Lab: Iterators and Comparators

This document defines the lab for the "Java Advanced" course @ Software University. Please submit your solutions (source code) of all below-described problems in Judge.

1. Book

Create a class **Book** from the **UML diagram** below:

Book
- title: String - year: int - authors: List<String>
- setTitle(String) - setYear(String) - setAuthors(String…) + getTitle(): String + getYear(): int + getAuthors(): List<String>

You can use only **one constructor**. There **can be no authors**, **one author,** or **many authors**.

Examples

Main.java

```
public static void main(String[] args) {
    Book bookOne = new Book("Animal Farm", 2003, "George Orwell");
    Book bookThree = new Book("The Documents in the Case", 2002);
    Book bookTwo = new Book("The Documents in the Case",1930,"Dorothy Sayers","Robert Eustace");

    List<Book> books = new ArrayList<>();
    books.add(bookOne);
    books.add(bookTwo);
    books.add(bookThree);
}
```

Solution

```
public Book(String title, int year, String... authors) {
    this.setTitle(title);
    this.setYear(year);
    this.setAuthors(authors);
}

private void setAuthors(String... authors) {
    if (authors.length == 0) {
        this.authors = new ArrayList<String>();
    } else {
        this.authors = new ArrayList<>(Arrays.asList(authors));
    }
}
```

2. Library

Create a class **Library** from the **UML diagram** below:

<<Iterable<Book>>> Library
- books: Book[]
+ iterator(): Iterator<Book>

Create a **nested class LibIterator** from the **UML diagram** below:

<<Iterator<Book>>> LibIterator
- counter: int
+ hasNext(): boolean
+ next(): Book

Examples

Main.java

```
public static void main(String[] args) {
Book bookOne = new Book("Animal Farm", 2003, "George Orwell");
   Book bookThree = new Book("The Documents in the Case", 2002);
   Book bookTwo = new Book("The Documents in the Case",1930,"Dorothy Sayers","Robert Eustace");

   Library library = new Library<>(bookOne, bookTwo, bookThree);

   for (Book book: library) {
         System.out.println(book.getTitle());
   }
}
```

Solution

```
public class Library<Book> implements Iterable<Book> {
    private Book[] books;

    public Library(Book... books) { this.books = books; }

    @Override
    public Iterator<Book> iterator() { return new LibraryIterator(); }

    private final class LibraryIterator implements Iterator<Book> {
        private int counter = 0;

        @Override
        public boolean hasNext() {...}

        @Override
        public Book next() {...}
    }
}
```

3. Comparable Book

Expand Book by implementing **Comparable<Book>.**

The book has to be **compared by title**. When the title is equal, **compare** them by **year.**

Expand **Book** from **UML diagram** below:

<<Comparable<Book>>> **Book**
- title: String - year: int - authors: List<String>
- setTitle(String) - setYear(String) - setAuthors(String…) + getTitle(): String + getYear(): int + getAuthors(): List<String> + **compareTo(Book): int**

You can use only **one constructor**. There **can be no authors**, **one author,** or **many authors**.

Examples

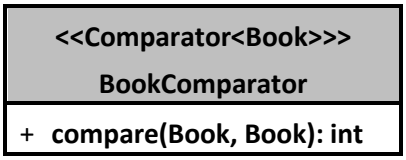
Main.java

```
public static void main(String[] args) {
   Book bookOne = new Book("Animal Farm", 2003, "George Orwell");
   Book bookThree = new Book("The Documents in the Case", 2002);
   Book bookTwo = new Book("The Documents in the Case", 1930, "Dorothy Sayers", "Robert Eustace");

   if (bookOne.compareTo(bookTwo) > 0) {
       System.out.println(String.format("%s is before %s", bookOne, bookTwo));
   } else if (bookOne.compareTo(bookTwo) < 0) {
       System.out.println(String.format("%s is before %s", bookTwo, bookOne));
   } else {
       System.out.println("Book are equal");
   }
}
```

4. Book Comparator

Create a class **BookComparator** from the **UML diagram** below:



BookComparator has to **compare** two books by:

1. Book title.
2. Year of publishing a book.

Examples

Main.java

```
public static void main(String[] args) {
   Book bookOne = new Book("Animal Farm", 2003, "George Orwell");
   Book bookThree = new Book("The Documents in the Case", 2002);
   Book bookTwo = new Book("The Documents in the Case", 1930, "Dorothy Sayers", "Robert
   Eustace");

   List<Book> books = new ArrayList<>();
   books.add(bookOne);
   books.add(bookTwo);
   books.add(bookThree);

   books.sort(new BookComparator());

   for (Book : books) {
         System.out.println(book.getTitle() + book.getYear());
   }
}
```