

Final Project Proposal Mini-Library and Reader

Our project stores an inventory of books in a three-dimensional ArrayList sorted alphabetically in one class, where the three dimensions are book genres (columns), subgenres (rows), and book titles. The user navigates the library by going from the most general to most specific criteria describing the book—once inside a genre, the user can view all of the subgenres within it, and upon selecting a subgenre, can view all the specific book titles. In a subgenre, they can choose a book by entering the name of the title, its index+1, or by prompting the computer to choose a random book from that subgenre. They can also search for a book title, and the program would see if the string they entered is within the string title of any of the books (both turned lowercase, disregarding punctuation). The random selection and search features are also available before selecting a genre or subgenre.

A separate class will contain methods to determine statistics. The user is asked to input their reading speed in words/minute, which will be used to calculate the estimated reading time. When the user is in a given subgenre they can choose to see either summary statistics or estimated length. If prompted, the program calculates the estimated time to read time the selected book based on the user's input of average words per minute. The information summary displays the following statistics:

- average word length
- average sentence length
- number of words
- number of sentences
- average words per sentence
- overall reading difficulty statistics (determined based on word length/sentence length)

For these two features (statistics and estimated reading time), the program will read the file and search for characters like spaces (for number of words and word

length), end punctuation (for sentences), etc. Once the user chooses to open the book, another class allows the user to read the book page by page, with their command line input allowing them to turn to the next page, go to a particular page, or exit the book. The program will print a fixed amount of words at a time (the range determined by number of words per page multiplied by the page number), and when “next” or “back” is entered, it will increment or decrease the page number by 1.

We plan on storing books as text files in another directory within the directory of the code, each within a directory whose name is the subgenre. The ArrayList will create the second and third layers based on these directory/subgenre names. The files will be read to present them to the user. As an example, a person could enter the Fiction category, then the Sci-Fi category, and could then read the book Futura Fantasia by Ray Bradbury.

There is also going to be a feature that allows for the user to save their place in a given book, which they will be asked upon exiting. At this time, they will also be asked if they would like to delete the page number stored by a previous user, if there is one. A file will store the last-saved page numbers for each book in the library, and when a user opens up a book, they will be given the choice of resuming reading from that page (if there is a page number saved).