

Что такое Hibernate ?

Самая главная задача Hibernate — сделать так, чтобы разработчик думал в терминах объектов. Причем не просто на уровне таблиц — на уровне отношений между таблицами. Т.е. если тот же студент учится в группе, то связь должна быть на уровне объектов, а не на уровне поля `groupId` у студента. Иными словами — загруженный объект «Студент» должен иметь ссылку на объект «Группа». Мало того — для получения данных о группе объект «Студент» не надо делать явный запрос к базе данных. Эту функцию должен взять на себя Hibernate — в коде на Java мы просто должны вызывать метод `student.getGroup()`

Все пакеты, подобные Hibernate, предназначены именно для такой цели — дать возможность разработчику думать в терминах объектов и отношений между ними. Причем эти отношения могут быть разными — один-к-одному, один-ко-многим, многие-к-одному, многие-ко-многим.

Что еще важно отметить — вы можете создавать иерархии классов. Да-да, именно наследование, которое так вдохновляет поборников объекто-ориентированного программирования. Конечно, не все абсолютно шикарно и гладко, но в общем и целом достаточно прилично.

Hibernate не должен являться заменой DAO. Да в общем-то и не очень это получится, из-за необходимости вызова хранимых процедур. Он просто позволяет вам удобно общаться с базой данных, создавая вместо вас тексты запросов.

Поначалу все эти сложности покажутся излишними — ну в конце концов, что нам стоит написать несколько SQL-запросов на получение данных — добавление, запрос, исправление и удаление. Но если у вас таблиц не 3-4, а 500-600 ? И если вам потребуется добавить какое-то поле и связь между таблицами ? Такого рода исправления становятся не тривиальной задачей, которая требует много времени и последующего тестирования.

С другой стороны обольщаться не стоит — Hibernate не так уж и здорово решает абсолютно все задачи. Например, сложные запросы, которые часто требуются для создания головомомных отчетов, часто проще и эффективнее сделать на SQL. Hibernate позволяет это сделать — в нем можно создать обычный SQL-запрос. Также массовые изменения в нем делаются не очень эффективно.

Hibernate существенно облегчает работу с базой данных и значительно упрощает код Вашего приложения.

Для решения своих задач Hibernate должен получить от разработчика следующую информацию:

- Параметры соединения с базой данных. В общем-то от этого никуда нам не уйти. Пакет пакетом, но соединение с базой надо как-то указывать
- Описание отображения классов на таблицы. Данное описание включает в себя связь между колонкой в таблице и полем в классе. Если вспомнить тех же студентов, то для каждого поля в классе `Student` у нас было поле в таблице. Вот это нам и надо будет сделать.
- Описание отношений между классами. Классы для участия в ORM должны удовлетворять критериями JPA

JPA (Java Persistence API) это спецификация Java EE и Java SE, описывающая систему управления сохранением java объектов в таблицы реляционных баз данных в удобном виде. Hibernate одна из самых популярных открытых реализаций последней версии спецификации (JPA 2.1).