

Activity Management

Student: Suciu Mihai

Group: E_30422

Cuprins

<i>1. Objective</i>	<i>2</i>
<i>2. Problem analysis, assumptions</i>	<i>2</i>
<i>2.1. Problem analysis.....</i>	<i>2</i>
<i>2.2.Assumptions</i>	<i>3</i>
<i>3.Design</i>	<i>4</i>
<i>3.1 UML diagram</i>	<i>4</i>
<i>3.2 Class design.....</i>	<i>5</i>
<i>4. Implementation and testing.....</i>	<i>5</i>
<i>5. Results</i>	<i>5</i>
<i>6.Conclusions</i>	<i>6</i>
<i>7. Bibliography</i>	<i>6</i>

1. Objective

The requirement of this project is :

Lambda Expressions and Stream Processing

Consider the task of analyzing the behavior of a person recorded by a set of sensors. 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, taken from [1,2] and downloadable from the file Activities.txt located in this folder. Write a Java 1.8 program using lambda expressions and stream processing to do the tasks defined below.

This problem can be split as follows:

- Creating the Monitor Data class.
- Creating the file reader.
- In main use file reader to create a list of Monitor data and apply the functions to solve the requirements.

2. Problem analysis, use case diagram, assumptions

2.1 Problem analysis

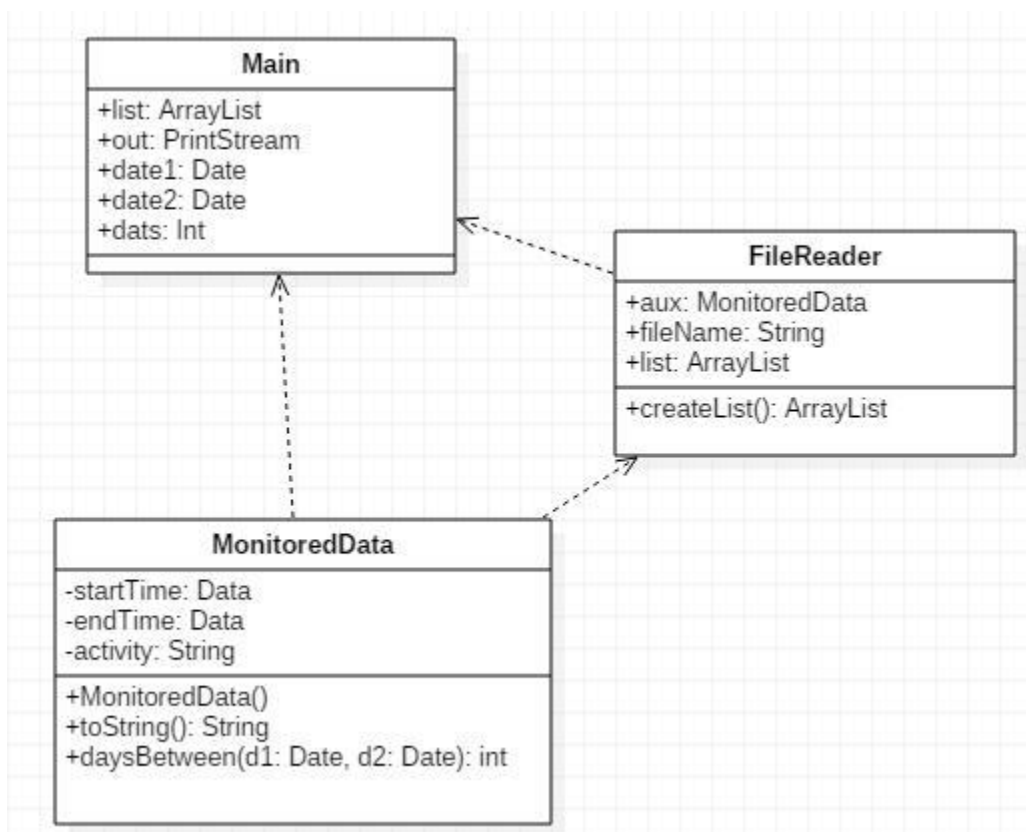
- For this implementation I used design pattern Model-View-Controller (MVC). **Model-view-controller (MVC)** is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development.
- The *model* is the central component of the pattern. It expresses the application's behavior in terms of the problem domain independent of

the user interface. It directly manages the data, logic and rules of the application.

- A *view* can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
- The third part or section, the *controller*, accepts input and converts it to commands for the model or view.
- The model is responsible for managing the data of the application. It receives user input from the controller.
- The view means presentation of the model in a format.
- The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.

3.Design

3.1 UML Diagram



3.2 Class design

The projection of this application started Monitored data. I will represent below each class with their attributes. Here I have 2 attributes of data type. Start time holds the first data and the time.

4.Implementation

Classes & important methods

First class is Monitored Data. Here I have 3 attributes , one constructor and getters and setters for each attribute + one function which calculates days between.

5.Results

How many days of monitored data appears in the log: 14

Breakfast : 14

Grooming : 51

Toileting : 44

Sleeping : 14

Leaving : 14

Spare_Time/TV : 77

Snack : 11

Showering : 14

Lunch : 9

6.Conclusions

During the design of this project I improved my abilities in classes. Before this I learned how to use the concept of mapping. More useful thinks in this project is lambda expression and streams.

Further development: solve more requirements.

7.Bibliography

<https://dzone.com/articles/java-streams-groupingby-example>

<http://winterbe.com/posts/2014/07/31/java8-stream-tutorial-examples/>

https://stackoverflow.com/questions/7103064/java-calculate-the-number-of-days-between-two-dates?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa