

Learning Objectives:

- review implementing the interface `Comparable<T>`
- programming against interfaces not objects.
- review reading in data from a file
- use the internet to learn about the interface `Comparator<T>`
- use a `Comparator` to provide an alternative way to order elements in a collection

Turn in:

Turn in the assignment via Canvas

Description:

Create a package called `books` that includes 3 files: `Book.java`, `BookApp.java`, and `books.csv` (provided).

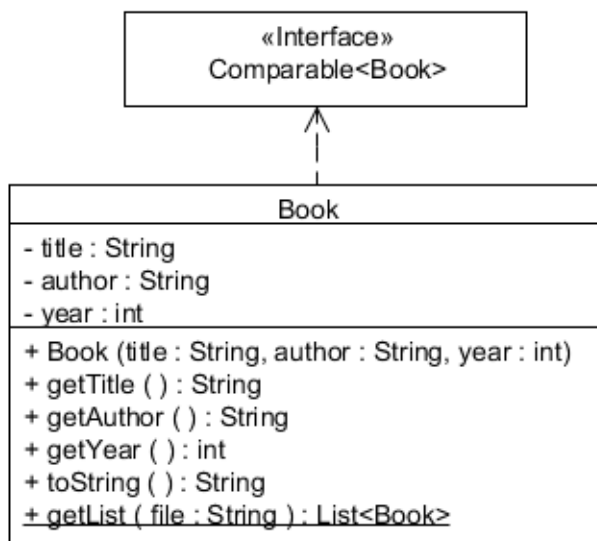
Class Book represents Pulitzer prize winning books that have a title, an author and a year, when they won the award. Implement the class `Book` exactly as specified in the UML diagram below.

You are allowed to create private methods to structure your code, but fields, constructors, and public methods must not be changed nor added or removed.

Notice that class `Book` implements the **interface `Comparable<T>`**. The method specified in the interface is not listed in the UML class diagram of class `Book`. However, the fact that `Book` implements `Comparable<T>` implies that the interface method needs to be implemented.

The interface `Comparable<T>` implements the natural order. In case of class `Book` it should **sort by title**.

Also notice that class `Book` is immutable. It has getters but no setters.



Method toString:

The `toString` method should return a string of the following form: *name by author (year)*

Make sure to include the `@Override` annotation

Method getList:

Note that the method `getList` is underlined. Underlining a method in a UML class diagrams indicates that the method is static.

The method `getList` should read in the data from the csv file `book.csv`. If a line doesn't follow the pattern *title,author,year* then a message should be written to the standard error stream (see sample output) The program should continue reading in the next line. NO exception should be thrown .

Please note that the sample output is only provided to help clarify the instructions. The program still needs to fulfill all the requirement when I test it with another csv file (e.g. where all lines are correct or other lines have an issue)

Write a test client called BookApp.java

- It should read in the data from the file `book.csv`. Two lines have an issue. Treat them as described above.
- Print the number of books that were read in.
Make sure to determine the number of books at run time. I will test your code with a different csv file.
- Sort the list in natural order and print the list
- **Sort the books in the list in reverse order using a `Comparator<T>` that is provided in class `Collections`**
- List the book in the newly reversed order

Sample Output:

Problem reading in "No Pulitzer prize for fiction was awarded in 2012"

Problem reading in "The Brief, Wondrous Life of Oscar Wao, Junot Diaz, 2008"

Number of books read in: 14

Sorted book list:

A Visit from the Goon Squad by Jennifer Egan (2011)
Empire Falls by Richard Russo (2002)
Gilead by Marilynne Robinson (2005)
Interpreter of Maladies by Jhumpa Lahiri (2000)
March by Geraldine Brooks (2006)
Middlesex by Jeffrey Eugenides (2003)
Olive Kitteridge by Elizabeth Strout (2009)
The Amazing Adventures of Kavalier & Clay by Michael Chabon (2001)
The Goldfinch by Donna Tartt (2014)
The Hours by Michael Cunningham (1999)
The Known World by Edward P. Jones (2004)
The Orphan Master's Son by Adam Johnson (2013)
The Road by Cormac McCarthy (2007)
Tinkers by Paul Harding (2010)

Reverse order:

Tinkers by Paul Harding (2010)
The Road by Cormac McCarthy (2007)
The Orphan Master's Son by Adam Johnson (2013)
The Known World by Edward P. Jones (2004)
The Hours by Michael Cunningham (1999)
The Goldfinch by Donna Tartt (2014)
The Amazing Adventures of Kavalier & Clay by Michael Chabon (2001)
Olive Kitteridge by Elizabeth Strout (2009)
Middlesex by Jeffrey Eugenides (2003)
March by Geraldine Brooks (2006)
Interpreter of Maladies by Jhumpa Lahiri (2000)
Gilead by Marilynne Robinson (2005)
Empire Falls by Richard Russo (2002)
A Visit from the Goon Squad by Jennifer Egan (2011)