Inheritance Mapping

Exercise IMP.1 – Mapping Inheritance

The Setup:

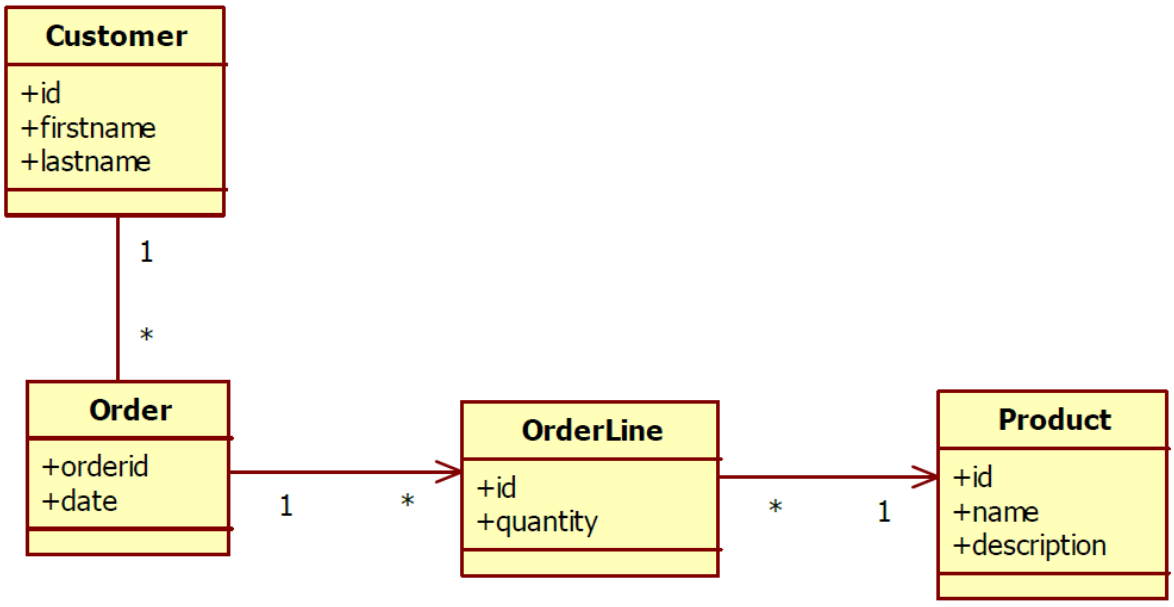
The objective of this exercise is to practice mapping inheritance. The choice of how you wish to implement your inheritance mapping, and which collections you wish to use for your association mappings is left up to you.

Start this exercise by creating a new project called **Lab5-Inheritance** Remember to also add the Hibernate, mysql, and log4j dependencies to the project's pom.xml.

This exercise consists out of 2 parts. In the first part you will create a basic application that does not use inheritance. Once you have tested to see that the application works, the second part asks you to add inheritance, and test it again.

The Exercise:

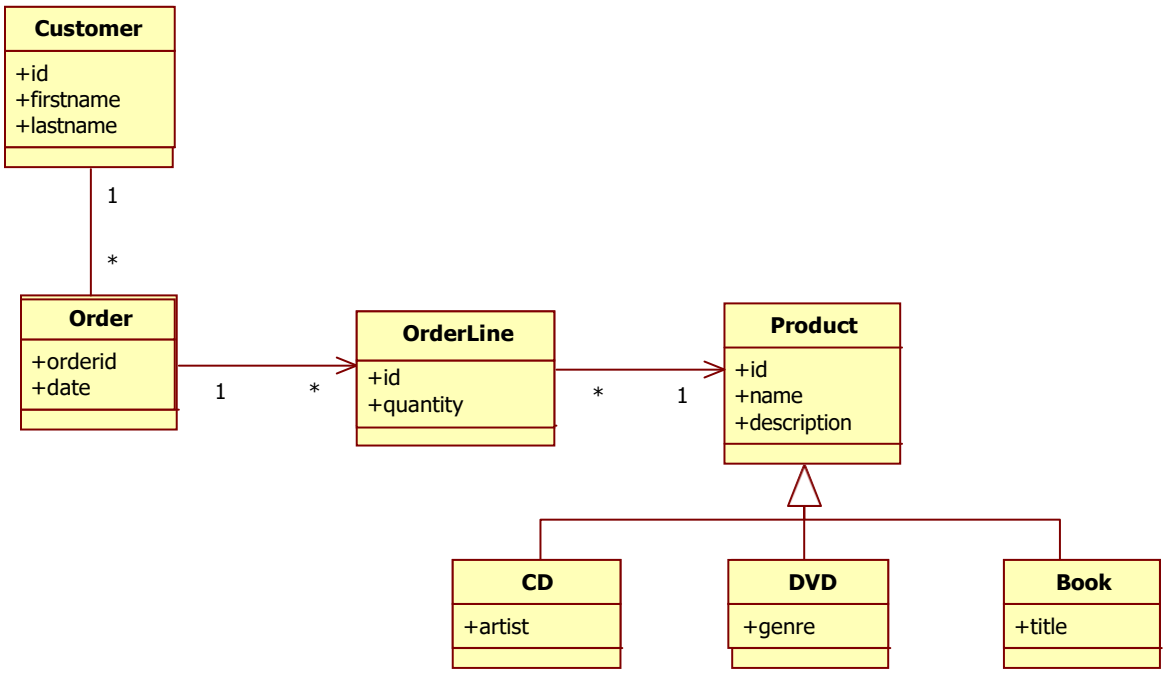
a) Create a small hibernate application that uses the following entities:



Please note: Order is a SQL reserved keyword; you can create an `Order` table by escaping it with back-tick. Alternately you can use the @Table annotation to provide a different table name.

Test the application by creating an object tree of the given domain classes, saving this tree to the database, and then retrieving the objects from the database to check if everything was persisted correctly.

b) Change the application to use the following updated domain:



Once again test the application by creating an object tree (now containing CDs, DVDs, and Books), saving it to the database, and then retrieving it to check if everything was persisted correctly.

Try it first with the single table strategy, run the code to see what the table looks like. Then update to the joined table strategy and check the database again.

Complex Mapping (CXM)

Exercise CXM.1 – Complex Mapping Exercise

The Setup:

The objective of this exercise is to practice mapping classes onto a table structure that does not closely match the class structure. In other words: to practice more complex mapping.

Start this exercise by creating a new project called **Lab5-Complex**. Remember to add the Hibernate, mysql, and log4j dependencies to the project.

The Exercise:

Create an application that maps the following classes onto the tables listed next to them:

```
public class Appointment {
    private Long id;
    private String appdate;
    private Patient patient;
    private Payment payment;
    private Doctor doctor;
```

```
public class Patient {
    private Long id;
    private String name;
    private String street;
    private String zip;
    private String city;
```

```
public class Payment {
    private String paydate;
    private double amount;
```

```
public class Doctor {
    private Long id;
    private String doctortype;
    private String firstname;
    private String lastname;
```

Appointment table:

ID	APPDATE	PAYDATE	AMOUNT	PATIENT	DOCTOR
1	15/05/08	12/06/08	100	1	1

Patient table:

ID	NAME
1	John Doe

Address table

PATIENT_ID	STREET	CITY	ZIP
1	100 Main Street	Boston	23114

Doctor table

ID	TYPE	FIRSTNAME	LASTNAME
1	Eye doctor	Frank	Brown

Once you have mapped the classes, test your mappings by creating an application which stores an object structure, and then retrieves it again to check everything was persisted correctly