Assignment 1 (weeks 1, 2)

**Submission details:**

* individual assignment
* maximum 35 % of your mid-term 2
* deadline: Sunday, the 4th of January, 23:59
* defend during following weeks during the lessons.
* submit to moodle project zip file
* Your score is 0 if:
  + you refuse to defend your work
  + show project which is not the same (any file)
  + cannot explain logic, items properly.
  + waste time during presentation and it takes longer than 15 minutes

**Task 1 description:**

There is a directory with random number of files. The task is to change names to start from 1 to n.

Example input: file\_1.txt, file\_2.jpg, file\_3.mp3

Example output: 1.txt, 2.jpg, 3.mp3

Depending on the number of threads your PC support, you need to allocate each thread per file. Then loop. If your PC supports 2 threads: thread 1 changes name of file\_1.txt, thread 2 changes name of file 2.jpg then thread 1 changes name of file\_3.mp3. Optimize your solution with less work:

If there are already files like 1.txt, 2.jpg in the directory. Do you need to rename in that case or check-ing for name takes even more work? Depending on what?

**Task 2 description:**

There are several txt files. You need to search for particular text.

1) If there is exact match with the input text. If it is the title or file name put first in the search results, then order by frequency of appearance of this text in the files.

2) If there is no exact match, try to search without some words like the, a, in etc. Maybe there is a close match. Then also: if it is the title/file name put first, then order by frequency of appear-ance of this word in the text.

3) Try to make such search for pdf files.

4) Use threads to search in each file and call another thread to wait and aggregate the results when any other thread finishes the search.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Required Primary items (up to 3 points each)** | | | | |
| Extends Thread | Implements Runnable | Synchronized | Thread sleep | Executors |
| Method that returns value\* | shutdown, shutdownNow() | ArrayBlockingQueue | notify, notifyAll | Use shared value\*\* |
| **Required Secondary items (up to 1 point each)** | | | | |
| Good looking html/template | OOP (inheritance, multiple interface) | Collections (Lists, Maps) | Generics | Rest API |

\*We know that overridden run methods’ return type is void. Think/search in the method that returns some value after execution.

\*\* Some value (possibly static) that is shared among threads, so there is a concurrency.

**Assessment**

All the task details must work. After that Java Programming Language items are assessed.

**\*Late submission:**

From 0 seconds to 24 hours: - 10 points

24 hours – 48 hours: - 20 points

More than 48 hours: - 30 points

**Good Luck!**