

Java Database Connectivity (JDBC)

Универсальное API для доступа
к данным

- JDBC API состоит из интерфейсов и классов, используемых для доступа к данным, независимо от их источника

`Connection, Statement, ResultSet`

- Использование конкретного источника данных (БД) и драйвера можно указывать в настройках приложения, чтобы исходный код не зависел от типа, имени и расположения базы данных

```
driver=oracle.jdbc.OracleDriver
url=jdbc:oracle:thin:@localhost:1521:stud
user=o01
password=o01
```

- Конкретный драйвер БД реализует все JDBC интерфейсы и набор их функций

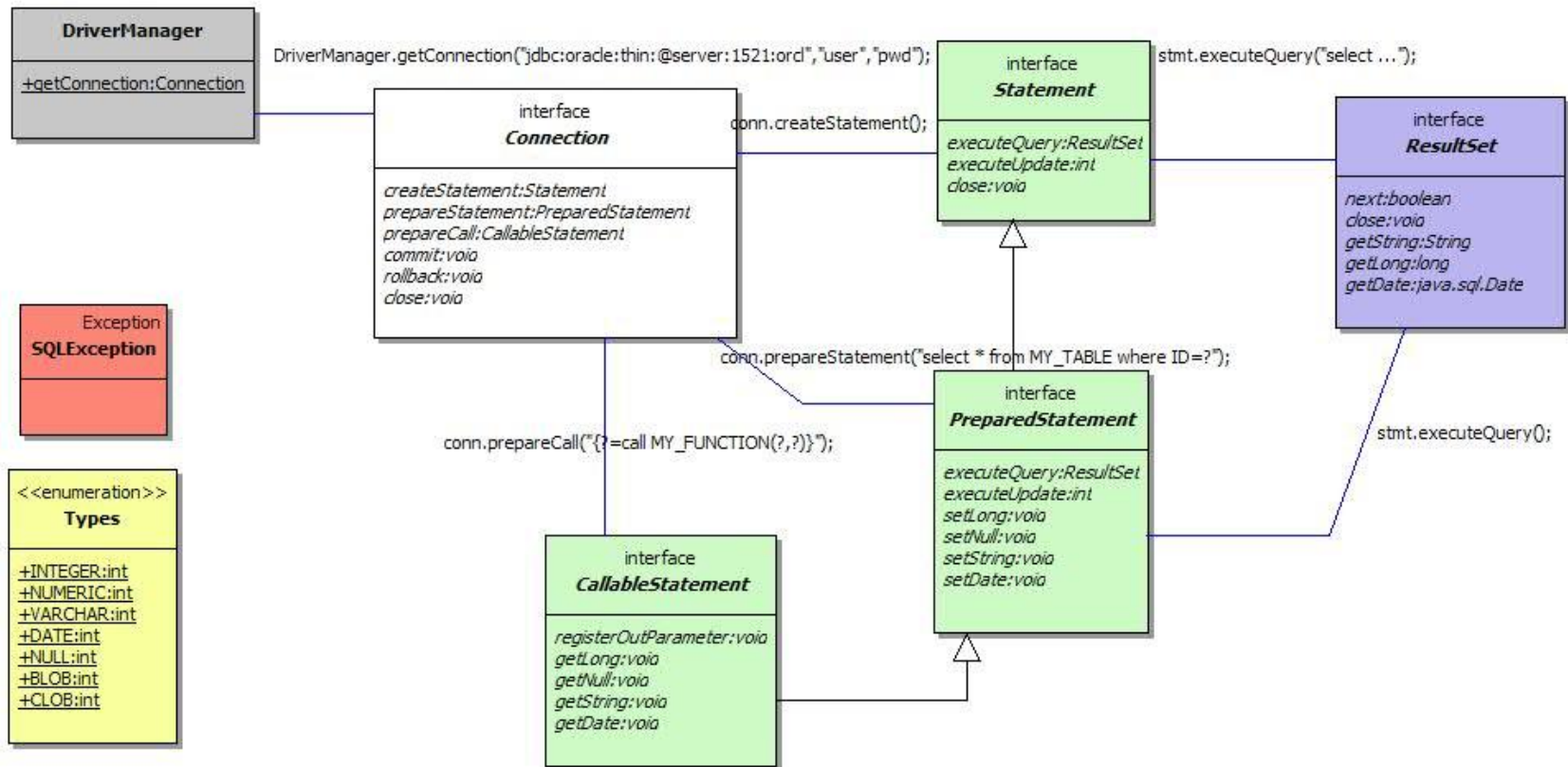
`OracleConnection, OracleStatement, OracleResultSet`

- Драйвера JDBC делятся на типы (1-4) по уровню зависимости от используемой платформы. Большинство современных драйверов 4 типа написаны целиком на Java, не зависят от платформы и не требуют установки дополнительных библиотек (например клиента Oracle)
- Возможность использования JDBC-ресурсов в распределенных транзакциях.
- Независимость от СУБД кончается там, где используется специфический для СУБД SQL-запрос, или нестандартная функция JDBC.

```
select * from TABLE1 where ROWNUM<5
((OracleResultSet)resultSet).getOracleObject(1)
```

□ Пример: `org.mai806.jdbcsample.QuerySample`

Основные классы JDBC



Подключение к БД

```
/* ===== Подключение к MS SQL Server ===== */
// Загрузка драйвера
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
// Соединение с базой данных
Connection connection = DriverManager.getConnection(
    "jdbc:sqlserver://localhost:1433;databaseName=o01;", "sa", "123");

/* ===== Подключение к Oracle ===== */
// Загрузка драйвера
Class.forName("oracle.jdbc.OracleDriver");
// Соединение с базой данных
Connection connection = DriverManager.getConnection(
    "jdbc:oracle:thin:@localhost:1521:orcl", "o01", "o01");
```

Параметры соединения:

1. Класс JDBC драйвера для СУБД
`com.microsoft.sqlserver.jdbc.SQLServerDriver`
2. URL – содержит протокол, имя сервера, порт и имя экземпляра БД
`jdbc:sqlserver://localhost:1433;databaseName=o01`
`jdbc:драйвер://сервер:порт;databaseName=экз.БД`
3. Имя пользователя (login)
`sa`
4. Пароль (password)
`sa`

Использование транзакций

- `java.sql.Connection`:
 - `getAutoCommit()/setAutoCommit(boolean)`
 - `commit()`
 - `rollback()`
 - `setTransactionIsolation()`

`autoCommit=true` – Транзакция начинается и заканчивается с каждой операцией с базой данных

`autoCommit=false` – Ручное управление транзакциями

Уровни изоляции:

`TRANSACTION_READ_UNCOMMITTED`

`TRANSACTION_READ_COMMITTED`

`TRANSACTION_READ_REPEATABLE_READ`

`TRANSACTION_READ_SERIALIZABLE`

□ Пример: `org.mai806.jdbcsample.TransactionSample`

□ Пример использования хранимой процедуры: `org.mai806.jdbcsample.StoredProcedureSample`