

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE**

Tema 5

Tehnici de programare

Ardeleanu Madalin-Florin

An academic : 2018 – 2019



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE**

1. Enuntul temei

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.

Task

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: start_time, end_time and activity label and create a list of objects of type MonitoredData.

Count how many days of monitored data appears in the log.

Count how many times has appeared each activity over the entire monitoring period.

Return a map of type representing the mapping of activities to their count.

Count how many times has appeared each activity for each day over the monitoring period.

For each line from the file map for the activity label the duration recorded on that line (end_time-start_time).

For each activity compute the entire duration over the monitoring period.

Filter the activities that have 90% of the monitoring records with duration less than 5 minutes .

Documentation.

2. Obiectivul temei

Invatarea de Java 8, folosirea stream-urilor si a expresiilor lambda si familiarizarea cu acestea.

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE
3. Analiza problemei, asumptii, modelare, scenarii, cazuri de utilizare, erori

Acest program nu are anumite cazuri speciale, singura conditie pentru ca acest program sa functioneze o reprezinta existenta fisierului “Activities.txt”.

In acest fisier sunt trecute mai multe activitati, fiecare pe cate un rand.

Activitatiile sunt caracterizate de o data de inceput, o data de terminare si numele activitatii, fiecare data avand urmatorul format: “yyyy-MM-dd hh:mm:ss”

3.1. Asumptii

Programul are nevoie de un fisier text suport din care sa citeasca datele citite de senzori, in acest program, “Activities.txt”.

Programul nu functioneaza fara acest fisier.

4. Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator, modul de tratare a erorilor) si implementare

ReadFile
+ format : DateFormat + formatWithoutTime: DateFormat +counterDays(ArrayList<MonitoredData> list) +getActivity(ArrayList<MonitoredData> list) +getActivitySub5(ArrayList<MonitoredData> list) +getDurationActivity(ArrayList<MonitoredData> list) +p4(ArrayList<MonitoredData> list) +p7(HashMap<String, Integer> list) +readFromFile()



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

MonitoredData
-activityLabel: String -startTime : Date -endTime : Date -hoursDuration : long -minutesDuration: long -secondsDuration: long
+ getStartTime() + getEndTime() + getActivityLabel() + getDurationHours() + getDurationMinutes() + getDurationSeconds() + getDay() + setStartTime(Date a) + setEndTime(Date a) + setActivityLabel(String a) + setDurationHours(long a) + setDurationMinutes(long a) + setDurationSeconds(long a) + toString()

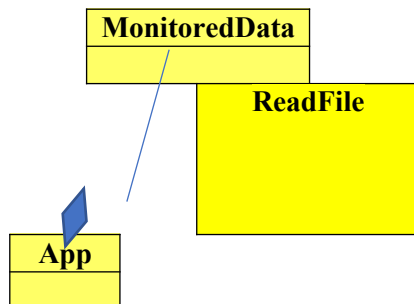


Fig 1. Diagrama Claselor



4.1. Proiectarea Claselor

Clasa ReadFile

Cu ajutorul acestei clase se executa toate cerintele problemei.

Cu ajutorul metodei **readFromFile** se realizeaza citirea din fisierul Activities.txt.

Este o metoda statica ce returneaza un ArrayList de obiecte MonitoredData, in aceasta metoda se realizeaza parcurgerea fisierului linie cu linie cu ajutorul unui stream, fiecare linie urmand sa fie descompusa in 5 string-uri, care vor fi salvate intr-un vector de string-uri dupa cum urmeaza: pe pozitia 0 se va afla data de inceput a activitatii, pe pozitia 2 se va afla data de sfarsit a activitatii iar pe pozitia 4 se va afla numele activitatii(iar pe pozitiile 1 respectiv 3) se vor salva taburile ce desparteau aceste informatii.

Dupa preluarea acestor date, se va crea un obiect de tipul MonitoredData si i se vor seta caracteristicile respective dupa care va fi adaugat in lista de obiecte MonitoredData ce va fi returnata.

Tot aici se calculeaza si durata fiecarei activitati ca diferenta intre timpul de terminare si cel de inceput.

Durata activitatiilor este la inceput in milisecunde, urmand sa fie convertita in secunde, minute si ore pentru a se putea actualiza toate detaliile cu referire la un obiect de tipul MonitoredData.

Urmeaza metoda **countersDays** ce primeste ca parametru un ArrayList de obiecte MonitoredData, aceasta metoda returneaza numarul de zile care au fost monitorizate in privinta rutinei unei persoane.

Aceasta se rezuma la parcurgerea ArrayList-ului de elemente MonitoredData cu ajutorul unui stream si formarea unui alt ArrayList de string-uri in care vor salva toate datele de inceput si de sfarsit a activitatiilor.

Pasul final se rezumat la numararea elementelor(in cazul nostru zilelor) distincti din ArrayList-ul nou format.

O alta metoda este **getActivity** ce va returna un HashMap cu denumirea fiecarei activitati si de cate ori apare aceasta pe intreaga durata de monitorizare.

Parametrul de intrare este ArrayList-ul cu elemente de tip MonitoredData.

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE**

Se parcurg elementele acestuia cu ajutorul unui stream, pentru fiecare activitate, se salveaza numele acesteia si i se cauta valoarea corespunzatoare din HashMap ce va urma sa o returnam. Initial valoarea va fii null, adica nici nu am inceput sa formam HashMap-ul, daca are valoarea null, introducem activitatea in HashMap cu valoarea 1 (adica dam pentru prima data peste activitatea asta), daca valoarea nu este null, inseamna ca nu e prima data cand dam peste aceasta activitate si nu trebuie decat sa ii modificam numarul de aparitii, adica valoare(valoare++).

Returnam HashMap-ul format.

Urmeaza metoda **getActivitySub5**, aceasta metoda este identica cu cea de sus, doar ca vrem sa formam un HashMap in care sa vedem de cate ori apare fiecare activitate cu conditia ca aceasta sa nu dureze mai mult de 5 minute.

Metoda **getDurationActivity** calculeaza durata totala a fiecarei activitati in parte pe intreaga durata de monitorizare, primeste ca parametru de intrare liste de obiecte de tip MonitoredData si va returna un HashMap de elemente ce va contine denumirea activitatii si durata acesteia in secunde.

Ideea de baza este aceeaasi, iteram peste lista de activitati, daca nu exista deja in HashMap o introducem cu durata acesteia, daca exista, modificam durata totala a activitatii ca fiind durata de dinainte + durata ultimei activitati gasite cu aceeaasi denumire.

Ultimele doua metode sunt **p4**, respectiv **p7** ce corespund task-urilor cu numerele 4 si 7.

In metoda **p4** se realizeaza numararea fiecarei activitati pe fiecare zi din intreaga perioada de monitorizare.

Parametrul de intrare este un ArrayList de obiecte de tipul MonitoredData iar parametrul de iesire este un HashMap care are cheia un Integer ce corespunde zilei respective din calendar si valoare un alt HashMap ce va contine la randul lui toate activitatiile care au loc in acea zi si numarul de aparitii ale acestora.

Toate detaliile acestea se scriu intr-un fisier f4.txt.



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE**

În metoda **p7** se realizează pur și simplu o scriere în fișierul **f7.txt**.

Aici se scriu activitățile care pe întreaga perioadă de monitorizare, în cel puțin 90% din cazuri, acestea au durat mai puțin de 5 minute.

Aceste activități sunt salvate în HashMap-ul care este trimis ca parametru și afișat, obținerea acestuia are loc în metoda static **main** unde se afișează toate rezultatele.

Modul de rezolvare:

La pașurile anterioare s-au format 2 HashMap, primul ce conține activitățile și numărul de apariție a fiecărei activități în parte pe întreaga durată a monitorizării și al doilea hashMap și conține activitățile și numărul de apariție a fiecărei activități care durează mai puțin de 5 minute pe întreaga durată de monitorizare astfel problema se rezolvă ușor.

Dacă 90% din valoarea aflată în primul HashMap ce corespunde numărului de activități respective este mai mică sau egală cu valoarea aflată în cel de-al doilea HashMap înseamnă că am găsit o activitate care în 90% din cazuri pe durata întreaga de monitorizare durează mai puțin de 5 minute.

Creăm un nou HashMap în care salvăm toate aceste activități și îl trimitem ca parametrul metodei **p7**.

Clasa MonitoredData

În această clasă definim caracteristicile fiecărui obiect de tip **MonitoredData**, ca atribute avem un timp de început al activității, un timp de sfârșit al activității, numele activității, și câte ore, minute și secunde durează aceasta.

Tot aici găsim câte un getter și setter pentru fiecare atribut și 2 constructori, unul cu și altul fără parametrii.

Se mai găsește și metoda **toString** în care afișăm fiecare activitate monitorizată sub forma: timp de început + timp de sfârșit + activitate.

Și nu în ultimul rând mai avem o metodă **getDay** ce returnează ziua respectivă dintr-o dată.

De exemplu dacă avem data 2010-11-11, va returna 11.

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

5. Rezultate

Nr. days of monitored data appears in the log:

14

Nr. of each activity over the entire monitoring period.

Breakfast : 14

Toileting : 44

Grooming : 51

Sleeping : 14

Leaving : 14

Spare_Time/TV : 77

Showering : 14

Snack : 11

Lunch : 9

Nr. of each activity for each day over the monitoring period: -> f4.txt

Duration recorded on each line:

Breakfast- Hours: 2, minutes: 58, seconds: 8

Toileting- Hours: 2, minutes: 20, seconds: 34

Grooming- Hours: 2, minutes: 40, seconds: 42

Sleeping- Hours: 131, minutes: 3, seconds: 31

Leaving- Hours: 27, minutes: 44, seconds: 44

Spare_Time/TV- Hours: 142, minutes: 28, seconds: 55

Showering- Hours: 1, minutes: 34, seconds: 9

Snack- Hours: 0, minutes: 6, seconds: 1

Lunch- Hours: 5, minutes: 13, seconds: 31

Activities that have 90% of the monitoring records with duration less than 5 minutes: -> f7.txt

BONUS: Activities with duration less than 5 minutes

Breakfast : 1

Toileting : 36

f4 - Notepad

File Edit Format View Help

```
1 {Breakfast=1, Toileting=2, Grooming=3, Sleeping=1, Leaving=1, Spare_Time/TV=6, Showering=1, Lunch=1}
2 {Breakfast=1, Toileting=3, Grooming=4, Sleeping=1, Spare_Time/TV=7, Snack=1, Showering=1, Lunch=1}
3 {Breakfast=1, Grooming=3, Toileting=2, Sleeping=1, Leaving=1, Spare_Time/TV=4, Showering=1}
4 {Breakfast=1, Grooming=2, Toileting=4, Sleeping=1, Leaving=1, Spare_Time/TV=6, Snack=2, Showering=1}
5 {Breakfast=1, Toileting=5, Grooming=6, Sleeping=1, Leaving=2, Spare_Time/TV=7, Snack=1, Showering=1, Lunch=1}
6 {Breakfast=1, Grooming=4, Toileting=3, Sleeping=1, Spare_Time/TV=5, Snack=1, Showering=1, Lunch=1}
7 {Breakfast=1, Toileting=6, Grooming=5, Sleeping=1, Leaving=1, Spare_Time/TV=8, Snack=2, Showering=1, Lunch=1}
8 {Breakfast=1, Toileting=1, Grooming=5, Sleeping=1, Leaving=1, Spare_Time/TV=4, Showering=1}
9 {Breakfast=1, Toileting=2, Grooming=5, Sleeping=1, Leaving=2, Spare_Time/TV=6, Showering=1}
10 {Breakfast=1, Toileting=1, Grooming=4, Sleeping=1, Leaving=2, Spare_Time/TV=3, Showering=1}
11 {Breakfast=1, Toileting=2, Grooming=3, Sleeping=1, Spare_Time/TV=3, Showering=1, Lunch=1}
28 {Breakfast=1, Grooming=2, Toileting=3, Sleeping=1, Leaving=1, Spare_Time/TV=4, Snack=1, Showering=1, Lunch=1}
29 {Breakfast=1, Grooming=3, Toileting=4, Sleeping=1, Leaving=1, Spare_Time/TV=6, Snack=1, Showering=1, Lunch=1}
30 {Breakfast=1, Grooming=2, Toileting=6, Sleeping=1, Leaving=1, Spare_Time/TV=8, Snack=2, Showering=1, Lunch=1}
```


MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

f7 - Notepad

— □ ×

File Edit Format View Help

Snack : 8