**21. Классы для работы с базами данных**

Взаимодействие с хранилищами электронных структурированных материалов может осуществляться в Java при помощи main interfaces.

В Java для работы с базами данных используются различные классы и интерфейсы, основанные на двух основных подходах: через JDBC (Java Database Connectivity) и через JPA (Java Persistence API). Ниже представлены ключевые классы для каждого из подходов:

**JDBC**

1. **DriverManager**: Этот класс управляет списком драйверов баз данных. Он используется для установки соединения с базой данных.
2. **Connection**: Представляет соединение с базой данных. С его помощью можно создавать объекты Statement для выполнения запросов.
3. **Statement**: Используется для выполнения статических SQL-запросов без параметров. Задействован для доступа к БД при решении общих вопросов. Активно применяется на практике со static SQL. А именно – выражениями во время функционирования утилиты. Не принимает параметры.
4. **PreparedStatement**: Подкласс Statement, позволяющий выполнить SQL-запросы с одним или несколькими входными параметрами. Интерфейс, напоминающий предыдущий.
5. **CallableStatement**: Подкласс PreparedStatement, используемый для выполнения хранимых процедур. Помогает заполучить доступ к разнообразным процедурам «хранилищ структурированных данных». Как и предыдущий вариант, принимает параметры в процессе функционирования контента.
6. **ResultSet**: Представляет результат выполнения запроса. Позволяет построчно читать данные из таблицы.

**JPA**

**Классы**:

1. **Query**: Используется для выполнения запросов, которые могут быть как простыми SQL-запросами, так и запросами на языке JPQL (Java Persistence Query Language).

**Интерфейсы**:

1. **EntityManagerFactory**: Используется для создания EntityManager. Обычно создаётся один раз на приложение.
2. **EntityManager**: Управляет персистентными сущностями (такими как объекты и их данные).
3. **EntityTransaction**: Управляет транзакциями для объектов, управляемых EntityManager.
4. **CriteriaBuilder**: Используется для создания критериев запросов для поиска и выборки данных, что позволяет строить типобезопасные SQL-запросы в рантайме.

Эти классы и интерфейсы предоставляют мощные инструменты для работы с базами данных в Java, позволяя разработчикам выбирать подходящий уровень абстракции в зависимости от задачи.ф