Specification for the program “BooKit”

Name

BooKit – Functions for managing a library

Description

BooKit reads commands from Command Line Interface (CLI) to manage all your book collection. The software allows you to add a book into the system, read the information of a book, update the information from a book and delete a book from the system.

Additionally, the software can find a book by author or title, and list all the registered books and authors in the storage.

How to Execute

bookit <input-file>

<input-file> is the file path that stores the books information. Once executed, a CLI will show up, then users can type functions there.

Input Data

All the input is going to be received by CLI, where each line represents action. The software will read the input and determine which action is going to be performed with the arguments.

Every input is composed of two parts, the specific function and their respective arguments:

<command> <arg1>;<arg2>;...;<argN>

The command name and the first argument are separated by one space. Each arguments are separated by semicolons (;). Arguments do not require quotations marks(“”), even if they contain spaces.

Example:

addBook Harry Potter;J.K. Rowling;8478884459;4;1999-03-25-123123

Function

- **addBook <title>;<author>;<rating>;<ISBN>;<publicationYear>**
  Adds a book to the file. The book must include the title of the book, the name of the author and the rating given to the book.
  If the book is added successfully, show “The book was added successfully” message.
  If the book already exists, the system should show the warning message “The book already exists”.

ISBN can be 10 or 13 digits, for example, 8478884459 or 8478884459100. If the user types an invalid ISBN, the system will show “Incorrect ISBN” message. *publicationYear* has to be a positive integer. If the user types an invalid publication year, the system will show “Incorrect publication year message. ***Note about rating****: The rating must be an integer between 1 and 5 (including both numbers), where 1 indicates the book is least liked and 5 indicates it is one of the most liked. Any incorrect format of rating during the add and update phrase should cause the system to show “Invalid Rating” message.*

- **updateRating <ISBN>;<rating>**
  Modifies the rating of a book by ISBN.
  If the update is successful, show “Update success” message and the rating of the book to which the ISBN belongs is changed to the new one provided.
  If the given ISBN doesn’t exist, the code should show the warning message “Book not found”.
  If the rating is invalid, show “Invalid Rating” message.
- **searchByTitle <title>**
  Search for books by title. The search should be case-insensitive and should support substring search. After search, the system shows a list of matching books which properties have the following format: Title;Author;ISBN;Rating;PublicationYear, each book’s properties are separated by a new line. If no book is found, show “Oops, no books are found.” message. ***Note about substring search****: Substring search is case-insensitive and matches the given text anywhere inside the title (not only the beginning). For example: searching “pot” returns books with titles like “Harry Potter” and searching “ring” matches “The Lord of the Rings”.*
- **searchByAuthor <author>**
  Searches for books by author. The search should be case-insensitive and should support substring search. After search, the system shows a list of matching books which properties have the following format: Title;Author;ISBN;Rating; PublicationYear, each book’s properties are separated by a new line. If no book is found, show “Oops, no books are found.” message.
- **searchByYear <year>**
  Searches for books by publication year using an exact match (i.e. searchByYear 2000 returns books published in the year 2000, entering 200 will not return a book from the year of 2000). After search, the system shows a list of matching books which properties have the following format: Title;Author;ISBN;Rating; PublicationYear, each book’s properties are separated by a new line. If no book is found, show “Oops, no books are found.” message.
- **listAll**
  ListAll has no arguments. Lists all the books, including their titles, authors, and ratings. Format: Title;Author;Rating

If the list has no books, show “The list is empty” message.

- **removeByISBN <ISBN>**
  Removes a book by the given ISBN, the ISBN must be an exact full-text match (for example, removeByISBN 8478884459 should remove the book with ISBN of 8478884459, but won’t remove a book with ISBN of 8478884459100).
  If the book is removed, the message “book successfully removed” will be shown on the screen and the book will have been deleted.
  If the book is not found, show “Book not found with <ISBN>” message.
  If the ISBN is invalid, the system will show “Incorrect ISBN” message.
- **removeByAuthor <author>**
  Removes all books by the given author, the author should be exact full-text match.
  If the author is found successfully, all the books with the author's name should also be removed and show “The author has been successfully removed” message.
  If the author is not found, show “Books not found with <author>” message.
- **bookOfTheDay**
  Returns a random book with a rating of 4 or 5.
  If it exists a book with that rating, the system will show the book in the format: Title;Author;ISBN;Rating;PublicationYear.
  If there are no books that meet the rating conditions, the system will display ‘No recommendation for today’. It is possible that the book may be repeated from other days.
- **orderBy <attribute>;<sorting>**
  Sort all stored books according to a specific attribute.
  <attribute> can be “title”, “author”, “rating”, “isbn”, “year”, and lexicographical sorting order will be used.
  <sorting> can be “incremental” or “decremental”, depending on how the books are sorted.
  If there are no books in storage, the system shows “No books to sort” message.

Limitations

- The functionality for reading files is only for test purposes and is therefore not especially fault tolerant. The input format should always be **<name>;<author>;<rating>;<ISBN>;<publicationYear>**, each book’s information is separated by a new line. Assuming all the book titles don’t have the “;” character. For example, one can assume that incorrectly spelled commands, commands with missing arguments, or commands with too many arguments will not be handled especially carefully.
- Publication years are assumed to be years after Christ and represented as positive integers only.