Programming Techniques
HOMEWORK 5

Student: Tudorica Andrei
Group: 30421

0. Contents

1. Objective of the homework

A smart house features a set of sensors that may be used to record the behavior of a person living in the house. The historical log of the person's activity is stored as tuples (startTime, endTime, activityLabel), where startTime and endTime represent the date and time when each activity has started and ended while the activity label represents the type of activity performed by the person: Leaving, Toileting, Showering, Sleeping, Breakfast, Lunch, Dinner, Snack, Spare_Time/TV, Grooming. The attached log file Activities.txt contains a set of activity records over a certain period of time. Define a class MonitoredData having startTime, endTime and activityLabel as instance variables and read the input file data into the data structure monitoredData of type List. Using stream processing techniques and lambda expressions introduced by Java 8, write the following set of short programs for processing the monitoredData.

1. Count the distinct days that appear in the monitoring data.
2. Determine a map of type that maps to each distinct action type the number of occurrences in the log. Write the resulting map into a text file.
3. Generates a data structure of type Map> that contains the activity count for each day of the log (task number 2 applied for each day of the log) and writes the result in a text file.
4. Determine a data structure of the form Map that maps for each activity the total duration computed over the monitoring period. Filter the activities with total duration larger than 10 hours. Write the result in a text file.
5. Filter the activities that have 90% of the monitoring samples with duration less than 5 minutes, collect the results in a List containing only the distinct activity names and write the result in a text file.

2. Problem Analysis

Big data defines itself as gigantic data sets that are so complex that classical data processing software applications can not manage these sets in convenient time and are unable to deal with them. The processes that are applied on these sets are: collect, storage or deposit, analyze and process, data curation, search and parse the sets, share them, transfer, visualize, query and update and last but not least security measures and data privacy measures. For performing these actions on big data structures different algorithms have been developed. For analyzing the data there is

the A/B testing natural language processing or machine learning. Big data defines itself as gigantic data sets that are so complex that classical data processing software applications can not manage these sets in convenient time and are unable to deal with them. The processes that are applied on these sets are: collect, storage or deposit, analyze and process, data curation, search and parse the sets, share them, transfer, visualize, query and update and last but not least security measures and data privacy measures. For performing these actions on big data structures different algorithms have been developed. For analyzing the data there is the A/B testing natural language processing or machine learning. Cloud computing and business intelligence and databases are big data storage solutions developed and still in development. An example of this kind of big data could be the fact that there are up to 5 billion mobile-phone subscriptions in this world. Managing the call history, messages and all the activities of these mobile phone subscriptions require advanced algorithms. Also up to 2 billion people access the internet daily. Logging and checking these activities also require very high level computing and managing applications. The Internet of things ot I O T is the network of physical devices, buildings, vehicles and all other kind of devices. Let's consider an example, in order to better understand the use cases of this application. Let's say we just bought a new smart house integrated system with a very large collection of sensors, able to record in real time what activity you process at a certain time and save all these activities in a log, with their start time, start date, activity name end time and end date. We bought this system in order to improve the time management we use and optimize the way we organize the day. That is why we would need a software application to manage and deal with the data received from the hardware system and compute a set of analysis on the data, giving as a result a set of statistics like the count of days in the recordings, on the activities with the duration over a certain time, or the number of occurrences of a certain activity in a day or other kind of data that could interest us. This networking system allows these devices to communicate with each other and exchange data, in order to improve the activity of these devices. The problem we focus on is managing data received from a set of sensors in a IOT house automation system, recording the daily activities of the owner . We could use collections from java, but starting with Java 8 there is an updated API bringing streams. Streams work on sets of data using multi-core processing, even if the programmer doesn't have to use a single line of multiprocessed or multi threadded code. This feature brings an entire set of actions that can be performed on any collection like filters, mapping instructions, sort, sum, collect instructions to bring filtered elements from the stream to a collection.

3. Modelling

The process of modelling is defined as the process of modularising a big problem into smaller ones, that are easier to understand and debug. This also helps making an abstract idea more clear. In software development, modelling is essential in order to build an application that has a strong background structure.

As stated above, we need to process the entries of a log file and count the distinct days that appear in the monitoring data. Determine a map of type that maps to each distinct action type the number of occurrences in the log. Write the resulting map into a text file. Generates a data structure of type Map> that contains the activity count for each day of the log (task number 2 applied for each day of the log) and writes the result in a text file. Determine a data structure of the form Map that maps for each activity the total duration computed over the monitoring period. Filter the activities with total duration larger than 10 hours. Write the result in a text file. Filter the activities that have 90% of the monitoring samples with duration less than 5 minutes, collect the results in a List containing only the distinct activity names and write the result in a text file. The architecture of the application is really simple as it contains a class of monitored data entries and a main processing unit with a GUI.

4. Use cases

Let’s consider an example, in order to better understand the use cases of this application. Let’s say we just bought a new smart house integrated system with a very large collection of sensors, able to record in real time what activity you process at a certain time and save all these activities in a log, with their start time, start date, activity name end time and end date. We bought this system in order to improve the time management we use and optimize the way we organize the day. That is why we would need a software application to manage and deal with the data received from the hardware system and compute a set of analysis on the data, giving as a result a set of statistics like the count of days in the recordings, on the activities with the duration over a certain time, or the number of occurrences of a certain activity in a day or other kind of data that could interest us.

5. Design

In the design process two classes were used : the monitored data class and the main processing class that also contains the minimal graphical user interface. The monitored data class contains two date time elements the start date time and the end date time. It also contains the activity label stored as a string. The class contains an empty constructor, a setter for the start time, a setter for the end time and a setter for the activity label. These setters are used for building the class instances from the data read from the log file. The class also contains a getter for the start time , a getter for the end time and also a getter for the activity label. The monitored data

class also contains a to string method that returns a user readable version of the class instance as a string. The last method in the monitored data structure is the get length of activity method, that, as stated in its name, returns the duration of a certain activity in milliseconds, as a difference of the end time minus the start time. The main class, the main processing unit of the software application contains two methods. The methods are the graphical user interface method and the main method. The graphical user interface is rather a simple one. It contains a frame. Inside the frame there is a tabbed pane called pnlMain, containing 5 Jpanels, one for each of the 5 tasks to be solved. In each of the five panels in the tabbed pane there is a text area containing the response for each of the assignments. We load the results in these text areas from files corresponding to the output of the exercises, using the read all lines function in the Files package. Then using the stream method for each we print the lines from the files in the text areas. The text areas are then added to the corresponding panels. The main class, the main processing unit of the software application contains two methods. The methods are the graphical user interface method and the main method. The graphical user interface is rather a simple one. It contains a frame. Inside the frame there is a tabbed pane called pnlMain, containing 5 Jpanels, one for each of the 5 tasks to be solved. In each of the five panels in the tabbed pane there is a text area containing the response for each of the assignments. We load the results in these text areas from files corresponding to the output of the exercises, using the read all lines function in the Files package. Then using the stream method for each we print the lines from the files in the text areas. The text areas are then added to the corresponding panels. The main method uses a string to store the file name of the log and an array list of the monitored data that will be filled using the contents of the activity log file received from the house automation hardware system. A stream method collect is used on the result of the lines function in the Files package to save and store each entry of the log file as a string in an array list of strings. After the file is read, using each line of the activity log stored previously we generate the array list of monitored data, by splitting each line and converting the elements from strings to datetimes where needed. At the beginning of the file a simple date format called formatter was defined to set the structure we will use for our dates and times. We convert the strings to java utils DAte elements and using those dates we create DateTime elements from the Joda time java package and generate the monitored data. For the second assignment, as required,we build a map of type < String Integer > that maps to each distinct action type the number of occurrences in the log. For the third assignment, as required,we build a map of type < Integer Map <String Integer>> that contains the activity count for each day of the log (task number 2 applied for each day of the log) and writes the result in a text file. For the fourth assignment, as required,we build a map of type <String DateTime> that maps for each activity the total duration computed over the monitoring period. Filter the activities with total duration larger than 10 hours. For the fifth assignment, as required,we build a list of strings that

filters the activities that have 90% of the monitoring data samples with duration less than 5 minutes. Next, for the first assignment we generate an array list of integers for all the days in the log file saved as the index of that certain day in the year. We also add the end times dates to this arraylist. The result for the first exercise will be the count of different elements in this array list. For the second assignment, as required,we build a map of type < String Integer > that maps to each distinct action type the number of occurrences in the log. For the third assignment, as required,we build a map of type < Integer Map <String Integer>> that contains the activity count for each day of the log (task number 2 applied for each day of the log) and writes the result in a text file. For the fourth assignment, as required,we build a map of type <String DateTime> that maps for each activity the total duration computed over the monitoring period. Filter the activities with total duration larger than 10 hours. For the fifth assignment, as required,we build a list of strings that filters the activities that have 90% of the monitoring data samples with duration less than 5 minutes.

6. Results

The result of the development of this application, that lasted for two weeks together with writing the documentation, is a software easy to use that makes it easy for the user to manage a couple of entries of a log received from a house automation I O T system and generating statistics on a couple of topics: count of the distinct days that appear in the monitoring data, determine a map of type that maps to each distinct action type the number of occurrences in the log, write the resulting map into a text file, generates a data structure of type Map> that contains the activity count for each day of the log (task number 2 applied for each day of the log) and writes the result in a text file. Determine a data structure of the form Map that maps for each activity the total duration computed over the monitoring period. Filter the activities with total duration larger than 10 hours, write the result in a text file. Filter the activities that have 90% of the monitoring samples with duration less than 5 minutes, collect the results in a List containing only the distinct activity names and write the result in a text file.

7. Further development

- More statistical data could be generated over the log files received from the hardware systems.
- With a more complex house automation system, the application could enlarge the range of topics to study and analyze in the activity of the user
- The application could give the user advice on how to improve different elements in the daily behaviour, like eating at better hours, eating slower (

over a longer period of time), using the TV less or sleeping more or in certain periods of time.

8.Conclusions

During the development of this project i learned and improved of several skills:

- A better use of Eclipse
- A better use of the Java coding language
- A better understanding and usage of the object oriented programming paradigms
- More organized and clear code
- Programming for the first time as i have never done that before in my life.
- Learning about serialization.
- A better understanding of the graphical user interface elements like Jtable and Tabbed pane.
- Learning about design pattern observer.
- Learning about design by contract.
- Learning about Java 8 Streams and how they can be used in managing large sets of data.
- An introduction along with a better understanding of lambda expressions.

9. Bibliography

1. Java general documentation https://docs.oracle.com/javase/7/docs/api/
2. Lecture and laboratory guidelines http://www.coned.utcluj.ro/~salomie/PT_Lic/3_Lab/
3. Word counter https://wordcounter.net/