Exercises:

Introduction

We continue with the hibernateExcerises project. So far we have learned to create, update, delete and retrieve data from our derby database by using Hibernate.
We have learned to have different class name and variable names from the table.

Overview

In this exercise:

- We will practice having two classes and make a many to one relationship between these two classes and see how it will affect our database.
- We create a table Author and we map this class to the Book class and make a many to one relationship. It means that a book can be written by only one author but an author can write one or more books.

Tasks

Create Author class

- Create a class named *Author* with these attributes and constructor:

  private String name
  private String address
  private int age
  A constructor with name as the argument.
  Accessors (get methods)
  toString()

- Now do the hibernate configurations for this class. And the necessary modifications in Author class.
- Do the necessary changes in the Book class and xml file.
- Create a method in the Book class to allocate an Author to the book.
- Add a method in the Book class to get the name of the author back.

In HarnessTest class

- Create a new Book (using the constructor with one argument:name) and a new Author in the main method.
- Allocate the author to the book.

- Persist the data to the database.
- Drop the book table and run and see the results.

- Now find the book by its id and print it out.
- Find the name of the author for this book.

- Create an object from the author that you have saved in the database (by getting help from the get method on session and the id of the author).
- Create a new book, allocate the author that you have got from the database to it and save it into the database.
- Now run the code and then check the book table and see if it has gotten the id of the author.