

Задание

1. Добавьте к новому maven проекту поддержку JPA: реализация Hibernate (можно EclipseLink). Настройте зависимости.
2. Разработайте уровень данных без web-интерфейса (пока)
3. У проекта будет следующий функционал:
 - a. В системе будет администратор и клиенты.
 - b. Администратор может зарегистрировать в системе нового клиента (тогда у клиента определяются характеристики, в том числе «Адрес» на основе композиции).
 - c. Клиент может быть привязан к новому или существующему «Счету» (например, счету другого «Клиента»).
 - d. Клиент может запросить издание банковской «Карты»
 - e. Администратор может заводить два типа карт «Дебетовые» (нельзя уходить в овердрафт) и «Кредитные» (возможно расходование денежных средств сверх внесенной клиентом суммы). Карты должны наследоваться от суперкласса Карта.
 - f. На одного «Клиента» можно регистрировать несколько «карт», привязанных к одному или разным «Счетам». Связи между «Клиентом» и «Картой» многие-ко-многим через «Счет».
 - g. «Счет» может быть закрыт «Администратором» по требованию «Клиента».
 - h. Клиент может выполнять действия со своим «Картами»: баланс, перевод с карты на карту, со счета на карту(ы), снять, положить сумму. «Карта» может быть заблокирована по требованию «Клиента».
 - i. Администратор просматривает статистику по Клиентам и счетам.
4. Все упомянутые бины системы должны быть сущностями (@Entity). Настройте их отображение в соответствии с известными вам стратегиями в таблицы бд MySQL (можно др.).
5. Одной или нескольким сущностям задайте генератор идентификаторов.
6. Настройте отображение 2 типов: одна таблица на иерархию классов и одна таблица на каждый класс. Реализуйте логику по внесению и извлечению различных сущностей из этой иерархии.
7. Создайте классы Manager-ы (DAO), которые позаботятся о постоянстве сущностей и управлении – удаление, добавление и извлечение сущностей. Добавьте функционал агрегационных функций в ваш DAO: максимальные и средние значения параметров, суммы и общее количество элементов.
 - a. При удалении настройте каскадирование (атрибут cascade).
8. На основе HQL разработайте запросы: вывод всех карт пользователя, вывод всех пользователей отдельного счета в отсортированном виде, и т.д. (например, все овердрафты на картах).

- a. Реализуйте механизм пагинации для результата ваших запросов. На странице должно быть не больше n объектов.
- b. Создайте несколько запросов Criteria с Restrictions (like, between, isNotNull и т.п.)

Вопросы:

1. Что такое JPA?
2. Поясните термин ORM (объекто-реляционное отображение).
3. В чем её отличие JPA от Hibernate?
4. Для чего используется Session и SessionFactory?
5. В чем отличие Entity и класса? Как создать Entity?
6. Перечислите требования к Entity в Hibernate?
7. Поясните аннотации: @Table, @Id, @GeneratedValue (какие есть стратегии), @Column, @JoinTable.
8. Как организовать связь между сущностями @OneToOne однонаправленную по умолчанию и с настройкой отображения?
9. Приведите пример организации отображения @OneToMany двунаправленной.
10. Приведите пример организации отображения @ManyToMany двунаправленной.
11. Поясните как происходит отображение наследования по стратегии одна таблица на класс.
12. Поясните как происходит отображение наследования по стратегии соединенный подкласс
13. Поясните как происходит отображение наследования по стратегии таблица на конкретный класс
14. Что такое HQL?
15. Что содержит файл hibernate.cfg.xml?