Module 04: Collection Mapping

Exercise 04.1 – Mapping Collections

The Setup:

The main objective of this exercise is to practice mapping the different types of collections.

Start this exercise by creating a new project called **exercise04.1**. Remember to add the dependencies to the pom.xml.

The exercise consists of 3 parts. Just like the previous exercise, you will have to create the entity classes used for each part. Remember to use the code-generating features of your IDE.

The Database:

Because this exercise has a (possibly different) Employee and Student classes than the ones already in the database, you are likely to get the following error:

```
org.hibernate.exception.SQLGrammarException: could not insert:
[collections.map.Student]
       ...
       ...
Caused by: java.sql.SQLException: Column not found: ID in statement [/*
insert ordering.Book */ insert into Book (title, id) values (?, ?)]
       ...
```

This error is caused by the difference between the Employee and Student classes from this exercise and the previous exercises. You can fix this error by first dropping the old tables from the database, and then running the application again.

You can drop the Student table by executing the following SQL in MySQL. Running the application again afterwards will make Hibernate generate the correct tables:

```
SET foreign_key_checks = 0;

DROP TABLE student;

DROP TABLE employee;

SET foreign_key_checks = 1;
```

Alternately you can also drop and re-create the entire database:

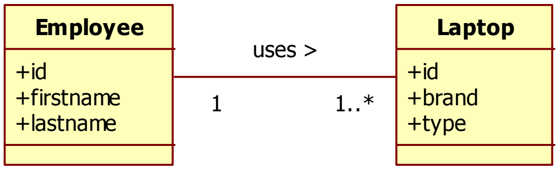
```
DROP DATABASE cs544;

CREATE DATABASE cs544;
```

The Exercise:

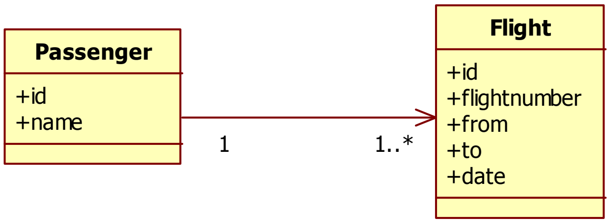
Be sure to also update App.java with some code to test each of the following mappings.

a) Create a **Bidirectional OneToMany/ManyToOne** association between **Student** and **Laptop** using a **Set**.

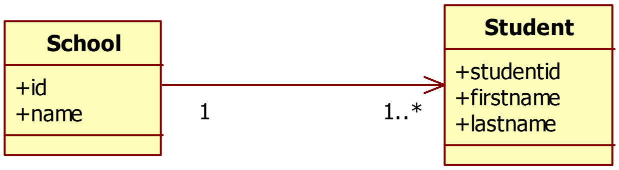


b) Create a **Unidirectional OneToMany** between **Passenger** and **Flight** using a **List**.

Please note that 'TO' and 'FROM' are SQL keywords and should either be escaped with backticks like `to` or changed to a different column name using the @Column annotation



c) Create a **Unidirectional OneToMany** association between **School** and **Student** using a **Map**, where **studentId** is used as the key for the map.


School
+id
+name
1
1..*
Student
+studentid
+firstname
+lastname