

Введения в JDBC

№ урока: 1 **Курс:** JDBC & Hibernate

Средства обучения: JDK, IntelliJ Idea

Обзор, цель и назначение урока

Рассмотрение основных понятий JDBC. Выполнения простых CRUD-операций с помощью Java-кода.

Изучив материал данного занятия, учащийся сможет:

- Интегрировать БД в IntelliJ Idea.
- Работать с БД в IntelliJ Idea
- Заполнять таблицы с помощью JDBC.
- Изменять существующие данные таблиц с помощью JDBC.
- Создавать выборки нужных данных с помощью JDBC.
- Удалять данные с помощью JDBC.

Содержание урока

1. Рассмотрение основных понятий JDBC.
2. Изучение 2-х звенной архитектуры JDBC.
3. Изучение 3-х звенной архитектуры JDBC.
4. Подключение JDBC-драйвера и получения Connection.
5. Рассмотрения основных запросов манипуляции данными с помощью JDBC
6. Рассмотрения статического Statement.
7. Рассмотрение PreparedStatement.

Резюме

- **JDBC (Java DataBase Connectivity — соединение с базами данных в Java)** — платформенно-независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД, реализованный в виде пакета `java.sql`, входящего в состав Java SE.
- **JDBC** основан на концепции так называемых драйверов, позволяющих получать соединение с базой данных по специально описанному URL. Драйверы могут загружаться динамически (во время работы программы). Загрузившись, драйвер сам регистрирует себя и вызывается автоматически, когда программа требует URL, содержащий протокол, за который драйвер отвечает.
- **Преимущества JDBC:**
 - Единый интерфейс для работы СУБД
 - Независимость от операционной системы (Windows, Linux, MacOS).
 - Для работы достаточно скачать JDBC драйвер (jar-файл), нет необходимости устанавливать дополнительный софт
- **Недостатки:**
 - Переход от одной СУБД к другой (с существующим программным кодом) не всегда легко удается выполнить
 - Различие в синтаксисе SQL-запросов.
- **Двухзвенная архитектура** часто используется в desktop приложениях (не Web). База может находиться как на том же компьютере, так и на другом.

Вся логика по управлению с базой данных (транзакции, распределение нагрузки и пр.) ложится на программный код.

Нет централизованного управления доступом в базе данных.

Часто изменения по работе необходимо производить в программном коде.

Невысокая производительность.

- **Трехзвенная** (в некоторых случаях *многозвенная*) **архитектура** представляет собой дальнейшее совершенствование технологии "клиент – сервер". Рассмотрев архитектуру "клиент – сервер", можно заключить, что она является 2-звенной: первое звено – клиентское приложение, второе звено – сервер БД + сама БД. В трехзвенной архитектуре вся бизнес-логика (деловая логика), ранее входившая в клиентские приложения, выделяется в отдельное звено, называемое сервером приложений. При этом клиентским приложениям остается лишь пользовательский интерфейс. Так, в качестве клиентского приложения в описанном выше примере выступает Web-браузер. Теперь при изменении бизнес-логики более нет необходимости изменять клиентские приложения и обновлять их у всех пользователей. Кроме того, максимально снижаются требования к аппаратуре пользователей. Логика по управлению доступом к базе данных ложится на сервер приложений. Программный код лишь осуществляет логику по работе с данными (запросы, обработка). База данных находится на отдельном компьютере с хорошей производительностью. Возможность централизованного увеличения производительности (кеширование, пулы соединений, оптимизация и пр.).
- Для того, чтобы пользоваться **JDBC** необходимо зарегистрировать JDBC-драйвер и получить Connection.
- В JDBC есть два вида доступа к запросам: статический Statement и PreparedStatement. В статическом мы пишем полностью наш запрос. А в PreparedStatement мы с помощью методов подставляем данные
- **execute()** – метод интерфейса Statement для выполнения простого запроса, кроме выборки.
- **executeUpdate()** – метод интерфейса Statement в основном для вставки и удаления.
- **executeQuery()** – метод интерфейса Statement для получения объекта ResultSet, с помощью которого мы сможем сделать выборку.
- После каждого запроса Connection и Statement нужно закрыть

Закрепление материала

- Что такое JDBC?
- Как работает схема JDBC?
- В чем разница между 2-х звенной архитектурой и 3-х звенной?
- Какую архитектуру лучше использовать в web-приложениях?
- Какие преимущества в JDBC?
- Какие недостатки в JDBC?
- Что такое JDBC-драйвер?
- Что такое Connection в JDBC?
- Что такое Statement в JDBC?
- В чем разница между PreparedStatement и обычным Statement?

Дополнительное задание

Задание

Создать базу данных в Workbench и подключить к IntelliJ Idea и создать тестовую таблицу. Заполнить ее данными с помощью запросов MySQL в IntelliJ Idea. Используя JDBC написать пример выполнения всех запросов.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Используя дополнительное задания все вопросы записать в текстовом файле с новой строки каждый и используя потоки ввода-вывода считать с файла все запросы и выполнить.

Задание 3

Создайте базу данных с именем "MyJoinsDB" используя Workbench.

Задание 4

Используя Workbench в данной базе данных создайте 3 таблицы,

В 1-й таблице содержатся имена и номера телефонов сотрудников компании.

Во 2-й таблице содержатся ведомости о зарплате и должностях сотрудников: главный директор, менеджер, рабочий.

В 3-й таблице содержится информация о семейном положении, дате рождения, и месте проживания.

Задание 5

Используя IntelliJIdea и JDBC сделайте выборку при помощи JOIN's для таких заданий:

- 1) Получите контактные данные сотрудников (номера телефонов, место жительства).
- 2) Получите информацию о дате рождения всех холостых сотрудников и их номера.
- 3) Получите информацию обо всех менеджерах компании: дату рождения и номер телефона.

Рекомендуемые ресурсы

JDBC

<http://www.javenue.info/post/java-jdbc-api>

2-х и 3-х звенная архитектура

<http://www.4stud.info/networking/lecture5.html>