***Programing Techniques***

*Assignment 5:*

***PROCESSING SENSOR DATA OF DAILY LIVING ACTIVITIES***

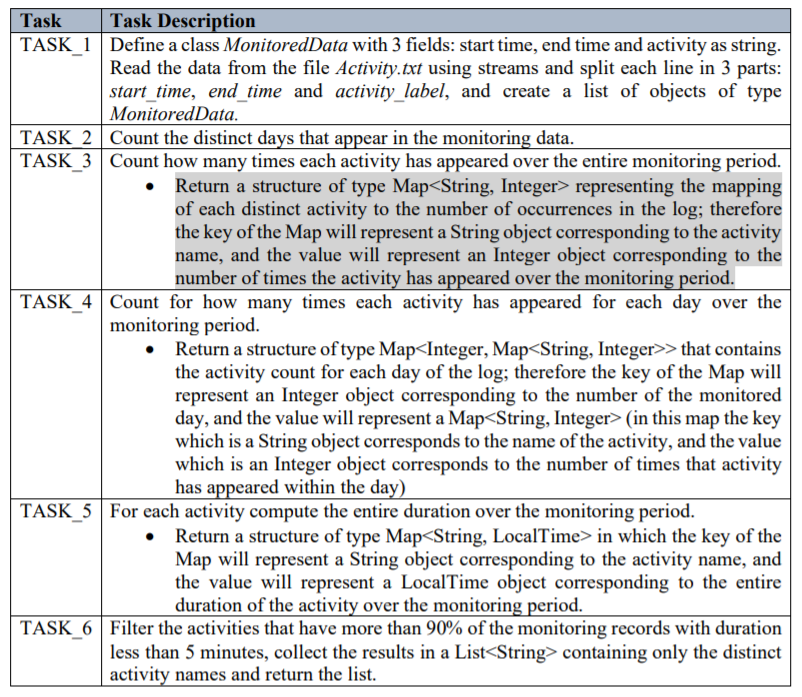
* *Rus Rares Tudor*
* *Group 30422*

*Contents:*

1. *Requirements . . . . . . . . . . .. . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ..3*
2. *Problem Analysis. . . . . . . . . . .. . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4*
3. *Use Cases . . . . . . . . . . .. . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5*
4. *Projection . . . . . . . . . . .. . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .6*
5. *Implementation . . . . . . . . . . .. . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .* **7**
6. *Conclusions . . . . . . . . . . .. . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8*
7. *Bibliography . . . . . . . . . . .. . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . 8*

*--Requirements :*

The main objective of this assignment is to design and implement a system for processing sensor data of daily living activities. Receiving a file in text format, representing a daily activities of a person for more days. I had to implement different tasks based on this activities using lambda expressions and streams.

Here is the official: Consider designing, implementing and testing an application for analyzing the behavior of a person recorded by a set of sensors installed in its house. The historical log of the person’s activity is stored as tuples (start\_time, end\_time, activity\_label), where start\_time and end\_time 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 data is spread over several days as many entries in the log Activities.txt. Write a program that uses functional programming in Java with lambda expressions and stream processing to perform the tasks listed in the table below. The results of each task must be written in a separate .txt file (each .txt file must be named according to the following template task\_number.txt).

*--Problem Analysis:*

Abstraction in its main sense is a conceptual process where general rules and concepts are derived from the usage and classification of specific examples, literal signifiers, first principles, or other methods.

Conceptual abstractions may be formed by filtering the information content of a concept or an observable phenomenon, selecting only the aspects which are relevant for a particular subjectively valued purpose. For example, abstracting a leather soccer ball to the more general idea of a ball selects only the information on general ball attributes and behavior, excluding, but not eliminating, the other phenomenal and cognitive characteristics of that particular ball. In a type–token distinction, a type (e.g., a 'ball') is more abstract than its tokens.

In the requirement I was asked to use lambda expressions and streams, so here are the definitions for both:

Lambda expressions basically express instances of functional interfaces (An interface with single abstract method is called functional interface. An example is java.lang.Runnable). lambda expressions implement the only abstract function and therefore implement functional interfaces

lambda expressions are added in Java 8 and provide below functionalities.

* Enable to treat functionality as a method argument, or code as data.
* A function that can be created without belonging to any class.
* A lambda expression can be passed around as if it was an object and executed on demand.

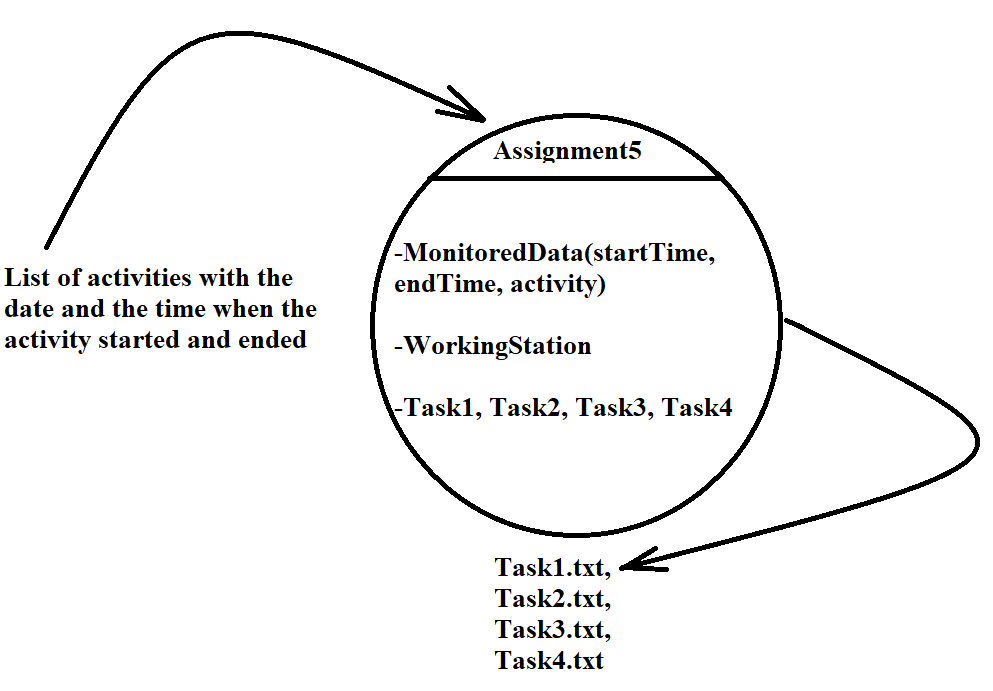
Introduced in Java 8, the Stream API is used to process collections of objects. A stream is a sequence of objects that supports various methods which can be pipelined to produce the desired result.  
The features of Java stream are –

* A stream is not a data structure instead it takes input from the Collections, Arrays or I/O channels.
* Streams don’t change the original data structure, they only provide the result as per the pipelined methods.
* Each intermediate operation is lazily executed and returns a stream as a result, hence various intermediate operations can be pipelined. Terminal operations mark the end of the stream and return the result.

Using streams the following operations can be used, most of them were also used in the implementation of the methods:

1. **map:**The map method is used to map the items in the collection to other objects according to the Predicate passed as argument.
2. **filter:** The filter method is used to select elements as per the Predicate passed as argument.
3. **sorted:** The sorted method is used to sort the stream.
4. **collect:** The collect method is used to return the result of the intermediate operations performed on the stream.
5. **collect:** The collect method is used to return the result of the intermediate operations performed on the stream.
6. **distinct:** The distinct method is used to filter or collect all distinct elements from a collection.

*--Use Cases:*



A user has to give a list of tuples as the following: the start time and date of the activity, the end time and date of the same activity and the name of the specified activity. Based on how much the application is implemented it should return a bunch of files in text format, these files are the tasks that was required by this assignment.

- *User Main SUCCES scenario*:

-User launches application( jar file) successfully.

-User provides relative inputs based on the task.

-The date introduced by the user are correctly processed.

-The running is performed successfully with no exceptions or errors.

-The results are displayed in more files in text format returned by the application.

*- User Main FAILED scenario*:

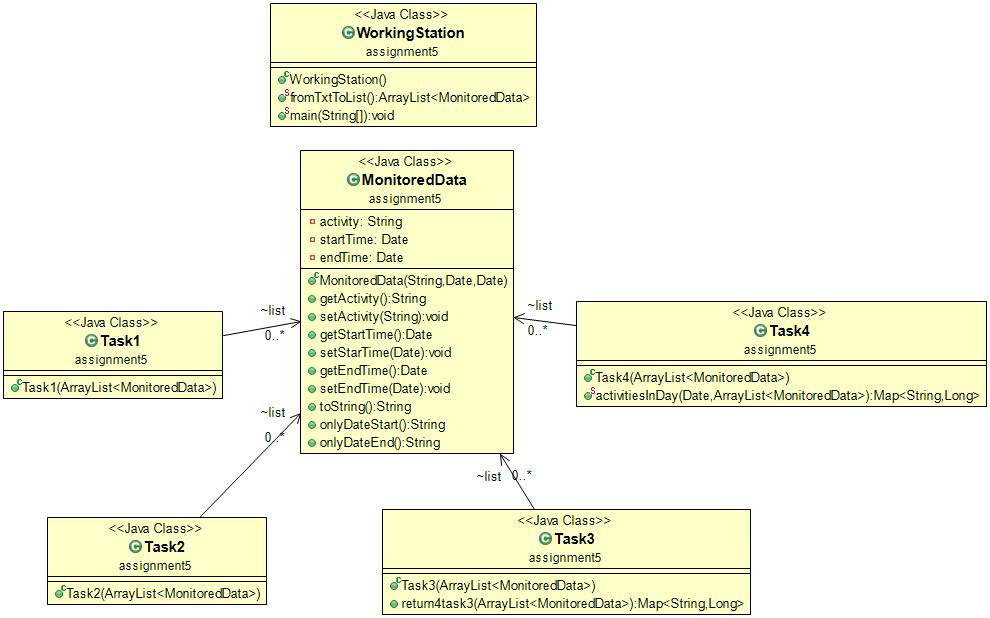
-User can’t launches application successfully.

-User provides wrong inputs for the requirements.

-The inputs are unsuccessfully transformed into the MonitoredData parameters.

-The resulted files are empty or note generated.

*--Projection:*



*--Implementation:*

In the following, we will take a closer look at the role and the implementation of each package, class with its attributes, constructors and methods. All my classes are implemented in the same package to avoid errors.

--*MonitoredData.java*

This is the class where the required data format is created, the date is recording a he behavior of a person recorded by a set of sensors installed in its house. The historical log of the person’s activity is stored as tuples (startTime, endTime, activity), 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. Firs methods observed are the seters and geters for each of the arguments of this class, I over rider the toString method so when this super method is called it should print a string not a record of data. In addition I implemented two methods which are doing the same thing and that is formatting the date format of each attribute.

--WorkingStation.java

Here, in this class, the main() method is present, in which the tasks are called, with a list of MonitoredData elements as parameter. In addition, a method that is very important and also is the task 1 requirement, is created, this method take the location of the text file Activities and line by line transforms a string into a MonitoredData type.

*--Task1.java*

Define a class MonitoredData with 3 fields: start time, end time and activity as string. Read the data from the file Activity.txt using streams and split each line in 3 parts: startTime, endTime and activity, and create a list of objects of type MonitoredData.

*--Task2.java*

Count the distinct days that appear in the monitoring data.

*--Task3.java*

Count how many times each activity has appeared over the entire monitoring period. Return a structure of type Map representing the mapping of each distinct activity to the number of occurrences in the log; therefore, the key of the Map will represent a String object corresponding to the activity name, and the value will represent an Integer object corresponding to the number of times the activity has appeared over the monitoring period.

*--Task4.java*

Count for how many times each activity has appeared for each day over the monitoring period. Return a structure of type Map> that contains the activity count for each day of the log; therefore the key of the Map will represent an Integer object corresponding to the number of the monitored day, and the value will represent a Map (in this map the key which is a String object corresponds to the name of the activity, and the value which is an Integer object corresponds to the number of times that activity has appeared within the day).

*--Conclusion:*

Java offers the real possibility that most programs can be written in a type-safe language. ... It extends Java with a mechanism for parametric polymorphism, which allows the definition and implementation of generic abstractions. The paper gives a complete design for the extended language

I personally, learned and better understood how to use classes and modifiers (which sets the class as public or private), modifiers are also used for variables and methods and also I have my first interaction with the threads and that’s awesome, but in my opinion are kind hard to understand and to work with them, but I had a lot of fun and I like it, because who doesn’t like an tuff challenge. The hardest thing to understand and work with lambda expressions and with streams, but after I start to understand them the requirements were more simple that if I worked with normal iterations.

What’s concerning the further developments, I have some ideas to adjust the interface to make it more attractive because in our days this is what an user is looking for, he or she has no concerns about how the application works or why it takes so much memory. Regarding to possible updates of the application, I have in mind to implement an algorithm which finds the roots of the polynomial and returns a graphical schema and also to increase performance so the application can work faster.

*--Bibliography:*

- <https://www.geeksforgeeks.org/>

- <https://www.wikipedia.org/>

- <https://stackoverflow.com/>