

## Практическое задание №5.

### Подготовительная работа.

Для выполнения данного задания должна быть проведена подготовительная работа из задания №4 и корректно настроено подключение к БД.

### Задание.

В рамках данной практической работы необходимо реализовать модель данных на стороне Java для мапинга с таблицами БД. Также должны быть реализованы базовые DAO классы и расширенные методы для запросов с фильтрами, с использованием Criteria API.

1. Создать в Java в пакете dao.model классы описывающие все используемые таблицы. Для этого потребуется создать классы с необходимыми полями, затем добавить к ним описывающие аннотации и добавить в файл конфигурации hibernate строки привязки классов. Подробнее о шагах создании модели можно посмотреть по ссылке в разделе Annotation Mapping (<https://easyjava.ru/data/hibernate/mapping-sushhnostej-v-hibernate/>)  
Подробнее об общих аннотациях (<https://easyjava.ru/data/jpa/jpa-entity-mapping/>)  
Подробнее о свойствах-связях (<https://easyjava.ru/data/jpa/jpa-i-svyazi-mezhdu-obektami/>)
2. Создать для класса базовой сущности (если такая имеется) и для всех классов справочников DAO классы. Для каждого DAO класса создать один метод получения всех возможных элементов с использованием Criteria API (<https://easyjava.ru/data/hibernate/hibernate-criteria-api/>)

#### SELECT

3. Перепишите методы реализованные в предыдущей работе на Criteria API.
4. В одном из DAO классов реализуйте хотя бы один метод получения информации, в котором набор используемых фильтров может меняться в зависимости от входных параметров. Т.е. секции add для Criteria могут динамически добавляться или убираться из запроса в зависимости от каких-либо условий. Например если в метод пришел аргумент Name не отличный от null – добавить проверку по имени.
5. Реализуйте с использованием Criteria API запрос, используемый при тестировании индексов в работе №3

#### INSERT, UPDATE, DELETE

6. Реализуйте в DAO классах методы, позволяющие получать, создавать, сохранять и удалять конкретную сущность в рамках сессии. Например:  

```
Humanface human = session.get(Humanface.class, 1)
```

  
Подробнее про управление сущностями в Hibernate вы можете почитать по ссылке: <https://easyjava.ru/data/hibernate/upravlenie-sushhnostyami-v-hibernate/>
7. Для демонстрации работы приложения на стороне Java:
  - Загрузите сущность (1) у которой имеется связь один ко многим
  - Создайте несколько объектов (n) класса, который могут быть к ней привязаны
  - Привяжите n к 1 и сохраните результат
  - Используя Criteria API выведите из БД список n привязанных к сущности 1