Name of the experiment: Write a Java application which will count the number of words of paragraph. You must do the counting using three threads.

Introduction: We have to define a class that will do the counting work. Then we have to define our own threads that will use the word counter class to count the words in a paragraph.

Objective:

- to learn how to use threads
- to learn how to use synchronization

Analysis: After analysing the problem we have found following components of the problem.

- we will have to define a helper class, CountWords that will open the file we want to read, then it will store all the contents into a buffer and then it will read one word at a time from the buffer.
- a class that implements Runnable interface, WordCounter, which will use the CountWords class and count all the words from it.
- a main class, which will contain the main method, which will demonstrate our solution.

From above analysis the conceptual class diagram is given below,

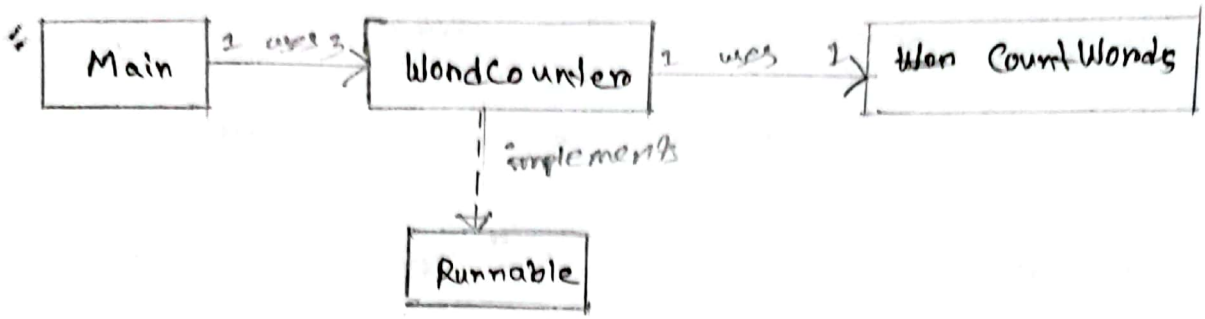

Figure-1: conceptual class diagram

Design: After analysing the problem, the design description of the solution is given below,

- class CountWords: the helper class that reads the paragraph word by word
  - Data Members:
    - text: a string that contains the whole text of the paragraph
    - index: current index into text string
  - Methods:
    - readNextWord: reads the next word from buffer and returns the word.

- class WordCounter: the class that actually counts the word in a paragraph.
  - Data Members:
    - countWords: a CountWords static typed object
    - paragraphToBeRead: to what percentage of paragraph has to be read by the current thread

- tName : name of the thread

- Methods:
  - run : overridden method from Runnable interface. this method counts all the words

- class Main : the main class of the program
  - Methods:
    - main : the main method of the program which will simulate the solution

From above design description the design architectural class diagram is given below,

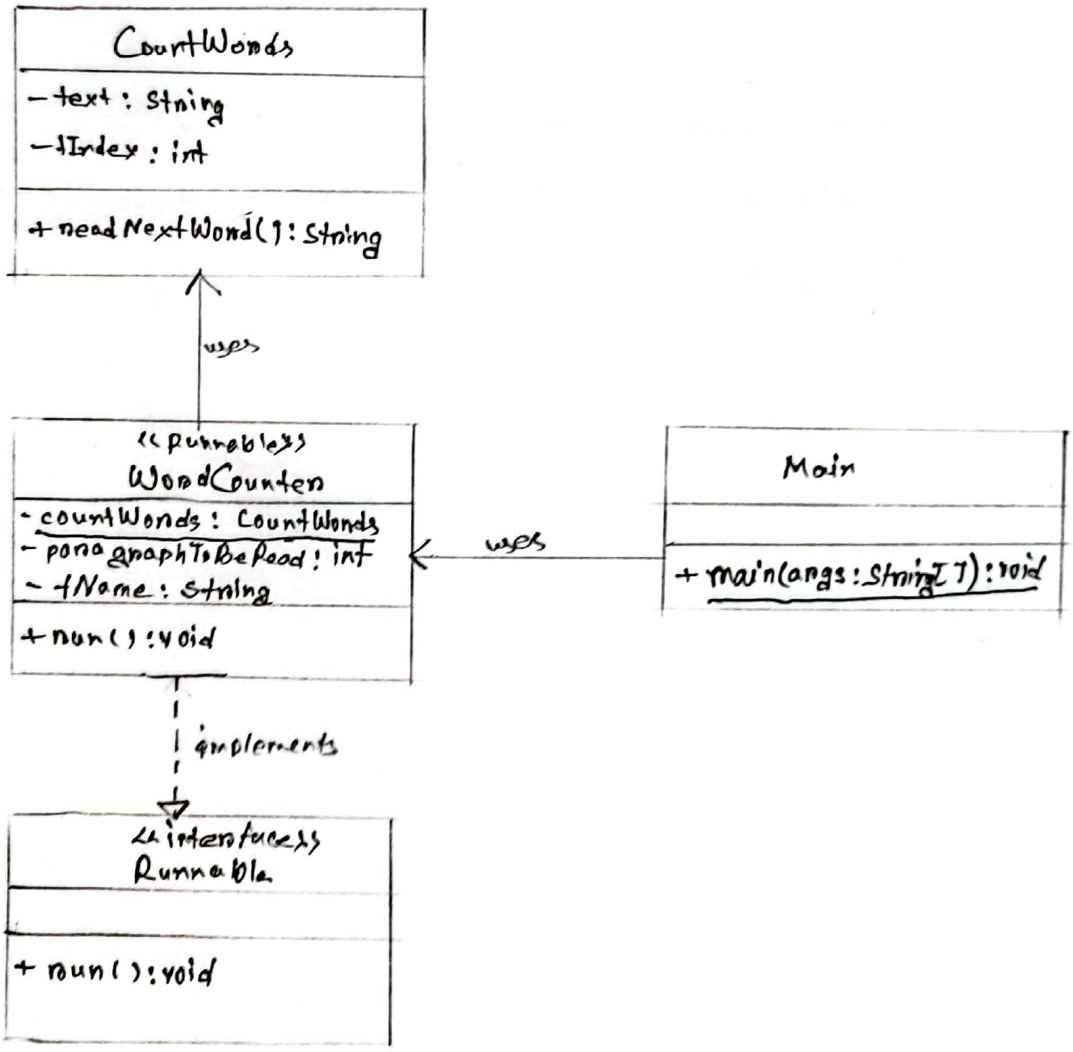

Figure-2 : architectural class diagram

From above design description, the pseudocode of the methods are given below:

CountWords :: readNextWord():

reads next word of the paragraph and returns it

WordCounter :: run():

count words and print 'counted words number

Main:: main():

simulate the solution using three threads. divide the whole paragraph into 3 part by giving the ~~per~~ numbers of what percentage of paragraph a thread will read.

Implementation:

* implementation is attached with the report *

Conclusion: we created a helper class to read the paragraph word by word, we created another class to actually count the words. And finally we created the Main class that simulated the solution using three threads.