Explanation of the structure of your code:

At the beginning I decided to use the MCV structure, but since the program just uses the command line interface and the whole View was included in just one method, I divided the files in two parts, controller and model. In the controller I have a file to load the files to the memory, a file with 2 validators and the main. In the main it is included a method to display the options and the main loop of the program.

Short justification for your logic design decisions:

I decided to create classes for Books, Readers and Borrowings (the borrowings take 1 reader and 1 book) and they are added to an ArrayList of their respective class. The arrays are sorted using the BubbleSort method seen in class and as a search algorithm I decided to use the LinearSearch since in such a small program the poor efficiency of this method is not really an issue and was easier to code.
The main loop of the program displays a number of messages asking the user to choose an option using a number, once the number is typed, the operation is carried and then the user is prompted to choose another operation or end the program.

Explanation of the structure of the text files used to ensure data persistency:

As text files I decided to use a CSV format because I found it easier to use. The readers, books and borrowings are loaded as soon as the program starts and borrowings can be added or removed. When added, the new record is added to the array and then the whole array is rewritten to the file. When removed, the record is removed from the array and the whole array is also rewritten.