**ORM технология. Hibernate и JPA.**

Собеседование по Java EE — Hibernate Framework <http://javastudy.ru/interview/jee-hibernate-questions-answers/>

Hibernate: ORM вообще и в Java в частности <https://www.kv.by/archive/index2008401108.htm>

Поскольку без поддержки баз данных не обходится практически ни одно современное приложение, то вопросам взаимодействия с ними уделяется пристальное внимание во всех учебниках по программированию. К сожалению, о такой полезной вещи, как ORM, в учебниках описывают редко.

* **ORM технология**

Большая часть языков программирования, которые используются сегодня в индустрии программирования, относятся к объектно-ориентированным языкам. Это значит, что программист, создавая программы на них, оперирует с объектами - некоторыми абстрактными сущностями, имеющими некоторые свойства и позволяющими применять по отношению к себе некоторые методы. При всём при этом базы данных, которые используются для хранения информации об этих объектах, являются реляционными.

При создании приложения на Java и работы с базой данных ,необходимо понимать крайне важную деталь: существует огромная разница между объектной моделью и реляционной.

СУБД даёт нам информацию в табличном формате, в то время, как Java даёт информацию в виде некоего графа объектов.

ORM-решения имеют много преимуществ, по сравнению с "ручной" работой с базами данных в проекте. Они позволяют оптимизировать количество запросов к базе данных, но при этом избежать загрузки избыточных на данный момент для приложения данных. Кроме того, благодаря единому для всех используемых СУБД API-интерфейсу, который предоставляет ORM-фреймворк, в случае необходимости смены СУБД по желанию заказчика или просто из-за чрезмерного роста количества данных очень легко можно перейти с одной СУБД на другую, поскольку все SQL-диалекты уже реализованы в фреймворке его разработчиками, и программисту, использующему данный фреймворк.

* **Hibernate и 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. Сама Java не содержит реализации JPA, однако есть существует много реализаций данной спецификации от разных компаний (открытых и нет). Это не единственный способ сохранения java объектов в базы данных (ORM систем), но один из самых популярных в Java мире.

Hibernate — библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения (object-relational mapping — ORM). Она представляет собой свободное программное обеспечение с открытым исходным кодом (open source), распространяемое на условиях GNU Lesser General Public License. Данная библиотека предоставляет легкий в использовании каркас (фреймворк) для отображения объектно-ориентированной модели данных в традиционные реляционные базы данных. Hibernate совместима с JSR-220/317 и предоставляет стандартные средства JPA.

Hibernate является одним из самых востребованных ORM фреймворков для Java, потому что:

1. Hibernate устраняет множество повторяющегося кода, который постоянно преследует разработчика при работе с JDBC. Скрывает от разработчика множество кода, необходимого для управления ресурсами и позволяет сосредоточиться на бизнес логике.
2. Hibernate поддерживает XML так же как и JPA аннотации, что позволяет сделать реализацию кода независимой.
3. Hibernate предоставляет собственный мощный язык запросов (HQL), который похож на SQL. Стоит отметить, что HQL полностью объектно-ориентирован и понимает такие принципы, как наследование, полиморфизм и ассоциации.
4. Hibernate — широко распространенный open source проект. Благодаря этому доступны тысячи открытых статей, примеров, а так же документации по использованию фреймворка.
5. Hibernate легко интегрируется с другими Java EE фреймворками, например, Spring Framework поддерживает встроенную интеграцию с Hibernate.
6. Hibernate поддерживает ленивую инициализацию используя proxy объекты и выполняет запросы к базе данных только по необходимости.
7. Hibernate поддерживает разные уровни cache, а следовательно может повысить производительность.
8. Важно, что Hibernate может использовать чистый SQL, а значит поддерживает возможность оптимизации запросов и работы с любым сторонним вендором БД и его фичами.

Файл конфигурации Hibernate содержит в себе данные о базе данных и необходим для инициализации SessionFactory. В .xml файле необходимо указать вендора базы данных или JNDI ресурсы, а так же информацию об используемом диалекте, что поможет hibernate выбрать режим работы с конкретной базой данных.

Существует четыре способа конфигурации работы с Hibernate :

* используя аннотации;
* hibernate.cfg.xml;
* hibernate.properties;
* persistence.xml.

Самый частый способ конфигурации: через аннотации и файл persistence.xml, что касается файлов hibernate.properties и hibernate.cfg.xml, то hibernate.cfg.xml главнее (если в приложение есть оба файла, то принимаются настройки из файла hibernate.cfg.xml). Конфигурация аннотациями, хоть и удобна, но не всегда возможна, к примеру, если для разных баз данных или для разных ситуаций хотите иметь разные конфигурацию сущностей, то следует использовать xml файлы конфигураций.