

OTE Career Shifter

Quiz 1

9/10/2024

Duration: 1 hour

Create a new Java project. Name it **Quiz1**. Share it on Github. Add the collaborators "tvarsamidis" and "iracleous". You must commit your code until the end of the quiz. Anything you commit after the end of the quiz, will not be taken into consideration.

In your project you will create 3 classes as shown below.

First, create the class "**Utilities**". Here you are going to create a few methods that will be used by other classes (some of the methods will be used, others will not be used).

QUESTION 1

Create a method called **getFirstRepeatedCharacter**(String text). The method must return the position of the first repeated character in the text. If the text has no repeated characters, return -1. For example:

`getFirstRepeatedCharacter("week") → 1`

`getFirstRepeatedCharacter("Hellooooo") → 2`

`getFirstRepeatedCharacter("Andersson") → 5`

`getFirstRepeatedCharacter("That's great!!!!") → 12`

`getFirstRepeatedCharacter("No repeated characters here") → -1`

QUESTION 2

Create a method called **goDownToOne**(int startNumber) which accepts a starting number and runs like this:

- if the number is even (can be divided exactly by 2) then divide it by 2 and keep the new value
- if the number is odd (not even) then multiply it by 2 and add 1 to it and keep the new value
- repeat the above process until you reach 1
- while running the process, print the new value of the number

So if we do

`goDownToOne(6)`

the output should show

6 3 10 5 16 8 4 2 1

The above means that 6 is even, so we divide by 2 and it becomes 3, 3 is odd so we multiply by 3 and add 1 and it becomes 10, 10 is even so it becomes 5, ...

QUESTION 3

Create a class called **Book**. A book has a **summary** (περίληψη) in it and a **number of pages**. We are not interested in other properties for the Book. Write the class, include the 2 attributes, getters and setters.

QUESTION 4

The Book must have a constructor which accepts a summary of its contents and the number of pages it has. For example, we can write the following for a book with 27 pages which talks about ancient cities:

`Book myResearch = new Book("This is a book about the ancient cities of Greece.", 27);`

QUESTION 5

The Book must also have a method called "boolean **isRepeatingSummary()**". This method returns 'true' if there are repeating characters in the summary, or 'false' if there are no repeating characters in the summary.

In the example of the previous book, **isRepeatingSummary** is true, because the summary has the word "Greece" in it, which has an 'e' character that repeats.

Use the method **getFirstRepeatedCharacter** to solve this question.

QUESTION 6

The book must have a method called "**showPagesToRead()**" which must use **goDownToOne** to show the reader how to read the book. The method **showPagesToRead** finds the middle page of the book, uses the middle page number with **goDownToOne**, and

goDownToOne starts printing the number of pages that the person must read. Maybe some pages will not exist in the book - we do not care, do not worry about it. Also, do not worry about books with 0 or 1 pages - we do not have any of these.

Therefore, if the book has 27 pages, **showPagesToRead** must start at page 13 and show the following:

13 40 20 10 5 16 8 4 2 1

QUESTION 7

Create a new class called **BookTest** with a main method. In main, create two book objects with the following data, and test their **isRepeatingSummary** and **showPagesToRead** methods, to prove that your methods work as expected.

Summary: "Exciting new recipes for people who want something quick an easy", 52 pages

Summary: "The big catalogue of all green animals ever existed", 55 pages

That's it, good luck!