Name of the Course : Complete Java SE8 Developer Bootcamp

Level : Difficult

Tool Stack : Java8 and Junit4

Problem Statement : Provide a code solution to display list of books in ascending order of either book number or book title or book author from a library, using java.util.List, static block , Lambda expression, Comparator interface and String formatting output etc features.

Description : The ***The People’s Library*** has a very good collection of books. Library needs to display list of books in ascending order of 1.Book number-wise 2.Book title-wise 3. Book author-wise. You need to write a menu based program in Java that will display 3 options and choose any option to display list of books accordingly.

You need to

1. create class name **Book** which has the following private attributes:

**Integer bookNumber,**

**String bookTitle,**

**String author,**

Create getter and setter methods, one parameterized and non parameterized constructors.

Override toString() method in the following format

**String.format("%-10s %-35s %-20s ",....)**

2.create class **Library** which has the following members:

a) private static List<Book> bookList;

b) static block to store at least 6 books object into the list.

c) public static List<Book> getAllBooks() : to return the bookList;

3.Create class **BookService** with the followings methods

a) public List<Book> arrangeBooksNumberWise(List<Book> bookList) to arrange books in ascending order of book number.

b) public List<Book> arrangeBooksTitleWise(List<Book> bookList) to arrange books in ascending order of book title.

c) public List<Book> arrangeBooksAuthorWise(List<Book> bookList) to arrange books in ascending order of author’s name.

**(Use *lambda expression* for Comparator interface*)***

4. Create class **Main** withthe method public static void main(String [] args). It will display a menu in the following format:

Menu

1.Display Book Number-wise

2.Display Book Title-wise

3.Display Book Author-wise

4.Exit

Enter choice(1-4):

finally displays the books based on sorting choice in the mentioned formatted manner with headings (see below).

Code:

**public** **class** Book {

**private** Integer bookNumber;

**private** String bookTitle;

**private** String author;

**public** Book() {

**super**();

}

**public** Book(Integer bookNumber, String bookTitle, String author) {

**super**();

**this**.bookNumber = bookNumber;

**this**.bookTitle = bookTitle;

**this**.author = author;

}

**public** Integer getBookNumber() {

**return** bookNumber;

}

**public** **void** setBookNumber(Integer bookNumber) {

**this**.bookNumber = bookNumber;

}

**public** String getBookTitle() {

**return** bookTitle;

}

**public** **void** setBookTitle(String bookTitle) {

**this**.bookTitle = bookTitle;

}

**public** String getAuthor() {

**return** author;

}

**public** **void** setAuthor(String author) {

**this**.author = author;

}

@Override

**public** String toString() {

String output=String.*format*("%-10s %-35s %-20s",bookNumber,bookTitle,author);

**return** output;

}

}

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** Library {

**private** **static** List<Book> *bookList*=**new** ArrayList<>();

**static**

{

*bookList*.add(**new** Book(10006,"War And Peace","Tolstoy"));

*bookList*.add(**new** Book(10002,"Song Offerings","Tagore"));

*bookList*.add(**new** Book(10004,"Macbeth","Shakespeare"));

*bookList*.add(**new** Book(10003,"Tempest","Shakespeare"));

*bookList*.add(**new** Book(10005,"Gora","Tagore"));

*bookList*.add(**new** Book(10001,"Anna Karenina","Tolstoy"));

*bookList*.add(**new** Book(10007,"Hamlet","Shakespeare"));

*bookList*.add(**new** Book(10010,"Hound of Baskerville","Doyle"));

*bookList*.add(**new** Book(10009,"Red Oleanders","Tagore"));

*bookList*.add(**new** Book(10008,"Adventure of Holme","Doyle"));

}

**public** **static** List<Book> getAllBooks()

{

**return** *bookList*;

}

}

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.List;

**public** **class** BookService {

**public** List<Book> arrangeBooksNumberWise(List<Book> bookList)

{

Comparator<Book> numberCompare=(Book a,Book b)->{

Integer number1=a.getBookNumber();

Integer number2=b.getBookNumber();

**return** number1.compareTo(number2);

};

Collections.*sort*(bookList,numberCompare);

**return** bookList;

}

**public** List<Book> arrangeBooksTitleWise(List<Book> bookList)

{

Comparator<Book> titleCompare=(Book a,Book b)->{

String title1=a.getBookTitle();

String title2=b.getBookTitle();

**return** title1.compareTo(title2);

};

Collections.*sort*(bookList,titleCompare);

**return** bookList;

}

**public** List<Book> arrangeBooksAuthorWise(List<Book> bookList)

{

Comparator<Book> authorCompare=(Book a,Book b)->{

String author1=a.getAuthor();

String author2=b.getAuthor();

**return** author1.compareTo(author2);

};

Collections.*sort*(bookList,authorCompare);

**return** bookList;

}

}

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Scanner scanner=**new** Scanner(System.***in***);

List<Book> bookList=Library.*getAllBooks*();

**while**(**true**)

{

System.***out***.println(" Menu ");

System.***out***.println("1.Display Book Number-wise");

System.***out***.println("2.Display Book Title-wise");

System.***out***.println("3.Display Book Author-wise");

System.***out***.println("4.Exit");

System.***out***.println("Enter choice(1-4): ");

String choice=scanner.nextLine();

**switch**(choice)

{

**case** "1":bookList=**new** BookService().arrangeBooksNumberWise(bookList);**break**;

**case** "2":bookList=**new** BookService().arrangeBooksTitleWise(bookList);**break**;

**case** "3":bookList=**new** BookService().arrangeBooksAuthorWise(bookList);**break**;

**case** "4":System.*exit*(0);

}// end of switch

String output=String.*format*("%-20s %-35s %-20s ","Book Number","Book Title","Author");

System.***out***.println(output);

**for**(Book book:bookList)

System.***out***.println(book);

}// end of loop

}

}

Junit Testing

**import** **static** org.junit.Assert.\*;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.junit.Test;

**public** **class** BookServiceTest {

@Test

**public** **void** testArrangeBooksNumberWise()

{

List<Book> bookList = Library.*getAllBooks*();

*assertEquals*(**new** Integer(10001),(Integer)**new** BookService().arrangeBooksNumberWise(bookList).get(0).getBookNumber());

}

@Test

**public** **void** testArrangeBooksTitleWise() {

List<Book> bookList = Library.*getAllBooks*();

*assertEquals*(**new** String("Adventure of Holme"),(String)**new** BookService().arrangeBooksTitleWise(bookList).get(0).getBookTitle());

}

@Test

**public** **void** testArrangeBooksAuthorWise() {

List<Book> bookList = Library.*getAllBooks*();

*assertEquals*(**new** String("Doyle"),(String)**new** BookService().arrangeBooksAuthorWise(bookList).get(0).getAuthor());

}

}

Test Data1

Menu

1.Display Book Number-wise

2.Display Book Title-wise

3.Display Book Author-wise

4.Exit

Enter choice(1-4):

1

Book Number Book Title Author

10001 Anna Karenina Tolstoy

10002 Song Offerings Tagore

10003 Tempest Shakespeare

10004 Macbeth Shakespeare

10005 Gora Tagore

10006 War And Peace Tolstoy

10007 Hamlet Shakespeare

10008 Adventure of Holme Doyle

10009 Red Oleanders Tagore

10010 Hound of Baskerville Doyle

Menu

1.Display Book Number-wise

2.Display Book Title-wise

3.Display Book Author-wise

4.Exit

Enter choice(1-4):

2

Book Number Book Title Author

10008 Adventure of Holme Doyle

10001 Anna Karenina Tolstoy

10005 Gora Tagore

10007 Hamlet Shakespeare

10010 Hound of Baskerville Doyle

10004 Macbeth Shakespeare

10009 Red Oleanders Tagore

10002 Song Offerings Tagore

10003 Tempest Shakespeare

10006 War And Peace Tolstoy

Menu

1.Display Book Number-wise

2.Display Book Title-wise

3.Display Book Author-wise

4.Exit

Enter choice(1-4):

3

Book Number Book Title Author

10008 Adventure of Holme Doyle

10010 Hound of Baskerville Doyle

10007 Hamlet Shakespeare

10004 Macbeth Shakespeare

10003 Tempest Shakespeare

10005 Gora Tagore

10009 Red Oleanders Tagore

10002 Song Offerings Tagore

10001 Anna Karenina Tolstoy

10006 War And Peace Tolstoy

Menu

1.Display Book Number-wise

2.Display Book Title-wise

3.Display Book Author-wise

4.Exit

Enter choice(1-4):

Learning outcome: Participant could able to learn Lambda expression in Comparator interface and String formatting output.