

# Программирование клиентских приложений

ЛЕКЦИЯ 1

# Содержание лекции

**01**

**ADO.NET**

**02**

Архитектура **ADO.NET**

# ADO.NET

# Что такое ADO.NET



**ADO.NET** — это набор классов, предоставляющих службы доступа к данным программистам, которые используют платформу **.NET Framework**.

# ADO.NET

**ADO.NET** предоставляет согласованный доступ к таким источникам данных, как **SQL Server** и **XML**, а также к источникам данных, предоставляемым при помощи **OLE DB** и **ODBC**.

Пользовательские приложения, использующие общие данные, могут использовать **ADO.NET** для соединения с этими источниками данных и для **получения, обработки и обновления** имеющихся в них данных.

# ADO.NET

**ADO.NET** разделяет доступ к данным и обработку данных на **дискретные компоненты**, которые могут использоваться отдельно или совместно.

**ADO.NET** включает **поставщиков данных .NET Framework** для соединения с базой данных, выполнения команд и получения результатов. Эти результаты, помещенные в объект **ADO.NET DataSet**, обрабатываются так, чтобы они могли быть предоставлены пользователю нерегламентированным образом, объединенные с данными из многих источников или передаваемые между уровнями.

# ADO.NET

Объект **DataSet** также может независимо использоваться поставщиком данных **.NET Framework** для управления локальными для приложения данными или данными, источником которых является **XML**.

Классы **ADO.NET** располагаются в пространстве имен **System.Data** и интегрируются с классами **XML** пространства имен **System.Xml**.

# АРХИТЕКТУРА **ADO.NET**



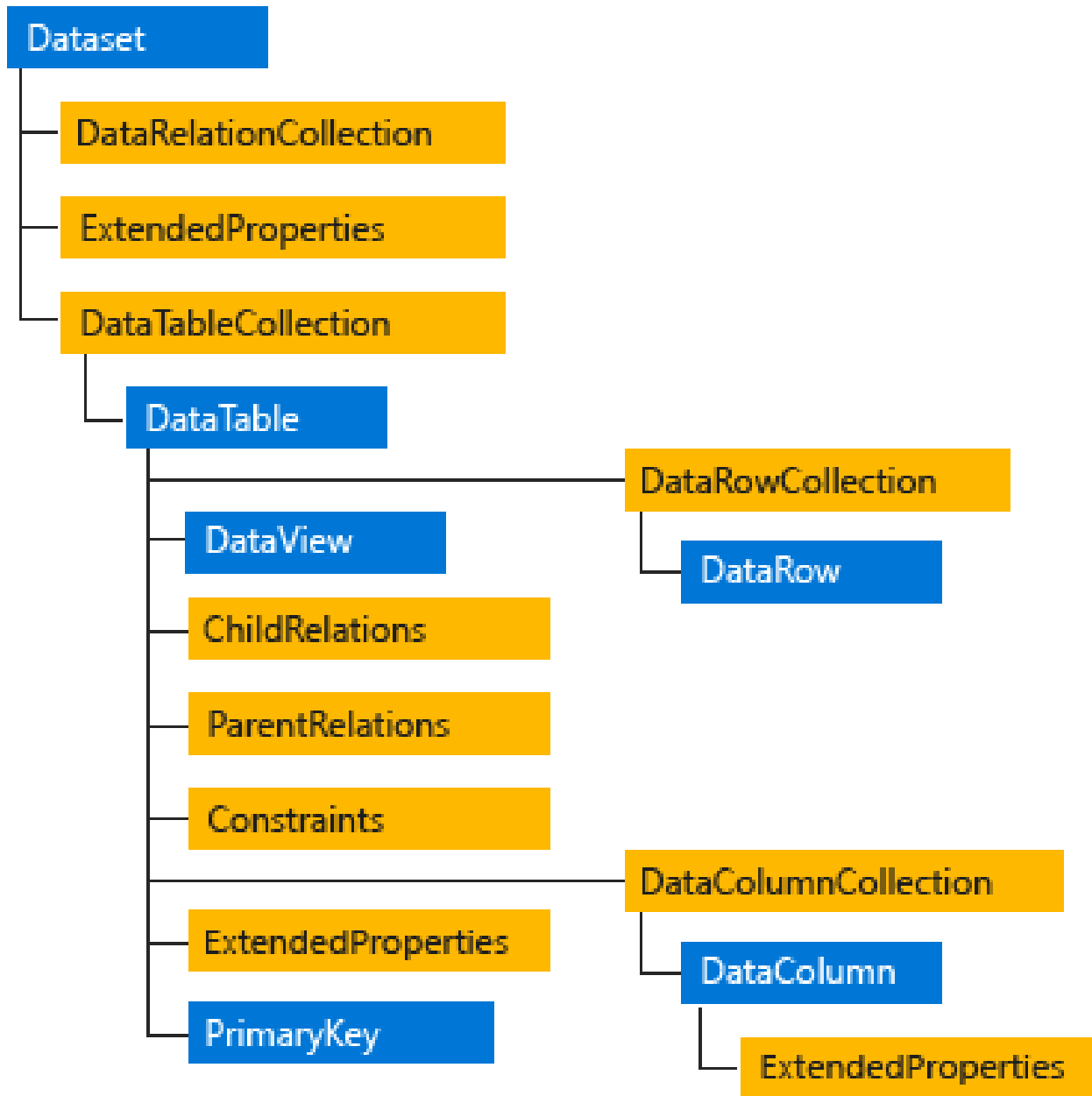
# Data Provider

Поставщики данных (**data provider**) — это компоненты, которые используются для подключения к базе данных, выполнения команд и получения результатов.

Эти результаты обрабатываются и помещаются в **DataSet** для дальнейшего использования пользователем.

Поставщик данных .NET Framework	Описание	Пространство имен
<b>SQL Server Data Provider</b>	Предоставляет доступ к данным для <b>Microsoft SQL Server</b> .	<b>System.Data.SqlClient</b>
<b>OLE DB Data Provider</b>	Для источников данных <b>OLE DB</b> .	<b>System.Data.OleDb</b>
<b>ODBC Data Provider</b>	Для источников данных <b>ODBC</b> .	<b>System.Data.Odbc</b>
<b>Oracle Data Provider</b>	Для источников данных <b>Oracle</b> . Поддерживает клиентское ПО <b>Oracle</b> версии <b>8.1.7</b> и выше.	<b>System.Data.OracleClient</b>
<b>EntityClient Data Provider</b>	Предоставляет доступ к данным для приложений модели <b>EDM (Entity Data Model)</b> .	<b>System.Data.EntityClient</b>
<b>SQL Server Compact 4.0 Data Provider</b>	Предоставляет доступ к данным для <b>Microsoft SQL Server Compact 4.0</b> .	<b>System.Data.SqlServerCe</b>

# Архитектура **ADO.NET**



Объект	Описание
<b>Connection</b>	Устанавливает соединение с конкретным источником данных. Базовым классом для всех объектов <b>Connection</b> является <b>DbConnection</b> .
<b>Command</b>	Выполняет команду в источнике данных. Обеспечивает доступность <b>Parameters</b> и может выполнять команды в области <b>Transaction</b> из <b>Connection</b> . Базовым классом для всех объектов <b>Command</b> является <b>DbCommand</b> .
<b>DataReader</b>	Считывает из источника данных однократный набор данных только для чтения. Базовым классом для всех объектов <b>DataReader</b> является <b>DbDataReader</b> .
<b>DataAdapter</b>	Заполняет <b>DataSet</b> и выполняет обновления в источнике данных. Базовым классом для всех объектов <b>DataAdapter</b> является <b>DbDataAdapter</b> .

Объект	Описание
<b>Transaction</b>	Добавляет команды к транзакциям в источнике данных. Базовым классом для всех объектов <b>Transaction</b> является <b>DbTransaction</b> . <b>ADO.NET</b> также поддерживает транзакции, использующие классы в пространстве имен <b>System.Transactions</b> .
<b>CommandBuilder</b>	Позволяет автоматически сформировать команду <b>DataAdapter</b> или извлечь сведения о параметрах из хранимой процедуры и заполнить коллекцию <b>Parameters</b> объекта <b>Command</b> . Базовым классом для всех объектов <b>CommandBuilder</b> является <b>DbCommandBuilder</b> .
<b>ConnectionStringBuilder</b>	Позволяет создавать и управлять содержимым строки соединения, которую используют объекты <b>Connection</b> . Базовым классом для всех объектов <b>ConnectionStringBuilder</b> является <b>DbConnectionStringBuilder</b> .

Объект	Описание
<b>Parameter</b>	Определяет входные, выходные и возвращаемые значения параметров для команд и хранимых процедур. Базовым классом для всех объектов <b>Parameter</b> является <b>DbParameter</b>
<b>Exception</b>	Возвращается при возникновении ошибки в источнике данных. Базовым классом для всех объектов <b>Exception</b> является <b>DbException</b> .
<b>Error</b>	Сведения, относящиеся к предупреждениям и ошибкам, возвращаемым источником данных.
<b>ClientPermission</b>	Предоставляется для атрибутов безопасности доступа к коду поставщика данных. Базовым классом для всех объектов <b>ClientPermission</b> является <b>DBDataPermission</b> .

# DataSet

Класс **DataSet** используется для доступа к данным независимо от источника данных.

Это расположенное в оперативной памяти представление данных, обеспечивающее согласованную реляционную программную модель данных.

Объект **DataSet** содержит коллекцию из одного или нескольких объектов **DataTable**, состоящих из строк и столбцов данных, а также первичный ключ, внешние ключи, ограничения и другие сведения о данных в объектах **DataTable**.

# Работа с DataSet

Разработчик может:

- программно создать **DataTable**, **DataRelation** и **Constraint** внутри **DataSet** и заполнить таблицы данными
- заполнить **DataSet** таблицами данных из существующего реляционного источника данных с помощью **DataAdapter**
- загрузить и сохранить содержимое **DataSet** с помощью **XML**-кода



# Пример работы с DataSet

```
DataSet customerOrders = new("CustomerOrders");

DataTable ordersTable = customerOrders.Tables.Add("Orders");

DataColumn pkOrderID =
    ordersTable.Columns.Add("OrderID", typeof(int));
ordersTable.Columns.Add("OrderQuantity", typeof(int));
ordersTable.Columns.Add("CompanyName", typeof(string));

ordersTable.PrimaryKey = new DataColumn[] { pkOrderID };
```

# ВОПРОСЫ ПО ЛЕКЦИИ