



DOCUMENTATIE TEMA 3

House Activity

Akysh Baymuhammedov

30221

Profesor Laborator Assist Antal Marcel

Contents

1. Cerinte Functionale	3
2. Obiective	3
2.1. Obiectiv Principal:	3
2.2. Obiective Secundare:	Ошибка! Закладка не определена.
3. Analiza Problemei	3
4. Proiectare	4
4.1. Diagrama de clase	5
4.2. Algoritmi	Ошибка! Закладка не определена.
5. Implementare	5
6. Concluzii si Dezvoltari Ulterioare	7
7. Bibliografie	7

1. Cerințe Functionale

(deduse din barem și cerința problemei)

Scrieti un program care citește din fișierul `activity.txt` activității și datele de început și sfârșit.

2. Obiective

2.1. Obiectiv Principal:

Scrieti program prin folosire update de Java 8 folosind expresii lambda și streamuri.

2.2.

Task	Description	Grading
0	Define a class <i>MonitoredData</i> with 3 fields: start time, end time and activity as string. Read the data from the file <i>Activity.txt</i> using streams and create a list of objects of type <i>MonitoredData</i> .	2 points
1	Count the distinct days that appear in the monitoring data	1 point
2	Determine a map of type <code><String, Integer></code> that maps to each distinct action type the number of occurrences in the log. Write the resulting map into a text file	2 points
3	Generates a data structure of type <code>Map<Integer, Map<String, Integer>></code> 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	1 point
4	Determine a data structure of the form <code>Map<String, DateTime></code> 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.	1 point
5	Filter the activities that have 90% of the monitoring samples with duration less than 5 minutes, collect the results in a <code>List<String></code> containing only the distinct activity names and write the result in a text file.	2 points
7	Documentation	1 point

3. Analiza Problemei

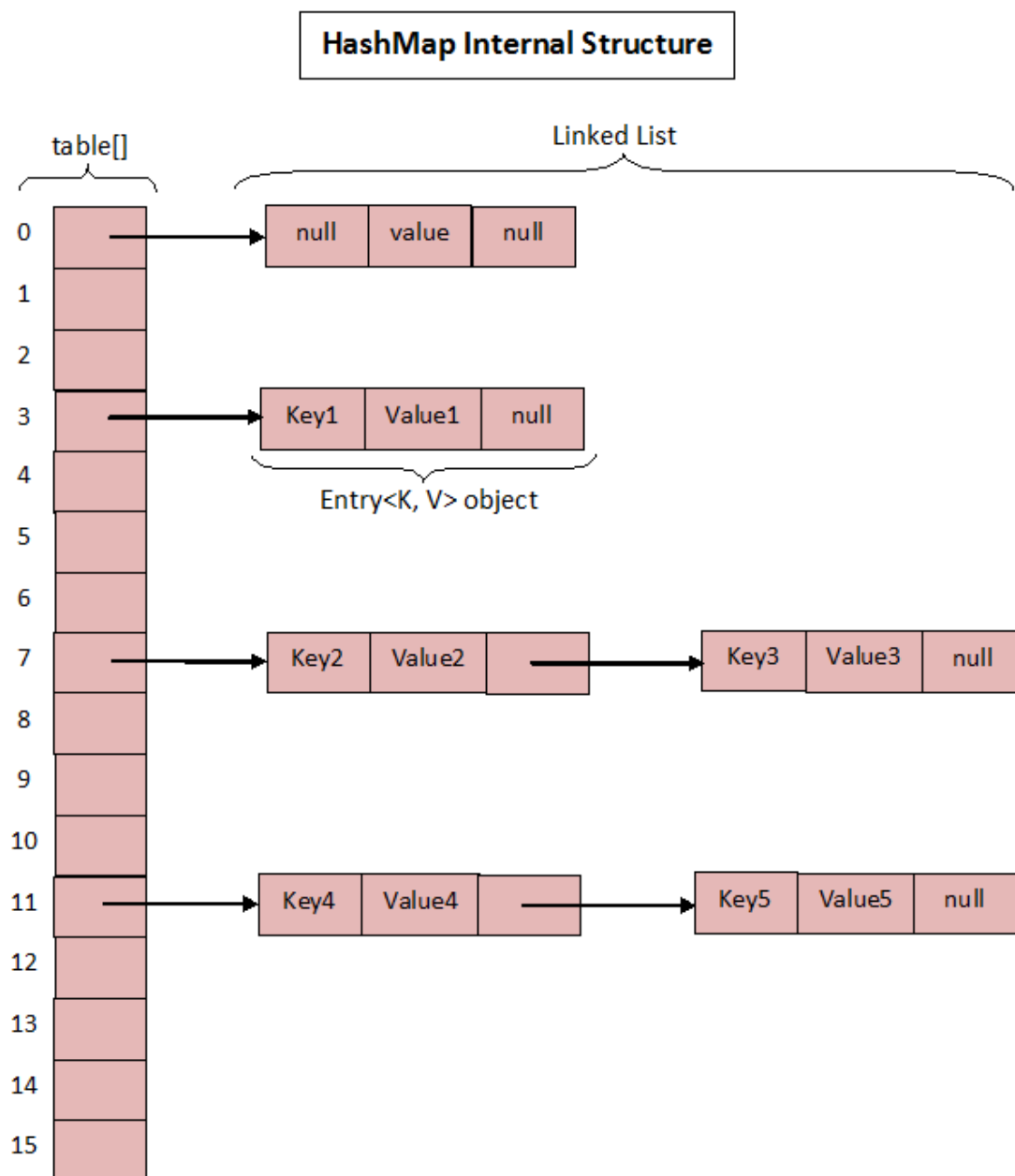
Use cases:

1. Numarare zilele distincte din fișier.
2. Numarare activități distincte și scriere în fișier și numarare de câte ori apare în fișier.
3. Generare un map care păstrează ziua și numarare activității distincte pentru ziua respective și scriere la un fișier.
4. Determinare un map care păstrează activității și calculează durata totală și apoi scriere în fișier.
5. Filtrare activității care au 90% durată mai mic decât 5 minute și păstrare într-o listă și apoi scriere în fișier.

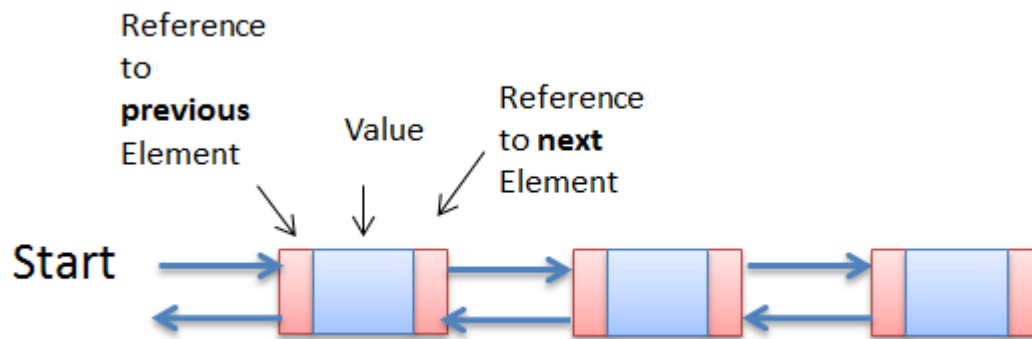
4. Proiectare

4.1. Structuri de date:

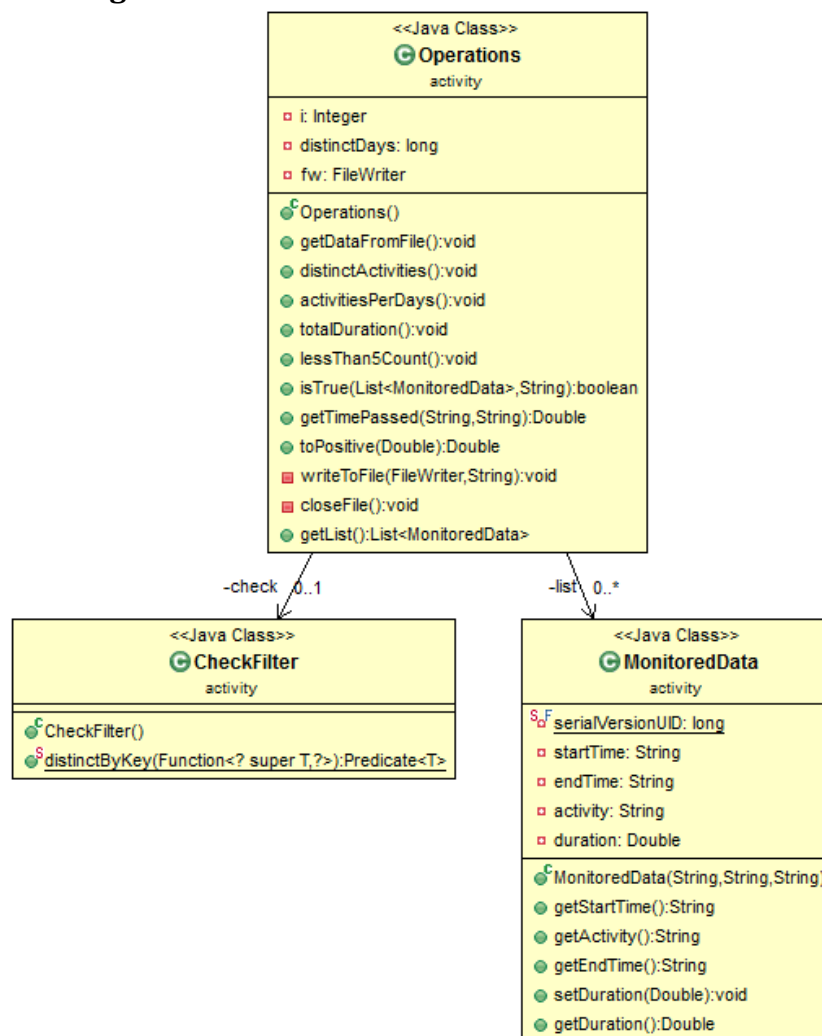
Am folosit map-uri pentru pastrarea datele:



Si lista:



4.2. Diagrama de clase



5. Implementare

Am 3 Clase MonitoredData, Operations si CheckFilter.

MonitoredData pastreaza datele de la fisier Activity.txt.

Clasa Operations face operatii:

Are campuri:

```
private List<MonitoredData> list = new ArrayList<MonitoredData>(); //pastrare
datele din fisier.
private long distinctDays; //Pentru calculare zilele distincte.
private CheckFilter check; //Apelare clasa check pentru filtrarea activitatii
distincte.
private FileWriter fw; //Pentru scriere la fisier.
```

Metodele:

```
public void getDataFromFile() throws IOException, ClassNotFoundException;
```

Citeste din fisier linie cu linie apoi il imparte la cuvintele si creaza un obiect de tip MonitoredData si il pastreaza in fisier. Apoi numereaza zilele distincte si afiseaza.

Apoi apeleaza metodele distinctActivities(); activitiesPerDays();
totalDuration(); lessThan5Count();

```
public void distinctActivities();
```

Genereaza un Map care are un alt map inaintur si citeste lista cu valori si numereaza numari de aparire activitatii pentru fiecare zi. Apoi scrie rezultatul in fisierul activitiesPerDay.txt.

```
public void totalDuration();
```

Calculeaza durata totala pentru fiecare activitatea si il pune intr-un map, apoi filtreaza activitati care au durata mai mult decat 10 ore si il scrie la fisierul "totalDurationMoreThan10H.txt".

```
public void lessThan5Count();
```

Scrie intr-o lista activitatii care au 90% durata lui mai mic decat 5 minute si apoi il scrie in fisierul ActivitiesLessThan5Min.txt".

```
public boolean isTrue(List<MonitoredData> list, String nume);
```

Este metoda care e apelat de catre metoda lessThan5Count() in timpul filtrarii datele. Metoda calculeaza procentul de durati care sunt mai mici de 5 minute si daca e egal sau mai mare decat 90 procent atunci returneaza true alt fel false.

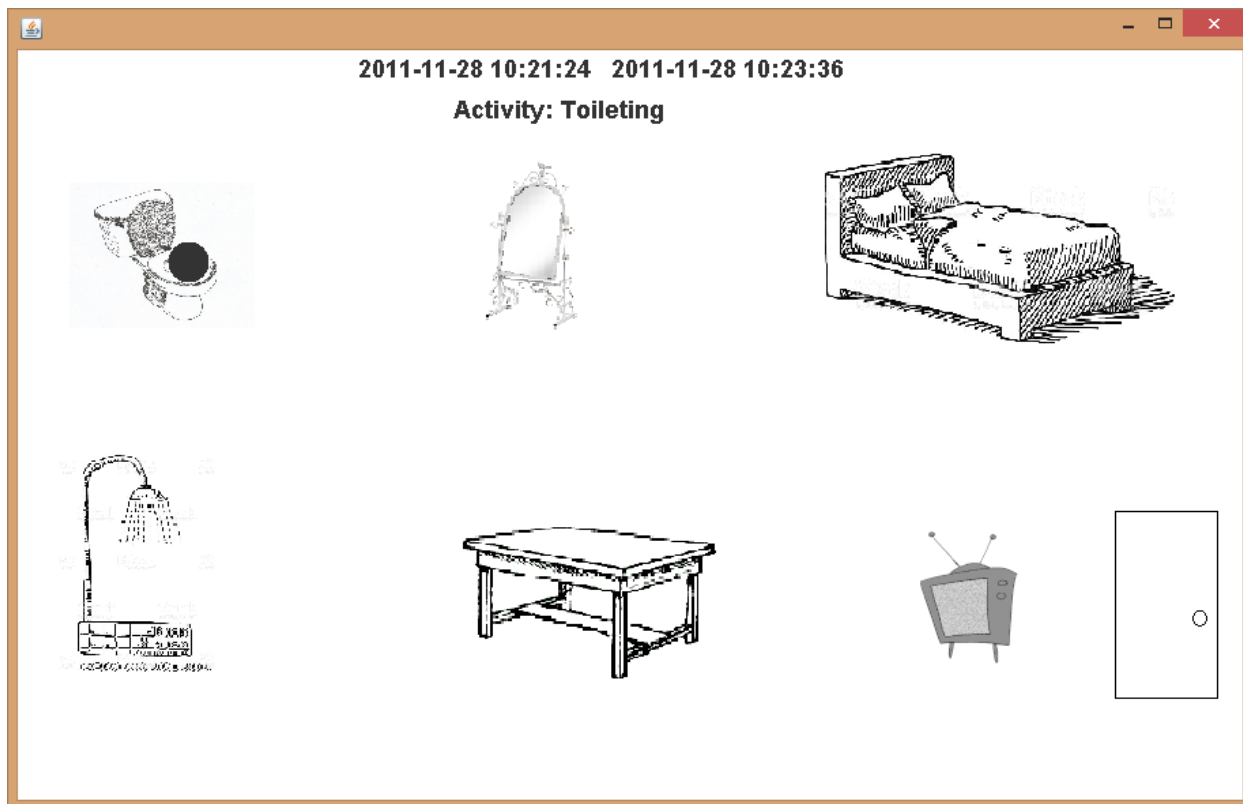
```
public Double getTimePassed(String startTime, String endTime) {
```

Este metoda care calculeaza durata pentru fiecare activitatea luand timpul de inceput si timpul de sfarsit si returneaza o variabile double.

Clasa CheckFilter are doar o functie Predicate care "selecteaza" valorile distincte la filtrare. In timpul de stream primeste parametri si il pune in HashMap si daca vine aceleasi parametru returneaza false in caz contrar true.

6. Concluzii si Dezvoltari Ulterioare

Am facut si simulare pentru activitati la o interfata grafica care ia timpuri durate si parcurge prin lista cu datele si arata ce activitatea face omul:



Punctul negru este omul. De exemplu acum este la toaleta sus arata cand a intrat si cand o sa iese.

7. Bibliografie

www.google.com

<https://stackoverflow.com/>

<https://www.youtube.com/>