

Введения в JDBC

№ урока: 3 **Курс:** JDBC & Hibernate

Средства обучения: JDK, IntelliJ Idea, Maven

Обзор, цель и назначение урока

Введение в Hibernate-ORM. Рассмотрение проекта с помощью сборщика Maven. Настройка проекта, создание первого приложения на Hibernate.

Изучив материал данного занятия, учащийся сможет:

- Получит базовые знания о ORM, JPA, Hibernate.
- Научиться создавать Maven-проект.
- Настраивать проект под Hibernate
- Писать первые приложения используя Hibernate-ORM

Содержание урока

1. Рассмотрение ORM.
2. Рассмотрение JPA.
3. Рассмотрение Hibernate
4. В чем разница между JDBC & Hibernate
5. Достоинства и недостатки Hibernate
6. Создание Maven-проекта
7. Настройка проекта под Hibernate
8. Написание первого приложения на Hibernate

Резюме

- **ORM(Object Relation Mapping)** – технология, которая работает с таблицами, как с объектами, которая напрямую к БД не обращается (хоть и это возможно).
- **JPA (Java Persistence API)** – спецификация, документ, в котором описаны правила и API для реализации принципов ORM для Java (аннотации, настройки, подход).
- **Hibernate** – фреймворк между БД и приложением, в котором не нужно создавать прямые SQL запросы через JDBC.
Может применяться в любом типе приложения (web/desktop/Spring и т.д.).
Основная цель – работа с таблицами БД как с объектами.
Легче создавать правильный абстрактный уровень для приложения (применять принципы ООП).
На низком уровне Hibernate выполняет запросы через JDBC.
- **Достоинства Hibernate:**
 - Более высокоуровневый (ООП-ориентированный) код без ручной работы JDBC API
 - Переносимость (перенос на другую БД в основном требует небольших изменений).
 - Много примеров и документации.
 - Работа со всеми популярными СУБД.
 - Работа с NoSQL решениями.
 - Интеграция с фреймворками, IDE
 - Удобство (для тех, кто разобрался с основами)
 - Можно не углубляться в SQL
 - Множество настроек (транзакции, кэширование, генерация таблиц и пр.).
 - Обновления, улучшения.

- **Недостатки Hibernate:**
 - В крупных проектах возможна потеря производительности (тяжеловесность), при неправильных настройках.
 - Дополнительно нужно изучать API для Hibernate и JPA.
 - Сложность для новичков, множество настроек, специфики, аннотаций.
 - Меньше контроля (доступ к БД на низком уровне).
 - Привязка к своему API (все меньше и меньше).
- **hibernate.cfg.xml** — файл конфигурации Hibernate
- **@Entity** — аннотация, которая применяется к классам, что обозначает сущность таблиц
- **@Id** — аннотация, которая применяется к полям, что обозначает первичный ключ.
- **Maven** — система автоматической сборки, построенная на принципах Apache Ant и Apache Gradle, но предоставляющая DSL на языке Groovy вместо традиционной XML-образной формы представления конфигурации проекта.

Закрепление материала

- Что такое ORM?
- Что такое JPA?
- Что такое Hibernate?
- В чем разница между JDBC и Hibernate?
- В чем достоинства и недостатки Hibernate?
- Что такое Maven?
- Как создать структуру Gradle-проекта?
- Как создать элементарное приложение на Hibernate?

Дополнительное задание

Задание

Создать новую базу данных с помощью MySQLWorkbench. Создать Maven-проект и настроить его правильно. Создать файл с настройками hibernate.cfg.xml. И создать класс Animal(int age, String name, boolean tail) с методами get и set, как сущность к нашей таблице. И с помощью hibernate создать новую таблицу с начальным значением.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Разобраться со всеми property в файле hibernate.cfg.xml

Задание 3

Найти документацию на сайте <http://hibernate.org/> и прочитать первый раздел “Getting Started Guide”.

Рекомендуемые ресурсы

Start with Hibernate

http://docs.jboss.org/hibernate/orm/5.2/quickstart/html_single/