

# ТИПОВЫЕ КЛИЕНТЫ ДОСТУПА К БАЗЕ ДАННЫХ НА ОСНОВЕ РАЗЛИЧНЫХ ТЕХНОЛОГИЙ (4 ЧАСА)

ПРЕПОДАВАТЕЛЬ: БЛЮМ ВАЛЕНТИНА СЕРГЕЕВНА

ДИСЦИПЛИНА: МДК.02.02 ТЕХНОЛОГИЯ РАЗРАБОТКИ И ЗАЩИТЫ БАЗ  
ДАННЫХ

ГРУППЫ: ПКС 3.1-17, ПКС 3.2.-17, ПКС 2.5-18

# ЦЕЛЬ РАБОТЫ:

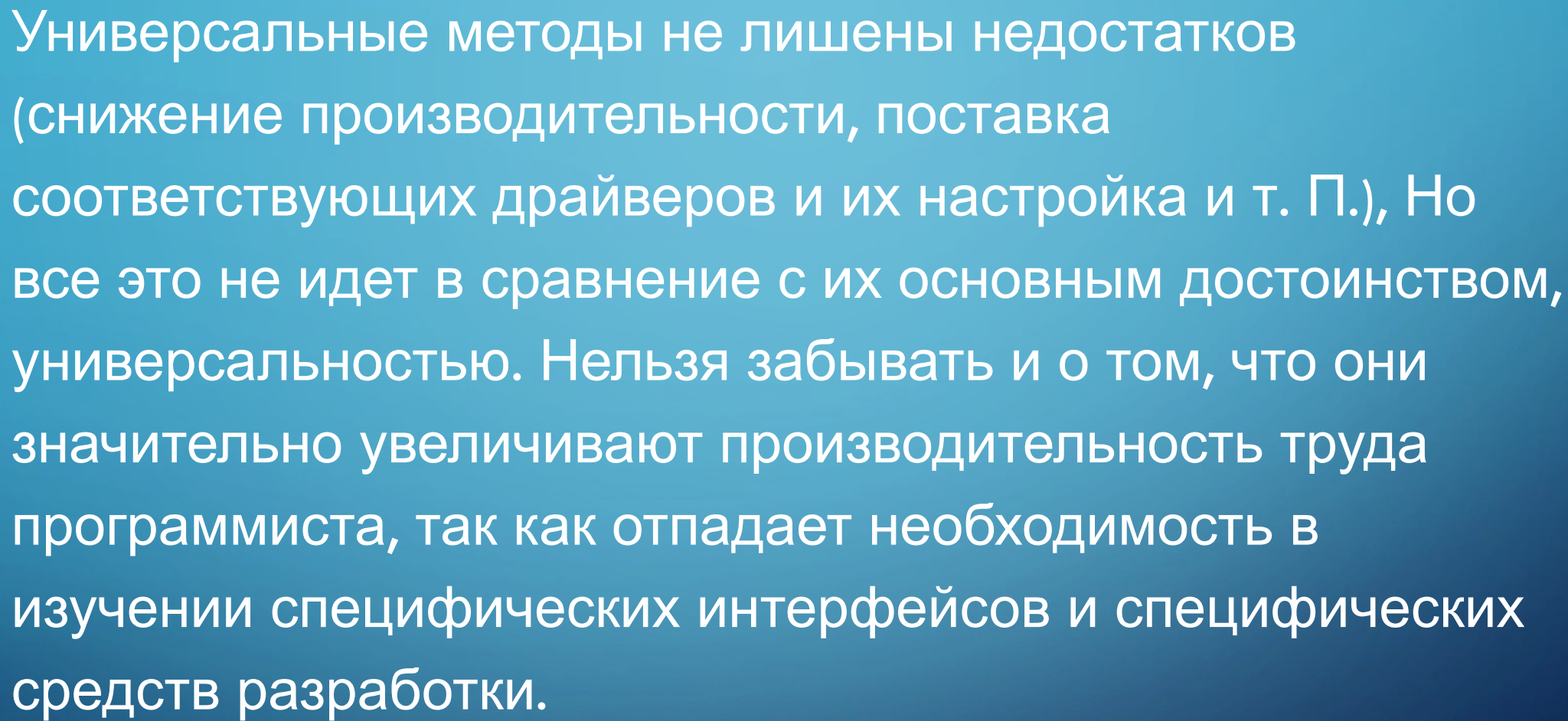
1. Познакомиться с универсальными методами доступа к данным, актуальными в настоящее время.
2. Изучить основы технологии ADO.
3. Подробно рассмотреть архитектуру ADO.NET

## ЗАДАНИЕ:

1. Написать конспект по данной презентации, ответить на вопросы и прикрепить фотоотчёт на ресурс.

Взаимодействие приложения с БД может быть организовано различными способами. Первоначально, как основные, преобладали способы, основанные на API (application programming interface) библиотек СУБД (позже, дополнительно, и COM объектов), входящих в состав клиентского программного обеспечения, устанавливаемого на компьютерах пользователей. Как правило, клиентское программное обеспечение включало в себя и собственную среду разработки прикладного программного обеспечения. Это приводило к тому, что замена СУБД требовала значительной переделки кода приложения. Как результат - в настоящее время главенствующими становятся универсальные механизмы доступа к данным.

Универсальные механизмы доступа к данным обычно реализованы в виде библиотек и дополнительных модулей (драйверов или провайдеров). Библиотеки содержат определенный стандартный набор классов, методов, параметров, и обеспечивают стандартный интерфейс доступа к данным. Дополнительные модули реализуют непосредственное обращение к функциям клиентского API конкретных СУБД. Причем, эти дополнительные модули, устанавливаются исходя из текущей потребности. Если, например, приложение работало с использованием ADO (см. Ниже) и, допустим, использовало визуальный компонент `ado table` - то теперь, при необходимости переноса БД с `mysql` на `postgres`, достаточно будет только установить соответствующий драйвер и изменить строку `connection` в параметрах компонента. Изменение кода приложения не требуется.

The background is a dark blue gradient. In the corners, there are decorative white and light blue lines that resemble circuit traces or a stylized network diagram. These lines connect to small white circles, some of which are also connected to each other, forming a sparse network structure.

Универсальные методы не лишены недостатков (снижение производительности, поставка соответствующих драйверов и их настройка и т. П.), Но все это не идет в сравнение с их основным достоинством, универсальностью. Нельзя забывать и о том, что они значительно увеличивают производительность труда программиста, так как отпадает необходимость в изучении специфических интерфейсов и специфических средств разработки.

Среди универсальных методов доступа к данным наиболее распространены в настоящее время:

ODBC - Open Database Connectivity.

OLE DB - Object Linking and Embedding Database.

ADO - ActiveX Data Objects.

BDE - Borland Database Engine.

ADO.NET - ActiveX Data Objects ТЕХНОЛОГИИ NET.

ODBC (Microsoft) - механизм взаимодействия с реляционными базами данных. Для обеспечения доступа к БД необходимы клиентская часть СУБД, ODBC-драйвер для доступа к этой СУБД (или только драйвер для некоторых СУБД) и соответствующая настройка ODBC на компьютере (обычно имя драйвера, пользователя, имя базы, пароль доступа и указание некоторых параметров драйвера). Драйвер, при выполнении приложения, загружается в адресное пространство приложения и используется для доступа к БД. Для каждой СУБД используется собственный ODBC-драйвер. ODBC API стандартизован. В визуальных средах разработки теперь достаточно помещения на форму соответствующего компонента, указания источника данных ODBC, имени таблицы и связь с базой данных установлена или через клиентскую часть СУБД или через соответствующий драйвер.



OLE DB (Microsoft) - механизм работы с самыми разнообразными источниками данных (в отличие от ODBC - где только работа с реализованными БД) на базе COM-интерфейса. OLE DB определяет набор COM-интерфейсов (Component Object Model), включающих различные сервисы однотипного доступа к различным данным (в том числе и к нереляционным БД, например, к папкам систем электронной почты или просто файлам), обеспечивая при этом поддержку работы с наборами данных и иерархическими наборами записей, подключенными непостоянно к сети. Для доступа к БД требуется установка OLE DB провайдера для СУБД (DLL СУБД загружается при выполнении приложения в его адресное пространство). Кроме того, фирмой разработан специальный провайдер (Microsoft OLE DB Provider for ODBC Drivers), который может работать не через API клиентской части СУБД, а через интерфейс ODBC API.

ADO (Microsoft) - своеобразная надстройка над OLE DB (использует библиотеки OLE DB) - дополнительный набор библиотек, содержащих COM-объекты, реализующие интерфейс доступа к данным. Этот набор библиотек первоначально включал две объектные технологии Microsoft: Data Access Objects (DAO) и Remote Data Objects (RDO) - два различных механизма доступа к локальным и удаленным базам данных соответственно.

VDE (Borland) - универсальный механизм доступа к данным, базирующийся на двух группах библиотек-драйверов (SQL Links - для серверных СУБД и ODBS Links - для серверных и автономных СУБД). Эти библиотеки позволили применять стандартный набор функций для доступа к данным dBase, Access, FoxPro, ODBC-источников, а также большинства серверных СУБД. VDE поддержан на уровне компонент в визуальных средах разработки фирмы Borland. Реализация механизма позволяет функциям приложения обращаться к функциям клиентского API или ODBC API, а также непосредственно манипулировать файлами данных некоторых СУБД. Для доступа к базе данных с помощью VDE на компьютере должны быть установлены библиотеки VDE общего назначения (обычно устанавливаются вместе со средой разработки и затем включаются в .exe файл), а также VDE-драйвер для данной СУБД.

ADO.NET (Microsoft) - технология работы с базами данных в three-tier (многоярусной архитектуре), когда соединение с базой данных устанавливается лишь на период выполнения операций с БД (как правило кратковременных). Ее появление связано с необходимостью разрешения противоречия между ростом числа обращений к БД (особенно в БД интернет-серверов) и невозможностью базы данных поддерживать неограниченное число активных соединений. ADO.NET призвана решить эти и другие проблемы и вместе с тем сохранить удобство и простоту программирования. ADO.NET, в ее современном виде - иерархический набор объектов, построенный в соответствии с новой идеологией базовых библиотек классов и смены протоколов COM на .NET.

# ОСНОВЫ ТЕХНОЛОГИИ ADO.NET

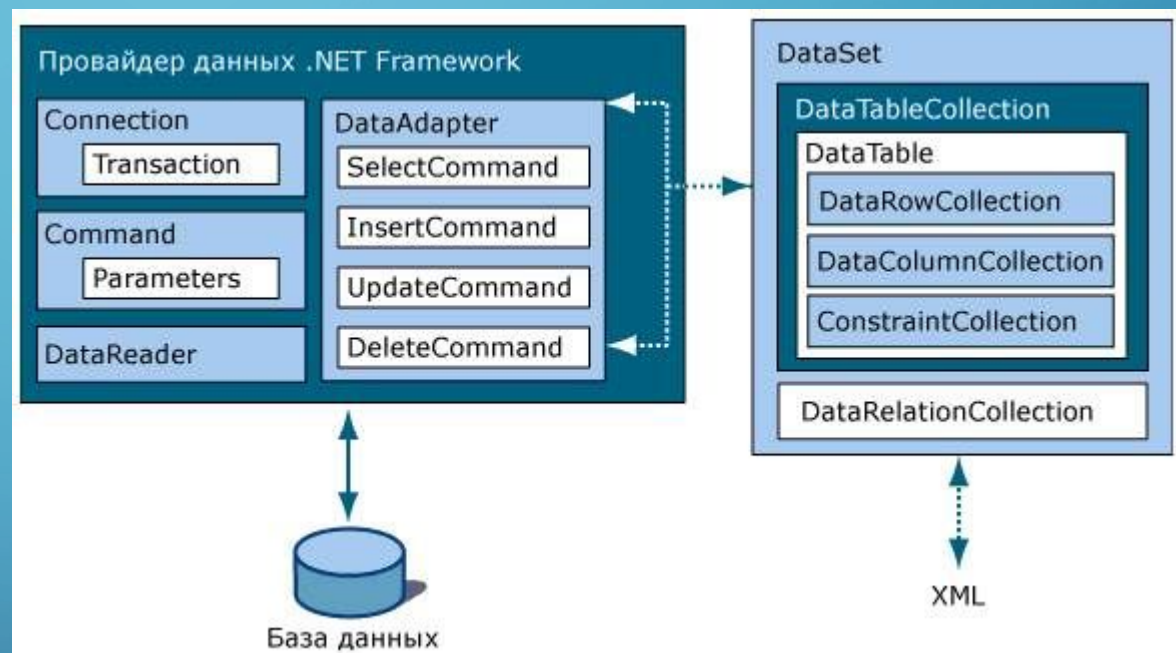
В ADO.NET используется модель доступа, при которой соединение с БД устанавливается лишь на короткое время, когда необходимо проводить какую либо операцию по обмену информацией с базой данных. Естественно, такая технология не является эффективной для режимов работы, когда приложение проводит частые и объемные изменения содержания записей в БД. Для данного случая, более отвечающими требованиям, как и обычно, являются традиционные методы доступа - ODBC, ADO.

# АРХИТЕКТУРА ADO.NET

Функциональность ADO.NET, предоставляемая разработчикам приложений, аналогична функциональности ADO, предоставляемой разработчикам COM, различие заключается в способе манипулирования данными.

Компоненты ADO.NET позволяют отделить доступ к данным от их обработки. Разделение выполняется с помощью двух основных компонентов ADO.NET:

1. DataSet
2. провайдер данных .NET Framework



# ПРОВАЙДЕР (ПОСТАВЩИК) ДАННЫХ .NET FRAMEWORK ПРЕДСТАВЛЯЕТ СОБОЙ НАБОР КОМПОНЕНТОВ, ЭТО:

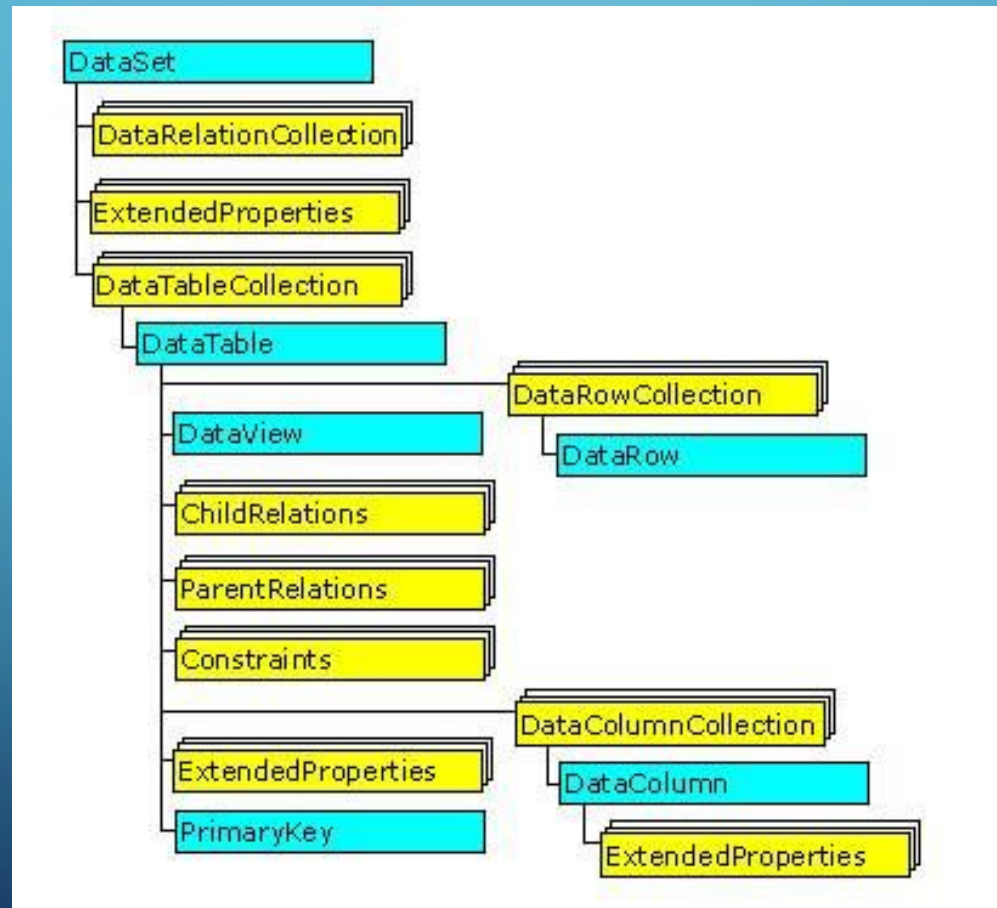
- `Connection` - компонент для подключения к источнику данных;
- `Command` - компонент для обеспечения выполнения команд в БД, изменения данных, запуска хранимых процедур, отправки и получения значений параметров;
- `DataReader` - компонент для создания высокопроизводительного потока данных от источника данных;
- `DataAdapter` - компонент для обеспечения взаимодействия объекта `DataSet` с источником данных.

На [рисунке](#) изображен один провайдер данных. На самом деле их пять:

- Провайдер SQL Server .NET. Он обеспечивает связь с SQL серверами. Он имеет библиотеки, позволяющие выполнять прямое взаимодействие с SQL серверами через протокол Tabular Data Stream (TDS).
- Провайдер данных OLE DB .NET необходим для организации взаимодействия с отличными от SQL Server базами данных (Oracle, DB2). Провайдер использует OLE DB для соответствующих баз данных.
- ODBC .NET. Предоставляет доступ к любому источнику данных, имеющему драйвер ODBC.
- Провайдер Oracle. Предоставляет оптимизированный под Net доступ к Oracle.
- Провайдер SQL ServerCE для доступа к базам данных SQL Server CE из устройств, базирующихся на Windows CE и расположенных в управляемой среде.



Данные, полученные с помощью компонент провайдера, либо непосредственно обрабатываются, либо помещаются в объект dataset.



- Объект `DataSet` является основой поддержки многоуровневой архитектуры данных и механизмов, позволяющих работать с отключением от источников данных. `DataSet` представляет собой находящийся в памяти объект со снимком реальных данных БД. Он может включать таблицы (`DataTableCollection`), отношения между таблицами (`DataRelationCollection`). Для этого `DataSet` содержит ноль или более таблиц, представленных объектами `DataTable`, ноль или более отношений - объекты `DataRelation` (ключевая информация таблиц, хранящаяся в виде имя отношения, имена связанных таблиц и столбцов). Кроме того, как сам `DataSet`, так и некоторые его дочерние объекты имеют свойства `ExtendedProperties` - коллекция `PropertyCollection`, в которых можно сохранять различную информацию (в том числе и SQL операторы). Коллекция `ExtendedProperties` хранится со сведениями о схеме `DataSet` (как и объекты `DataTable` и `DataColumn`).

## Вопросы:

- 1. Дополните недостатки универсальных методов доступа к данным описанные в презентации.
- 2. Исходя из каких потребностей появилась технология ADO?
- 3. Можно ли провести частые и объёмные изменения содержания записей в БД для технологии ADO.NET. Почему?