Extra Lab 4: Hibernate Entity Inheritance

In this lab, you will define Hibernate entity classes that use inheritance.

Objectives:

- To use inheritance in a persistent object model in Hibernate

Part 1: Creating an Object Model with Inheritance

In this part, you will create classes that related by inheritance and composition.

The object model is for a fictitious rental-car company, Blossom Rentals, that imports cars from various locales and must pay a import duty on each imported vehicle.

Steps:

_1. In the JBoss Developer Studio (this lab assumes the Studio instead of Eclipse), create a new Java project named **ExtraLab4** according to these specifications:

```
Name:           ExtraLab4
Libraries:      Hibernate User Library
```

_2. Copy the following files from *ExtraLab1/src* to the *src* folder in the new project:

```
hibernate.cfg.xml
log4j.properties
```

_3. Edit the copied hibernate.cfg.xml and make the following changes:

a. Change the *mapping resource* element to refer to:

```
com/blossomrental/cars/Auto.hbm.xml
```

You will create this file in a later step.

b. Define additional *mapping-resource* elements for **com/blossomrental/cars/EuropeanAuto.hbm.xml"** and **com/blossomrental/cars/JapaneseAuto.hbm.xml"**.

c. Ensure that the following property is present and not commented-out:

```
<property name="hibernate.hbm2ddl.auto">create</property>
```

Save and close the configuration file.

_4. Create a new class named **Auto** in the **com.blossomrental.cars** package. This new class doesn't need a *main* method.

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

_5. Complete the Auto entity:

a. Define fields:

```
private Long vehicleID;
private double price;
private double dutyRate;
private String model;
```

b. Use the *Source - Generate Getters and Setters* menu to generate get/set methods for all of the fields.

c. Write a method in the new class that calculates the import duty. For generic Autos, the duty is zero:

```
public double payDuty()
{
    return 0;
}
```

_6. Create a new class named **EuropeanAuto** in the *com.blossomrental.cars* package that extends **Auto**. The class doesn't need a *main* method.

Complete the class according to the following specifications:

a. Add this field and generate get/methods for it:

```
private double euMembershipDiscount;
```

b. Write a *payDuty* method that calculates the import duty based on the vehicle's price and duty rate, taking into account the *euMembershipDiscount*:

```
@Override
public double payDuty()
{
    double dutyRate = getDutyRate();
    double price = getPrice();

    double duty = dutyRate * price -
        price * euMembershipDiscount;
    return duty;
}
```

Note that this method overrides behavior from its superclass, Auto.

_7. Create a new class named **JapaneseAuto** in the *com.blossomrental.cars* package. The class doesn't need a *main* method.

Complete the class according to the following specifications:

a. Add a field and generate get/set methods:

```
private String portOfDeparture;
```


Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

b. Write a *payDuty* method that calculates import duty based on the vehicle's price, duty rate and port of departure:

```
@Override
public double payDuty()
{
    double dutyRate = getDutyRate();
    double price = getPrice();

    boolean fromYoko =
        portOfDeparture.equals("Yokohama");

    double duty = dutyRate * price;
    if (fromYoko)
        duty -= price * 0.1;

    return duty;
}
```

_8. Next, you must configure the entities for Hibernate. You will use the *Table Per Subclass (JOINED)* approach to map the inheritance. Follow these steps:

a. Right-click on the *com.blossomrental.cars* package and choose *New - Other - Hibernate - Hibernate XML Mapping File (hbm.xml)*, then press Next followed by Finish.

The Studio generates mapping files for each of the entities.

b. Open Auto.hbm.xml into the editor and change the *generator* to **identity**.

_9. Open EuropeanAuto.hbm.xml into the editor.

Note that the Studio generated this mapping in a separate XML file and then specified the *extends* attribute in the class element. That's equivalent to putting *joined-subclass* in the superclass's mapping file.

You don't need to modify this file.

_10. Open JapaneseAuto.hbm.xml and ensure you understand it.

_11. Create a new class named **CreateAutos** in the **com.blossomrental.test** package with a *main* method.

Complete the CreateAutos.java class:

a. Open a Hibernate session and start a transaction.

b. Create an Auto entity, initialize its properties and save it via the Hibernate session:

```
Auto a = new Auto();
a.setDutyRate(0.0);
a.setModel("Ford");
a.setPrice(20000);
session.save(a);
```

c. In the same fashion, create, initialize and save a EuropeanAuto to these specifications:

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

```
Duty rate:   20%
EU discount:  5%
Model:       BMW
Price:       35000
```

d. In the same fashion, create, initialize and save a JapaneseAuto to these specifications:

```
Duty rate:   15%
Model:       Honda
Price:       32000
Port:        Yokohama
```

e. Print out each of the vehicle's ID to the console.

f. Commit the transaction and clean up. (you

_12. Run CreateAutos.java as a Java Application. You should see the vehicle IDs.

_13. To verify that Hibernate inserted into the database, follow this procedure, use the Derby ij command processor to execute the following queries:

a.

```
SELECT * FROM Auto;
```

You should see a row for each entity your program created. Note that it contains the superclass-defined properties.

b.

```
SELECT * FROM EuropeanAuto;
```

You should see an entry for the European auto. Note that it contains a foreign key into the Auto table to associate it with its superclass's data.

c.

```
SELECT * FROM JapaneseAuto;
```

You should see an entry for the Japanese auto. Note that it contains a foreign key into the Auto table to associate it with its superclass's data.

_14. Next, write a simple program that will query the entities:

a. Create a new class in the *com.blossomrental.test* package named **QueryAutos** with a *main* method.

b. In the *main* method, open a Hibernate session. You can copy and paste this code from CreateAutos.java.

c. Execute a Hibernate query to retrieve all Auto entities.

d. Write a loop that accumulates the total duty paid by all vehicles and displays it.

e. Clean up. You can copy and paste this code from CreateAutos.java.

_15. Before you run, open *hibernate.cfg.xml* into the editor and comment-out the **hibernate.hbm2ddl.auto** property. If you left it in, Hibernate would create a new empty table - there would be no entities to query!

_16. Run QueryAutos.java as a Java Application. If you used the values suggested above, the total duty should be 6850.00.

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.