**Java**

Object Oriented Development

Comparisons Exercises

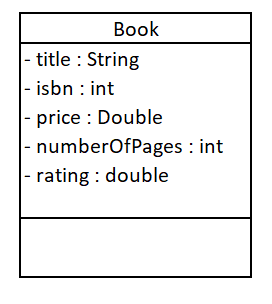
© FDM Group Ltd 2020. All Rights Reserved.

Any unauthorised reproduction or distribution in part  
or in whole will constitute an infringement of copyright.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 25 / 11 / 20 | Nick Lawton | First draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Setting up

1. Create a package called com.fdmgroup.comparisonsExercises.
2. Create the Book class shown in the UML below:



1. Create a class called Runner with a main method. In the main method create the following Book objects:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | title | isbn | price | number of pages | rating |
| book1 | Learn Java | 45678 | 19.99 | 500 | 4.3 |
| book2 | Learn UNIX | 65432 | 17.5 | 650 | 3.9 |
| book3 | Learn SQL | 87956 | 9.75 | 220 | 4.4 |
| book4 | Learn Agile | 35367 | 25.5 | 500 | 4.4 |
| book5 | Learn React | 55676 | 28.75 | 490 | 4.8 |
| book6 | Learn JavaScript | 34343 | 14.99 | 300 | 4.4 |
| book7 | Learn Python | 65767 | 27.25 | 500 | 4.1 |

1. Create an ArrayList of type Book and add all the Book objects to it.

## Creating Comparators

1. Write a comparator class which sorts Books by their number of pages in ascending order.
2. Write a comparator class which sorts Books by their name in ascending order.
3. Write a comparator class which sorts Books by thir price in descending order.
4. Write a comparator which sorts Books on 2 attributes. It should sort first on number of pages in ascending order and then by isbn in descending order.
5. Write a comparator which sorts Books on 2 attributes. It should sort first on rating in descending order and then by price in ascending order.

## Using Comparators with Collections.sort()

Use the Collections.sort() method to test out each of your comparators in turn. After each sort, use a for each loop to print out the attributes of the books in the ArrayList. Verify that they have been sorted into the right order.

## Using Comparators with a TreeSet

Create a TreeSet of books using the comparator you created in part 1.2.5 (descending rating and then price). Add all of your Book objects to the TreeSet. Use a for each loop to iterate through the TreeSet printing out the ratings and prices of the books. Verify that they have been sorted in the right order.

## Using a built in Comparator

Write a method which takes in a variable argument of type double. It should return a PriorityQueue of type Double where the elements are in descending order. Do not create your own Comparator class for this. You should research how to use an existing built in Comparator to do this.