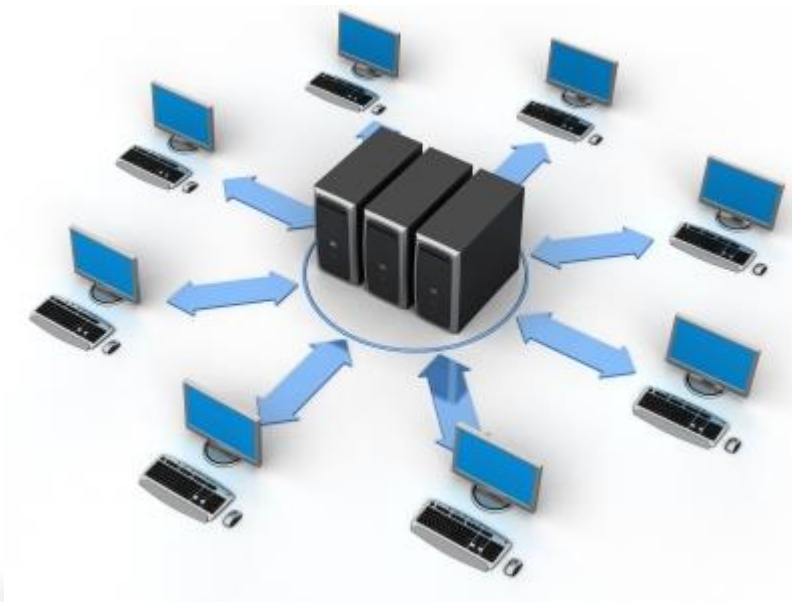


# JDBC



Универсальный API для  
доступа к данным

Евгений Беркунский,  
кафедра ИУСТ, НУК  
<http://www.berkut.mk.ua>  
[eugeny.berkunsky@gmail.com](mailto:eugeny.berkunsky@gmail.com)

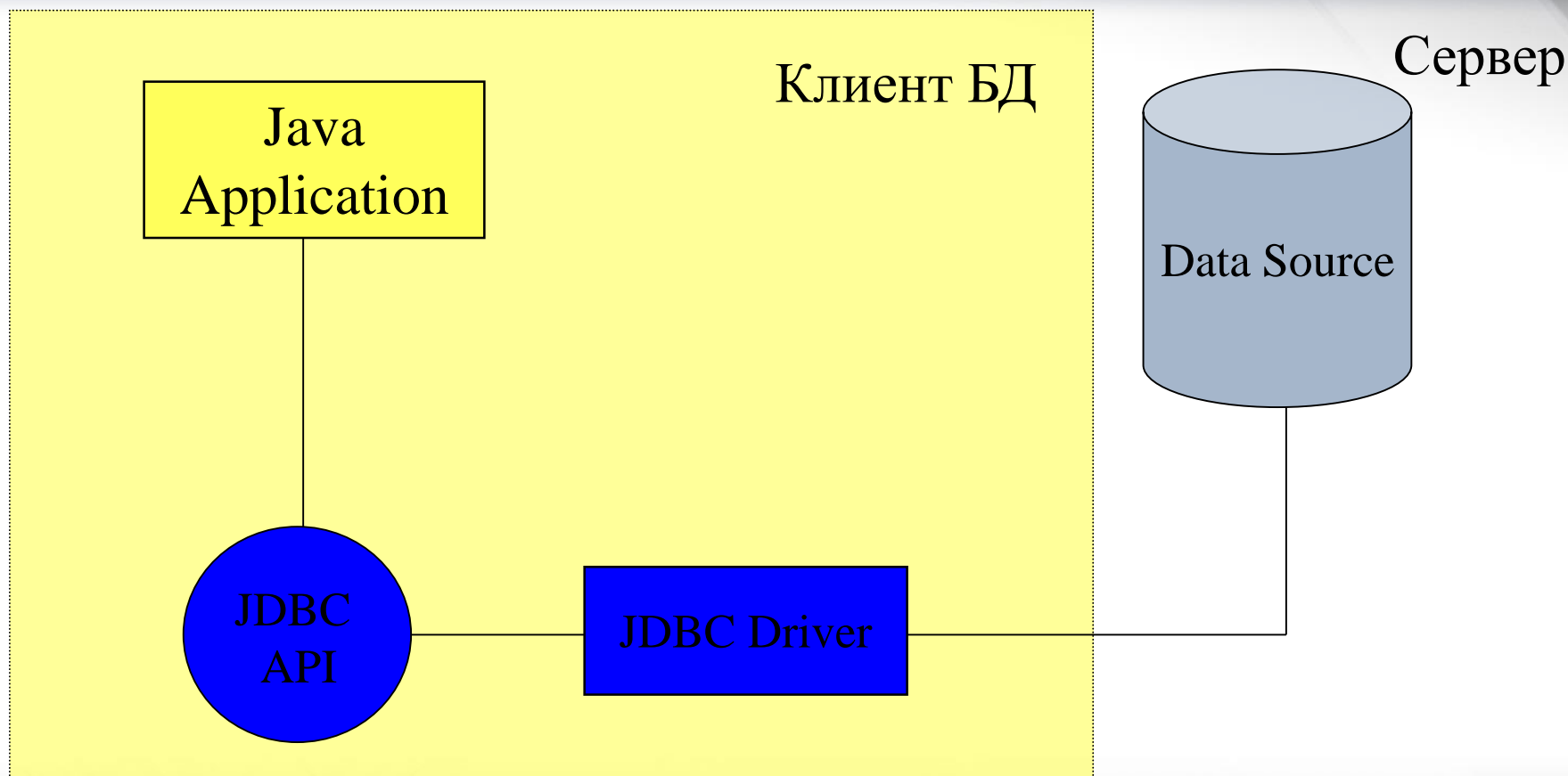
# JDBC API

- JDBC (**Java Database Connectivity**) API позволяет программам на Java подключаться к базам данных
- **Доступ** к БД одинаков для всех вендоров БД
- JVM использует **JDBC драйвер** для перевода обобщенных JDBC команд в команды БД конкретного вендора
- Существует **четыре типа** JDBC-драйверов
  - Мы будем рассматривать Тип 4 (Pure Java)...

# Pure Java Driver (Type 4)

- Драйверы этого типа преобразуют команды JDBC API в прямые вызовы по сети с использованием **специфичных для вендора сетевых протоколов** используя прямые сетевые подключения к БД
- Это наиболее **эффективный** метод доступа к БД, как по производительности, так и по времени разработки
- Так просто развернуть приложение для **БД**
- Большинство **вендоров** БД предоставляют Pure Java JDBC драйверы к своим БД.
- Кроме того, существуют драйверы от независимых разработчиков

# Доступ к БД из Java-приложения



# JDBC API

- 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=com.mysql.jdbc.Driver  
url= jdbc:mysql://localhost:3306/example  
user=eugeny  
password=123
```

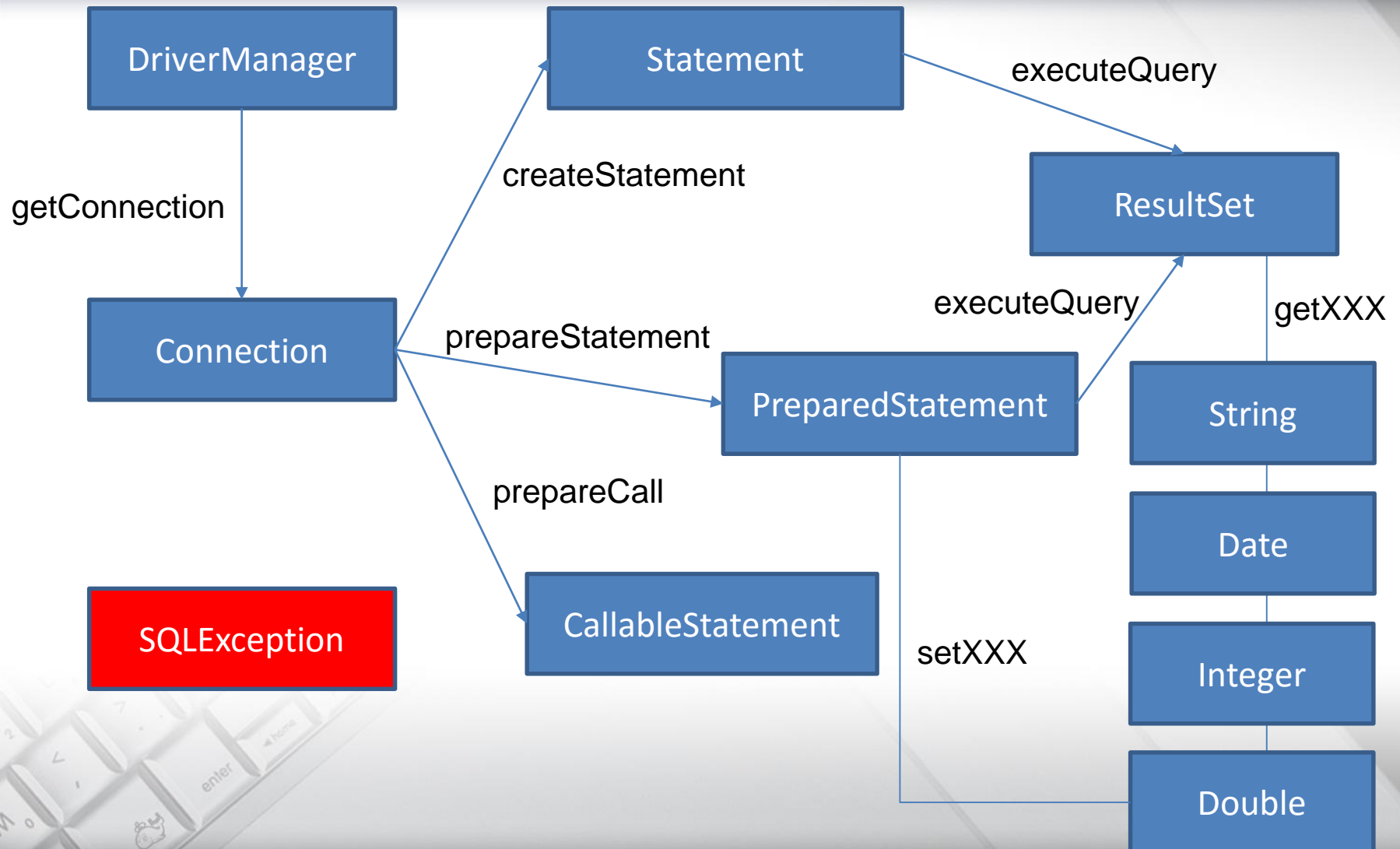
- Конкретный драйвер БД реализует все JDBC интерфейсы и набор их функций

`MySQLConnection, MySQLStatement, MySQLResultSet`

# Типичный порядок использования JDBC

1. Загрузить драйвер БД
2. Получить подключение (connection)
3. Создать и выполнить запросы (SQL запросы)
4. Использовать наборы результатов (result sets), для навигации по результатам запроса и извлечения данных из них
5. Закрывать соединение

# Основные классы и интерфейсы JDBC



# 1. Загрузка драйвера БД

В литературе (и Internet) написано, что  
загружать драйвер нужно таким вызовом:  
**Class.forName (“com.mysql.jdbc.Driver”);**

*Однако, официальная документация Oracle сообщает:  
Applications no longer need to explicitly load JDBC drivers using  
Class.forName(). Existing programs which currently load JDBC  
drivers using Class.forName() will continue to work without  
modification*

<http://docs.oracle.com/javase/8/docs/api/java/sql/DriverManager.html>



## 2. Подключение к БД

```
/* ===== Подключение к MySQL Server ===== */  
// Соединение с базой данных  
Connection connection = DriverManager.getConnection(  
    "jdbc:mysql://localhost:3306/example","eugeny","123");  
  
/* ===== Подключение к PostgreSQL ===== */  
// Соединение с базой данных  
Connection connection = DriverManager.getConnection(  
    "jdbc:postgresql://localhost:5432/proba","eugeny","123");
```

Параметры соединения:

1. URL – содержит протокол, имя сервера, порт и имя экземпляра БД  
*jdbc:драйвер://сервер:порт*
2. Имя пользователя (login)
3. Пароль (password)

# 3. Создание и выполнение запросов

Добавление, удаление, обновление данных: **executeUpdate**

```
Statement stmt = con.createStatement();  
stmt.executeUpdate ( "INSERT INTO COFFEES " + "VALUES  
    ('Colombian', 101, 7.99, 0, 0)");  
stmt.executeUpdate ( "INSERT INTO COFFEES " + "VALUES  
    ('French_Roast', 49, 8.99, 0, 0)" );  
stmt.executeUpdate ( "INSERT INTO COFFEES " + "VALUES  
    ('Espresso', 150, 9.99, 0, 0)" );  
stmt.executeUpdate ( "INSERT INTO COFFEES " + "VALUES  
    ('Colombian_Decaf', 101, 8.99, 0, 0)" );  
stmt.executeUpdate ( "INSERT INTO COFFEES " + "VALUES  
    ('French_Roast_Decaf', 49, 9.99, 0, 0)" );
```

Выборка данных: **executeQuery**

```
ResultSet rs = stmt.executeQuery (  
    "SELECT COF_NAME, PRICE FROM COFFEES");
```

*Примечание: для параметризованных запросов лучше использовать preparedStatement*

# 4. Извлечение и обработка результатов

Выборка данных: **executeQuery**

```
ResultSet rs = stmt.executeQuery (  
    "SELECT COF_NAME, PRICE FROM COFFEES");  
while (rs.next()) {  
    String s = rs.getString("COF_NAME");  
    double price = rs.getDouble("PRICE");  
    System.out.println (s + " " + price);  
}
```

# 5. Закрyтіє соединенія

```
connection.close();
```





НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ  
КОРАБЛЕБУДУВАННЯ  
ІМЕНІ АДМІРАЛА МАКАРОВА

# Демонстрація



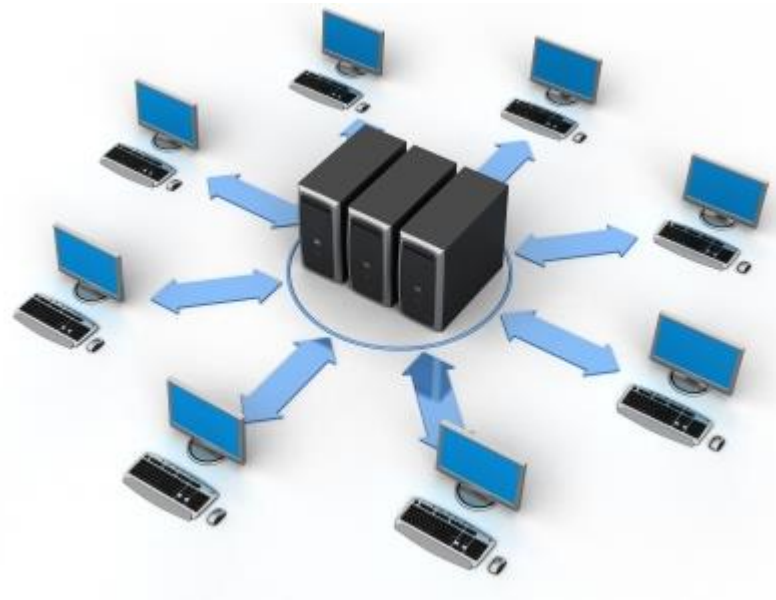


НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ  
КОРАБЛЕБУДУВАННЯ  
ІМЕНІ АДМІРАЛА МАКАРОВА



# Технологии распределенных систем и параллельных вычислений

## Лекция 2: JDBC



Универсальный API для  
доступа к данным

Евгений Беркунский,  
кафедра ИУСТ, НУК  
<http://www.berkut.mk.ua>  
[eugeny.berkunsky@gmail.com](mailto:eugeny.berkunsky@gmail.com)