Collections

Sr. No.	Assignment Question
1	Over the past 50 years, Arizona book publishers proved its excellent response and services towards the publishing industry and print media. It has a market record for printing 500 million copies of books in a single quarter. However, the manual process of printing the index for books having more than 5000 pages was a tedious and cumbersome process for the company. Now, the higher management has initiated process-management strategies to automate the index generation process for all sorts of books. To achieve this goal, consider you to be a part of the team that implements the solution for designing the application. Create an application in Java, which can be used to read a book as an input file, find new and distinct words with its line number, and print the same into a new output file. This application makes an index or concordance of all main words in a book along with their immediate contexts. A concordance lists all the words that occur in the book, along with all the line numbers on which each word occurs. (Words of length less than 3 are omitted, and "the" is omitted.) The index/concordance is written to an output file. The names of the input and output files must be specified on the command line when the application is run. The main class also makes use of a Comparator that can be used for comparing Strings according to alphabetical order. It should generate an error to apply this Comparator to objects that are non-strings. The application also creates a TreeMap that holds the concordance. Each word from the book file is used as a key in the map. The value associated with each word is a set that contains the line numbers on which the word occurs in the book file. The set contains values belonging to the wrapper class, Integer. The application should add a line reference to the concordance. The concordance is stored in the global variable, index. The word term has been found on line number lineNum. Another class is created as a subclass of FilterReader to provide methods for reading data expressed in human-readable ASCII text format. This class also defines three public subclasses of RuntimeException to represent errors that can occur during input. These classes are static nested inside the TextReader class. Fnally the application exits safely printing the number of distinct word found in the book and printing the index with the corresponding line numbers in the output file.