

Junior Knights Object-Oriented Homework: Book Class

Write a Book class that describes a “Book” object.

Here are the instance variables of the Book class:

```
private String author;  
private String title;  
private int pages;  
private int curpage;
```

Here are the methods your class should include:

```
// Creates a book object with the given information, setting  
// the current page to 0.  
public Book(String writer, String thistitle, int numpages);  
  
// Advances the bookmark by numpages. If there are fewer pages  
// to read, then the curpage is set to the last page.  
public void read(int numpages);  
  
// Creates a new Book object with the same author as this Book  
// the same title with “II” appended to it, with numpages  
// number of pages and sets the bookmark to 0. This new object  
// is returned.  
public Book makeSequel(int numpages);  
  
// Returns the number of pages left to read in this Book.  
public int pagesLeft();  
  
// Returns the number of pages in the Book.  
public int getNumPages();  
  
// Returns a string representation of this book.  
public String toString();
```

The code above should be stored in Book.java

Program 1: Reading one book

In a separate class, RunBook.java, write a main method that takes care of the following:

- 1) Prompts the user for the necessary book information.
- 2) Simulates the user reading some pages from that book every day.
- 3) When the user finishes, the program should print out how many days it took for the user to finish the book, the average number of pages read a day, and automatically create a new Book object, the sequel, to the original book.
- 4) Then simulate the user reading some pages from this sequel every day.
- 5) When this is done, print out how many days the user took to read the sequel, the average pages read a day for this book, and the average pages read over the course of reading both books.

Follow the format shown below. In particular, assume that the user will always read a positive number of pages, but if they enter too many pages to read (more than are left in the book), provide an error message that states this and tells the user how many pages they actually read to finish the book.

After the first book is done, always ask the user how many pages are in the sequel. The program should terminate after the second book is read and the statistics are all printed.

Sample Run (User input in bold)

What is the title of your book?

TomSawyer

Who is the author?

MarkTwain

How many pages is the book?

248

Now you are reading TomSawyer by MarkTwain!

How many pages did you read on day 1?

40

How many pages did you read on day 2?

30

How many pages did you read on day 3?

108

How many pages did you read on day 4?

50

How many pages did you read on day 5?

26

You only had 20 pages left, so this is what you read.

Congrats on finishing your book!

You read 248 pages in 5 days for an average of 49.6 pages/day.

How many pages does the sequel have?

300

Now, you are reading TomSawyerII by MarkTwain!

How many pages did you read on day 1?

100

How many pages did you read on day 2?

100

How many pages did you read on day 3?

100

Congrats on finishing your book!

You read 300 pages in 3 days for an average of 100 pages/day.

In all, you read 548 pages in 8 days for an average of 68.5 pages/day.

Program 2: Reading several books

At the beginning of your program, ask the user to enter the information for all books in their library. Ask them if they have another book to enter. If they do, then ask them for the author, title and number of pages for that book and add that to the library (An ArrayList<Book>...) Once they don't have another book to enter, prompt the user with a main menu that asks them whether or not they want to read a book or quit. If they choose the former then show the user another menu of each book they have in their library (should print out author, title, total number of pages and # of pages left to read.) Ask the user which book they want to read. Have them select by index (list the indexes in the print out.) Then, ask the user how many pages they read. Update the appropriate book object and reprompt the main menu. Continue this process until the user chooses to quit off of the main menu.

SAMPLES TO BE ADDED LATER!!!