Sabou Cezara
Computer Science, year 2
group 30421

Activities Management

1.Objectives

1.1Main objective

The main objective of this homework is to analyze the behavior of a person recorded by a set of sensors, which is stored in tuples : start time, end time and activity label. This analysis is done by performing some operations on the data.

1.2 Secondary objectives

The secondary objectives derive from the main objective:

1. The ability to use a bigger set of data
2. Being able to convert the given input into corresponding adequate candidates for processing;
3. Understanding the necessary steps to convert and process the given information;

2. Problem Analysis

Monitoring behavior is an interesting aspect of many studies performed on humans. The numbers performed on these averages could influence the future solutions of many of our daily problems. For instance, if some activity lasts longer than a certain amount of time, the industry involved in that certain activity could develop solutions to obtaining faster results.

I will describe the functionality of the system in the following paragraphs.

Before manipulating the monitored information, we firstly need to obtain it. We easily notice that the data is kept in those specific tuple, as mentioned in the Objectives section. This means that the data can easily be separated and stored into a special list of items. The two times present in the data set represent the time the activity started and the time the activity ended, that is the duration of the activity.

Reading the information is done through a file reader and converted from the text file into a stream, which by definition is “a sequence of elements from a source that supports aggregate operations”. The reason the streams are used in this homework is because we can perform those aggregate operations, such as, create a sequential stream, create a stream of functions(which I did for obtaining the information in the text file), filtering the information, mapping information, altering the information, grouping the information, all in just one line of code.

Sabou Cezara
Computer Science, year 2
group 30421

The required tasks are:

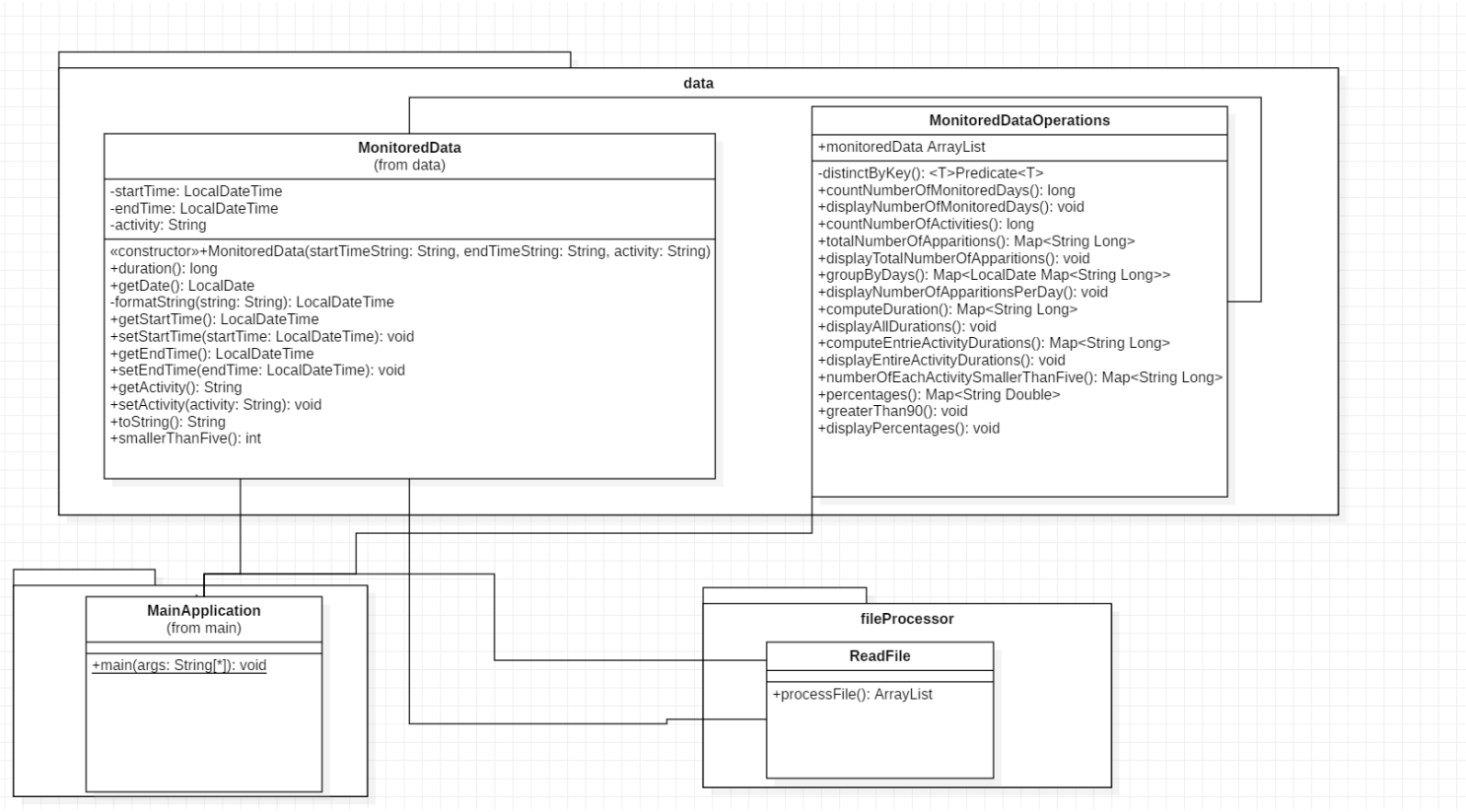
- Define the class MonitoredData with three fields : start time, end time and activity and create a list of objects of type MonitoredData;
- Count how many days of monitored data appears on the log;
- Count how many times each activity appeared over the entire monitoring period;
- Count how many times each activity appeared for each day of the monitoring period;
- For each line of the activity label the duration recorded;
- For each activity compute the entire duration of the monitoring period;
- Filter the activities that have 90% of the monitoring records with duration less than five minutes.

3.Design

The structure of the application is based on 3 main different packages; they are structured in the data package, where data is handled, the file processor, where the text file is processed, and the main package where the monitored data operations are called.

The application is designed to take the information from the file processor, which handles the text file through streaming and perform computations on the map created.

I created this UML diagram in order to show the main relationships of the application.



Sabou Cezara
Computer Science, year 2
group 30421

4.Implementation

In the data package:

4.1 Monitored Data

Data is kept in tuples : start time, end time and activity label class receives in its constructor the three attributes in string form, as the file reader splits the information in strings. The start time and the end time is then converted into “LocalDateTime” format, in order to gain easy access to the dates and also the times.

The duration method simply returns the difference measured in minutes between the end time and the start time. The getters and setters are used to get information to the MonitoredDataOperations class which processes the monitored data.

The smallerThanFive method returns 1 if the duration is smaller than five or not.

The toString method is overridden to return all the information about the monitored data object.

4.2 Monitored Data Operations

This class is used to perform all the requirements of the homework.

- ## Requirement 1: Count how many days of monitored data appears on the log

  For the counting the number of days of the monitoring period, I used the stream filtering operation, and took all the encountered objects through a verification method that kept record if they were seen before or not.

  In order to display this information I created another method of void type.

- ## Requirement 2: Count how many times each activity appeared over the entire monitoring period

  For counting the number of apparitions of each activity I used the method totalNumberOfApparitions, which creates a map of strings and long values, and takes the monitored data list and streams it, in order to be able to group the items in the list by activity name and count each apparition.

  In order to display this information I created another method of void type.

- ## Requirement 3: Count how many times each activity appeared for each day of the monitoring period

In order to count the number of apparitions of each activity I first had to group the items by day, which I have done by the groupingBy operation, and the argument of this operation was the local date, which represents the day. I created a new map of items which has as key the local date and as value another map which has as key the activity name and as value the number of activities.

I then transferred this newly obtained map to the other method which will give the final result : I iterate with a forEach operation through the key set and display all the values, which are in fac the activity names and their corresponding occurences.

- ## Requirement 4: For each line of the activity label the duration recorded

  The duration I chose to do it in the main class, as each object of Monitored Data already holds this attribute, and I believe it was the easiest way.

- ## Requirement 5: For each activity compute the entire duration of the monitoring period

  The entire duration was a bit more difficult to manage at first, but i figured out a solution in the end : I created a map which took as keys the activity names, and as values the durations. I the duration of each individual object using the Collectors.SummingLong().

  In order to display this information I created another method of void type.

- ## Requirement 6: Filter the activities that have 90% of the monitoring records with duration less than five minutes

  This was by far the hardest requirement. In order to obtain the result I had to first create a map which took as key the name of the activity, and the value was computed by counting the number of durations found which were smaller than five. After this I created a new map which was called percentages, which calculates the percentages of all the activities individually, for this I used the compute operation and inside I described a lambda expression that divided the number of activities smaller than five to the total number of activities of that kind and then multiplied it with 100. This way the information was only altered instead of creating a new map.
  After all these operations I created another method called greaterThan90, which filtered and kept only the percentages greater than 90% and placed them into a Map which took as key, again, the activity name, and as value the percentage, in order to show the correctness.

  In order to display this information I created another method of void type.

In the file processor package:

4.3 Read File

The name of the class is pretty self explanatory, as the class' job is to take the information from the file called "Activities.txt" and convert it into a stream. I noticed that the items were separated by double tabs. The map

was created using lambda expressions. The line was split with the double tabs, and then the three strings obtained are used to create a new MonitoredData object. The data is then added to the monitoredData arraylist.

In the main application package:

4.3 Main Application

This class simply declares the list of monitored data objects, then takes what is obtained from the file reader and instantiates a new monitored data operations object based on the monitored data arraylist.

5.Results

The results are displayed in the console, as there is no requirement for creating a graphical user interface or placing them into a file. Some of the results have been checked manually in order to verify the correctness.

I decided to print out some additional information in order to make the results easier to read and understand.

For example I chose to print the number of activities with duration smaller than five, as well as show the calculated percentages of all the activities, in order to check the validity.

6.Conclusions

In conclusion, the application is created in OOP style and it serves as a monitored data manipulation system. In this case, the regular user is not so involved , and so the graphical user interface is not needed, nor is the use case diagrams.

By developing this application I gained experience in using streams, getting information from a file and placing it into a map, as well as using lambda expressions in order to make computation easier and faster. I became more familiar to the Map objects and also linking the methods designed by me.

I am aware that the application is not yet in its final form, and there are some adjustments that need to be done. The input conversion could be improved. I also believe that there are more efficient ways in which I could implement the methods that process the input. I also would add other operations, such as : find the activity with biggest duration, create averages of the total durations of each activity, see most frequent activity in the monitoring period. In addition, in the future I can insist on arranging the application in an easier and more aesthetic way.

7.References

http://users.utcluj.ro/~cviorica/PT2019/5-PT-Laboratory-10.pdf http://javarevisited.blogspot.ro/2011/02/how-hashmap-works-in-java.html

http://coned.utcluj.ro/~salomie/PT_Lic/4_Lab/HW5_Tema5/HW5_Tema5.pdf

Sabou Cezara
Computer Science, year 2
group 30421

https://stackoverflow.com/

https://www.geeksforgeeks.org/lambda-expressions-java-8/