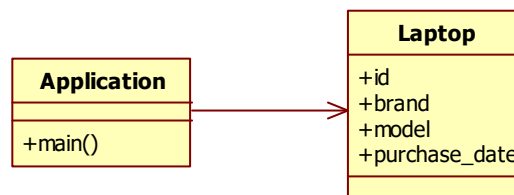## Exercise 02.2 – Basic Hibernate XML

The Setup:

The goal of this exercise is to practice using XML mappings for an object instead of annotations. Start this exercise by opening the **exercise02.2** project.

The Application:

The application is similar to the car application in exercise 02.1. The main difference is that instead of using annotations for persistence mapping, we have used XML.

```
                    ┌──────────────────┐
                    │      Laptop      │
┌──────────────────┐├──────────────────┤
│   Application    ││ +id              │
├──────────────────┤│ +brand          │
│                  │──────────────────►│ +model          │
│ +main()          ││ +purchase_date   │
└──────────────────┘├──────────────────┤
                    └──────────────────┘
```

The Hibernate mapping instructions for the Laptop class are in the **laptop.hbm.xml** file. Running Application.java should produce the following output:

```
Hibernate: /* insert hibernate.Laptop */ insert into laptop (id, brand,
model, purchase_date) values (null, ?, ?, ?)
Hibernate: call identity()
Hibernate: /* insert hibernate.Laptop */ insert into laptop (id, brand,
model, purchase_date) values (null, ?, ?, ?)
Hibernate: call identity()
Hibernate: /* from Laptop */ select laptop0_.id as id0_, laptop0_.brand as
brand0_, laptop0_.model as model0_, laptop0_.purchase_date as purchase4_0_
from laptop laptop0_
Apple iBook 2006-04-17
IBM t60 2007-03-30
```

The Exercise:

Study the code, and once you are comfortable with how the Laptop Object / Relational mapping is performed in XML, create a **Person** class and a **AppPerson** class.

The Person class should have the following attributes:

```java
public class Person {
    private long id;
    private String firstname;
    private String lastname;
    private Date dateofbirth;
```

The **AppPerson** class should perform the same operations as the AppBook in the previous exercise:

- Create / save 3 person objects
- Retrieve all person objects from the database and print them to the console
- Update a person, and delete another
- Retrieve all person objects from the database and print them to the console