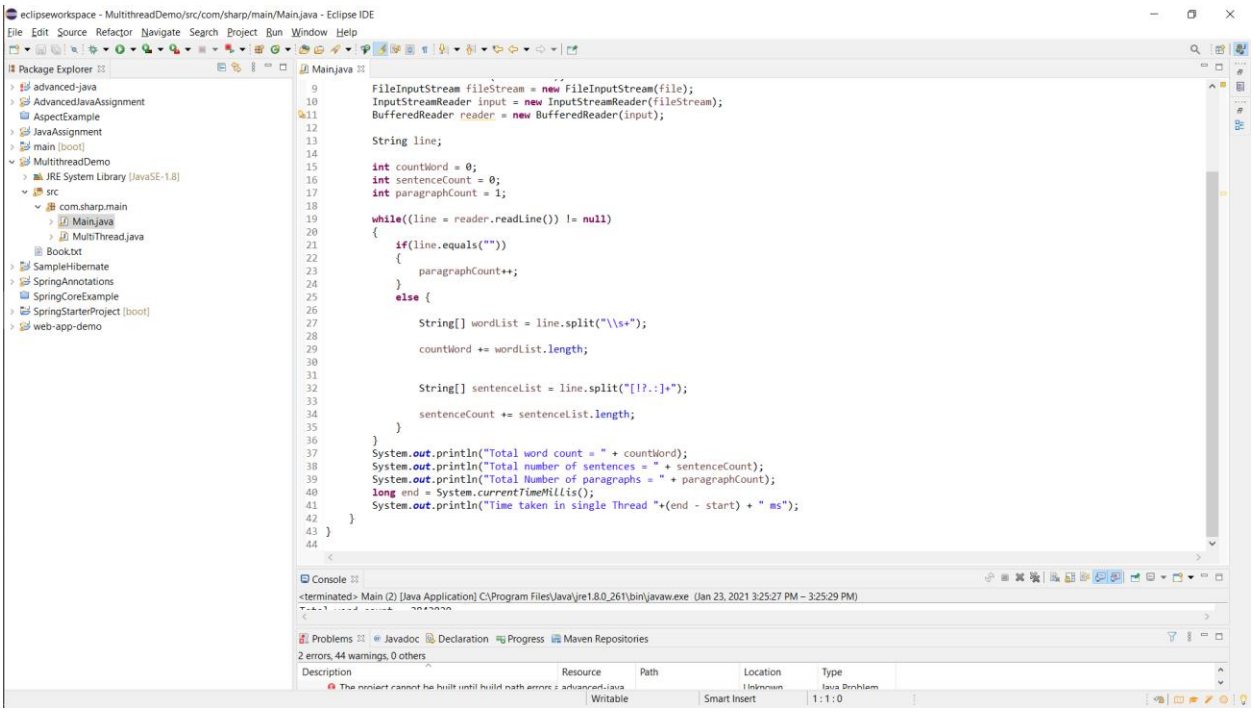
Spring AU ’21 – Java Multithreading – Morning Session

Name: Sheik Abudhahir K

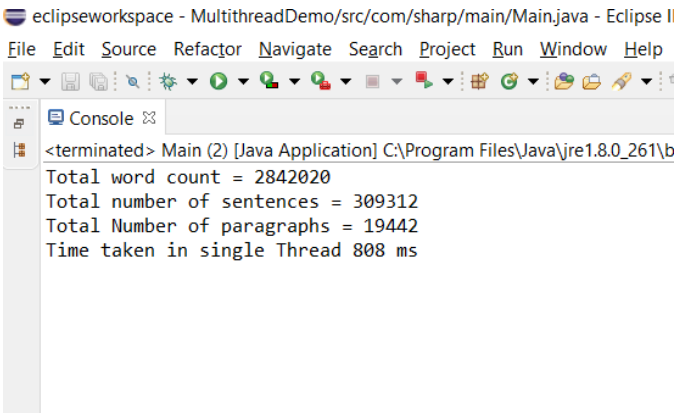
Date: 19/01/2021

Download the attached text document. It's book in plain text format. Analyze the book using a java program and perform a word count operation. You have to find out the count of each word in the book. Along with words, you also need to find out the number of paragraphs and sentences present in the book. Try it using single threading program and repeat the same through multi-threading approach. Make note of the performance improvement.

Program using single thread:



Output: Program executed in 808ms while run in single thread



Program using MultiThreading: Lets see the difference

Split the file content into two parts and run each part in each thread;

```
39
40        List<ArrayList<String>> parts  = new ArrayList<ArrayList<String>>();
41
42        for(int i = 0; i < list.size()/2; i++){
43          parts.add(new ArrayList<String>());
44          for(int j =0; j < 2; j++){
45            parts.get(i).add(list.get(i*2+j));
46          }
47        }
48
49        Thread t1 = new Thread(new Runnable() {
50
51            @Override
52            public void run() {
53                s.resource(list.size()/2, parts, 0);
54
55            }
56        });
57
58
59        Thread t2 = new Thread(new Runnable() {
60
61            @Override
62            public void run() {
63                s.resource(list.size()/2, parts, 1);
64            }
65        });
```

I call the same counting method that is user defined method, I attached the code as separate folder

Output: For execution using 2 threads both thread runs take maximum of 772 ms

```
<terminated> MultiThread [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe  (Jan 2
Ended
Total word count in Thread 1 = 1421063
Total number of sentences in Thread 1 = 154723
Total Number of paragraphs in Thread 1 = 9716
Time taken for each thread 765 ms
Total word count in Thread 2 = 1420956
Total number of sentences in Thread 2 = 154588
Total Number of paragraphs in Thread 2 = 9725
Time taken for each thread 772 ms
```

Split the file content into 5 parts and run each part in each thread;

```
        }

        Thread t1 = new Thread(new Runnable() {

            @Override
            public void run() {
                s.resource(list.size()/5, parts, 0);

            }
        });


        Thread t2 = new Thread(new Runnable() {

            @Override
            public void run() {
                s.resource(list.size()/5, parts, 1);
            }
        });

        Thread t3 = new Thread(new Runnable() {

            @Override
            public void run() {

                s.resource(list.size()/5, parts, 2);
            }
        });

        Thread t4 = new Thread(new Runnable() {

            @Override
            public void run() {

                s.resource(list.size()/5, parts, 3);
            }
        });

        Thread t5 = new Thread(new Runnable() {
```

Output: For using 5 threads it takes maximum 592 ms for execution

```
<terminated> MultiThread [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Jan 23, 2021 5:32:33 PM – 5:32:34 PM)
Ended
Total word count in Thread 4 = 568144
Total number of sentences in Thread 4 = 61799
Total Number of paragraphs in Thread 4 = 3936
Time taken for each thread 562 ms
Total word count in Thread 1 = 567992
Total number of sentences in Thread 1 = 61719
Total Number of paragraphs in Thread 1 = 3942
Time taken for each thread 565 ms
Total word count in Thread 5 = 568923
Total number of sentences in Thread 5 = 62004
Total Number of paragraphs in Thread 5 = 3821
Time taken for each thread 582 ms
Total word count in Thread 3 = 568293
Total number of sentences in Thread 3 = 61837
Total Number of paragraphs in Thread 3 = 3884
Time taken for each thread 584 ms
Total word count in Thread 2 = 568667
Total number of sentences in Thread 2 = 61952
Total Number of paragraphs in Thread 2 = 3856
Time taken for each thread 592 ms
```

Finally split the file into 10 parts and run each parts in each thread

It is the final code I uploaded in Git

Output: It takes maximum of 467 ms

Sometimes this time of execution varies because of my system performance