

Laboratory Assignment #3

Part 1: Super Store with Inverted Index using Threads

You have developed the Super Store application in the previous laboratory assignment using threads. Repeat the same approach for getting the results using inverted index. This time, you are asked to find the total sales (TL) data for certain products. Use one thread per file with a data structure passed as argument to that thread. Once the threads are complete, aggregate sale results from the data structures and display them to the user according to their preference of the product. The sample run below is for 3 files, but your submissions are expected to run for all 12 files in our dataset.

Sample Run

```
C:\java ThreadedSuperStore 01-January.csv 02-February.csv 03-March.csv
```

```
Thread parsing 02-February.csv...
```

```
Thread parsing 03-March.csv...
```

```
Thread parsing 01-January.csv...
```

```
Threads are complete. Aggregating results.
```

```
Which product do you want to search?
```

```
B
```

```
For the product B,
```

```
In-store sales: £23435
```

```
Online sales: £32235
```

```
Total sales: £55670
```

Part 2: Super Store with Threads and Shared Data Structure

You have developed the Super Store application in the previous laboratory assignment using threads and an array of data structures. Your task now is to write the same Java application with the threads, this time by sharing the same instance of a data structure instead of working on separate instances.

You should follow these steps in your Java applications: (1) For each file passed as a parameter, you will need a thread to parse the file and a shared data structure passed to the threads to save the results. (2) Each thread reads the file assigned to it, and uses the shared data structure to add or update the sales statistics. (3) Once all threads are complete, display total sales (online & in-store) on the console.

Consider the flow diagram given in Figure 1. In the main method, a shared data structure is created -e.g., a `HashMap`- and its reference is passed to each thread along with the file names from the `args` parameter. Each thread works on the shared data structure and the file they are assigned to. Once they are all complete, the data structure should be readily available.

Sample Run

```
C:\java ThreadedSuperStore 01-  
January.csv 02-February.csv 03-  
March.csv
```

```
Thread parsing 02-February.csv...  
Thread parsing 03-March.csv...  
Thread parsing 01-January.csv...  
Threads are complete.
```

```
There are,  
In-store: £21100  
Online: £43940  
Total: £65040 worth of sales for  
all products.  
...
```

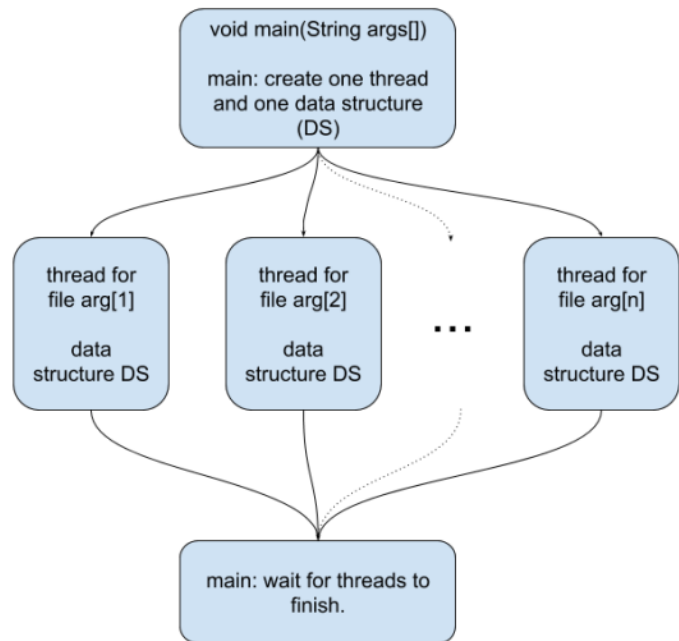


Figure 1: The flow diagram for both Super Store and Inverted Index.

NOTE: Part 1 & part 2 should be implemented separately: You may use two separate Java classes, each with different main functions.