возвращает одно скалярное значение, например, число. Подходит для sql-выражения SELECT в паре с одной из встроенных функций SQL, как например, Min, Max, Sum, Count.

**DataSet** представляет хранилище данных, с которыми можно работать независимо от наличия подключения, а **SqlDataAdapter** заполняет **DataSet** данными из БД.