



Facultatea de Automatica și Calculatoare

Tema 5:
Expresii lambda și
procesarea stream-urilor

Disciplina: Tehnici de programare

Student:

Coman Vasile

An II

Grupa 30221

Profesor coordonator:

Antal Marcel

Profesor curs:

Ioan Salomie



Cuprins

1. Obiectivul temei.....	3
2. Analiza problemei, asumptii, modelare, scenarii, cazuri de utilizare, erori.....	3
3. Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator, modul de tratare a erorilor).....	3
4. Implementare.....	5
5. Rezultate.....	7
6. Concluzii.....	9
7. Bibliografie.....	9



1. Obiectivul temei

Obiectivul principal al temei este de efectuare a operațiilor asupra unor informații ce cuprind analiza comportamentului unui individ care efectuează diferite activități pe parcursul mai multor zile, informații stocate într-un fișier text prin nume activitate și data de început și de final a activității, utilizând expresii lambda și procesarea streamurilor.

Cerința problemei ne pretinde să rezolvăm diferite task-uri pe baza informațiilor date.

Obiective secundare:

- definirea unei clase care conține 3 câmpuri (startTime, endTime și activity) pentru a putea citi din fișier linie cu linie fiecare activitate;

- pe baza cerințelor impuse de problema utilizând expresii lambda se va rezolva fiecare cerință.

2. Analiza problemei, asumptii, modelare, scenarii, cazuri de utilizare, erori

Această aplicație permite utilizatorului să efectueze următoarele operații:

- va putea citi din fișier fiecare activitate cu asumptia că fiecare linie are formatul corect fiind compusă din 2 date de început și de final și de numele activității fiind despartite prin 2 taburi;

- va putea să determine pe câte zile se desfășoară analiza individului;

- va putea determina de câte ori apare fiecare activitate;

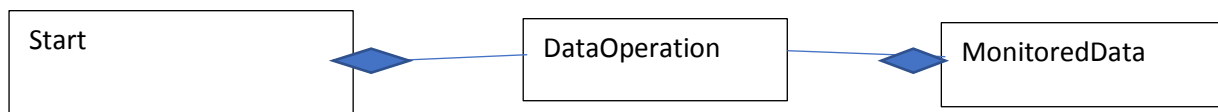
- va putea determina de câte ori apare fiecare activitate în fiecare zi din perioada analizei;

- va putea determina durata fiecărei activități;

- va putea determina durata fiecărei activități pe tot parcursul perioadei de analiză;

- va putea determina care sunt activitățile dintre care 90% au durată mai puțin de 5 minute.

3. Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfețe, relații, package-uri, algoritmi, interfața utilizator, modul de tratare a erorilor)





Am ales sa impart aplicatia in 3 clase:

- MonitoredData pt a descrie campurile ce sunt prezente in fiecare linie din fisier;
- DataOperation pt a implementa operatiile pe lista de obiecte de tip MonitoredData;
- Start care contine programul principal care apeleaza operatiile.

O funcție lambda (funcție anonimă) este o funcție definită și apelată fără a fi legată de un identificator. Funcțiile lambda sunt o formă de funcții „incuibate” (nested functions) în sensul că permit accesul la variabilele din domeniul funcției în care sunt conținute.

Funcțiile anonime au fost introduse de către Alonzo Church în anul 1936, în teoria sa despre calculele lambda. În limbajele de programare, funcțiile anonime sunt implementate din anul 1958 ca parte a limbajului Lisp. În unele limbajele orientate pe obiect, precum Java, apar concepte similare, precum clasele anonime. Abia în versiunea 8 a limbajului Java sunt adăugate și funcțiile anonime. Alte limbaje, precum C#, JavaScript, Perl, Python, Ruby ofereau demult suport pentru acest concept.

Lambda expresiile ne permit să creăm instanțe ale claselor cu o singură metodă într-un mod mult mai compact.

O lambda expresie constă:

dintr-o listă de parametri formali, separați prin virgulă și cuprinși eventual între paranteze rotunde,

săgeata direcțională ->,

un body ce constă dintr-o expresie sau un bloc de instrucțiuni.

O interfață funcțională (functional interface) este orice interfață ce conține doar o metodă abstractă. Din această cauză putem omite numele metodei atunci când implementăm interfața și putem elimina folosirea claselor anonime. În locul lor vom avea lambda expresii. O interfață funcțională este anotată cu `@FunctionalInterface`.



4. Implementare

Start	MonitoredData	DataOperation
private String startTime private String endTime private String activity		private String fileName2 private BufferedWriter writer private List<MonitoredData> activities
Public MonitoredData() public String getActivity() public void setActivity() public void setEndTime() public void setStartTime() public String getEndTime() public String getStartTime() public String toString()	Public static void main()	public DataOperation() public void readData() public void countDays() public void countActivities() public void countActivitiesForEachDay() public void durationOfEachActivityForEachLine() public void entireDurationOfEachActivity() public void activitiesLessThanFiveMinutes()

Clasa MonitoredData contine 3 campuri de tip String startTime, endTime si activity si vor defini o linie din fisier, fiecare activitate avand un nume si o data de inceput si de final.

De asemenea aceasta clasa contine gettere si settere pentru fiecare dintre cele 3 campuri. Tot aceasta clasa contine suprascrierea metodei toString() deoarece o vom folosi la afisare.

Clasa DataOperation implementeaza operatiile pe care utilizatorul le poate efectua asupra date de analiza din fisier. Contine 3 campuri un String fileName2 care reprezinta calea fisierului in care vom scrie rezultatele operatiilor. Un obiect de tip BufferedWriter care reprezinta bufferul de scriere in fisier si un ArrayList de obiecte de tip MonitoredData care reprezinta activitatile din fisier fiecare linie fiind o activitate.

Constructorul acestei clase initializeaza bufferul de scriere si ii transmite fisierul in care se vor scrie rezultatele. In metoda readData() un Stream de stringuri va lua fiecare linie din fisierul dat si apoi va transmite liniile prin metoda toList unei liste de Stringuri. Fiecare linie va fi cate un string. Acestui string i se va apela metoda split dupa 2 taburi creandu-se un obiect de tip MonitoredData dupa valorile care au fost despartite si formandu-se o lista de obiecte de tip MonitoredData. Apoi se va scrie in fisier aceasta lista.

Metoda countDays() apeleaza stream() pe lista de obiecte de tip MonitoredData numita activities si parsea startTime din String intr-un obiect de tip date(an-luna-zi) numarand datele distincte apeland count() si distinct() si returnand cate zile sunt apoi acestea sunt scrise in fisierul de rezultate si afisate in consola.

Metoda countActivities contine un HashMap care are cheia un String si valoarea un obiect de tip long. Intr-o lista de stringuri vom salva fiecare activitate distincta. Iar apoi pentru fiecare activitate vom parcurge din nou lista de activitati si vom numara de cate ori apare. Apoi vom scrie in fisier si vom afisa in consola.

In metoda countActivitiesForEachDay() salvam intr-o variabila de tip array de date fiecare data distincta prin parsarea campului startTime din activitati. Apoi pentru fiecare zi din acel array salvam activitatile intr-o lista de obiecte de tip MonitoredData iar apoi pentru acea lista



de activitati numaram de cate ori apare si afisam in consola si scriem in fisier de cate ori a aparut fiecare activitate pentru fiecare zi.

Metoda `durationOfEachActivityForEachLine()` parseaza fiecare `startTime` si `dateTime` dupa modelul(an-luna-zi ora:minute:secunde) si folosind `Duration.between` obtinem diferenta dintre cele 2 obiecte de tip `date` iar apoi scriem in fisier si afisam in consola obiectul de tip `Duration` apeland metoda `toString`.

Metoda `entireDurationOfEachActivity` salveaza intr-o lista de stringuri numele fiecarei activitati apoi pentru fiecare activitate se creeaza un array de obiecte de tip `Duration` si parcurgandu-se din nou toate activitatile se vor scrie in array duratele de timp a acelei activitatii apoi se va face suma acestor durate rezultand durata totala a activitatii si se va afisa in consola si se va scrie in fisier.

Metoda `activitiesLessThanFiveMinutes` intr-o lista de stringuri fiecare activitate si pentru fiecare activitate se filtreaza cele care au durata mai mica de 5 min si se salveaza intr-un `HashMap` cu cheia un string care reprezinta activitatea si valoarea care este un obiect de tip `long` si reprezinta de cate ori activitatea dureaza mai putin de 5 min. Apoi intr-o alta lista se salveaza activitatile care in cel putin 90% din cazuri dureaza mai putin de 5 minute si se afiseaza aceste activitati in consola.

Clasa `Start` care contine un obiect de tip `DataOperation` care apeleaza fiecare metoda.



5.Rezultate

La rularea aplicatiei sunt apelate toate metodele din clasa DataOperation si se afiseaza in consola si se scriu in fisier rezultatele.

Pentru metoda readData():

```
Rezultat - Notepad
File Edit Format View Help
Rezultate
Citirea si scrierea in fisier a activitatilor
Activitatea Sleeping incepe la 2011-11-28 02:27:59 si se termina la 2011-11-28 10:18:11
Activitatea Toileting incepe la 2011-11-28 10:21:24 si se termina la 2011-11-28 10:23:36
Activitatea Showering incepe la 2011-11-28 10:25:44 si se termina la 2011-11-28 10:33:00
Activitatea Breakfast incepe la 2011-11-28 10:34:23 si se termina la 2011-11-28 10:43:00
Activitatea Grooming incepe la 2011-11-28 10:49:48 si se termina la 2011-11-28 10:51:13
Activitatea Spare_Time/TV incepe la 2011-11-28 10:51:41 si se termina la 2011-11-28 13:05:07
Activitatea Toileting incepe la 2011-11-28 13:06:04 si se termina la 2011-11-28 13:06:31
Activitatea Leaving incepe la 2011-11-28 13:09:31 si se termina la 2011-11-28 13:29:09
Activitatea Spare_Time/TV incepe la 2011-11-28 13:38:40 si se termina la 2011-11-28 14:21:40
Activitatea Toileting incepe la 2011-11-28 14:22:38 si se termina la 2011-11-28 14:27:07
Activitatea Lunch incepe la 2011-11-28 14:27:11 si se termina la 2011-11-28 15:04:00
Activitatea Grooming incepe la 2011-11-28 15:04:59 si se termina la 2011-11-28 15:06:29
Activitatea Spare_Time/TV incepe la 2011-11-28 15:07:01 si se termina la 2011-11-28 20:20:00
Activitatea Snack incepe la 2011-11-28 20:20:55 si se termina la 2011-11-28 20:20:59
Activitatea Spare_Time/TV incepe la 2011-11-28 20:21:15 si se termina la 2011-11-29 02:06:00
Activitatea Sleeping incepe la 2011-11-29 02:16:00 si se termina la 2011-11-29 11:31:00
Activitatea Toileting incepe la 2011-11-29 11:31:55 si se termina la 2011-11-29 11:36:55
Activitatea Grooming incepe la 2011-11-29 11:37:38 si se termina la 2011-11-29 11:48:54
Activitatea Showering incepe la 2011-11-29 11:49:57 si se termina la 2011-11-29 11:51:13
Activitatea Breakfast incepe la 2011-11-29 12:08:28 si se termina la 2011-11-29 12:18:00
Activitatea Grooming incepe la 2011-11-29 12:19:01 si se termina la 2011-11-29 12:22:00
Activitatea Spare_Time/TV incepe la 2011-11-29 12:22:38 si se termina la 2011-11-29 12:24:59
Activitatea Snack incepe la 2011-11-29 13:25:29 si se termina la 2011-11-29 13:25:32
Activitatea Spare_Time/TV incepe la 2011-11-29 13:25:38 si se termina la 2011-11-29 15:12:26
Activitatea Toileting incepe la 2011-11-29 15:13:28 si se termina la 2011-11-29 15:13:57
Activitatea Lunch incepe la 2011-11-29 15:14:33 si se termina la 2011-11-29 15:45:54
Activitatea Grooming incepe la 2011-11-29 15:49:51 si se termina la 2011-11-29 15:50:54
Activitatea Spare_Time/TV incepe la 2011-11-29 15:52:04 si se termina la 2011-11-29 16:17:58
Activitatea Toileting incepe la 2011-11-29 16:18:00 si se termina la 2011-11-29 16:31:27
Activitatea Spare_Time/TV incepe la 2011-11-29 16:34:17 si se termina la 2011-11-29 17:08:07
Activitatea Toileting incepe la 2011-11-29 17:08:58 si se termina la 2011-11-29 17:09:29
Activitatea Spare_Time/TV incepe la 2011-11-29 17:43:00 si se termina la 2011-11-29 18:57:22
Activitatea Leaving incepe la 2011-11-29 19:02:15 si se termina la 2011-11-29 20:23:38
Activitatea Spare_Time/TV incepe la 2011-11-29 20:28:00 si se termina la 2011-11-30 01:19:47
Activitatea Sleeping incepe la 2011-11-30 01:22:33 si se termina la 2011-11-30 10:07:31
Activitatea Toileting incepe la 2011-11-30 10:11:07 si se termina la 2011-11-30 10:13:59
Activitatea Showering incepe la 2011-11-30 10:14:45 si se termina la 2011-11-30 10:18:33
Activitatea Breakfast incepe la 2011-11-30 10:22:59 si se termina la 2011-11-30 10:35:00
```

Pentru metoda countDays()

```
Activitatea Spare_Time/TV incepe la 2011-12-11 15:43:51 si se
Numarul de zile in care a avut loc monitorizarea este: 14
Activitatea Leaving apare de 14 ori
```

Pentru metoda countActivities()

```
Activitatea Leaving apare de 14 ori
Activitatea Breakfast apare de 14 ori
Activitatea Sleeping apare de 14 ori
Activitatea Grooming apare de 51 ori
Activitatea Snack apare de 11 ori
Activitatea Showering apare de 14 ori
Activitatea Spare_Time/TV apare de 77 ori
Activitatea Toileting apare de 44 ori
Activitatea Lunch apare de 9 ori
```

Pentru metoda countActivitiesForEachDay()



Activitatea Toileting apare de 1 ori
 Activitatea Lunch apare de 9 ori
 Activitatea Sleeping in ziua de 28 apare de 1 ori
 Activitatea Toileting in ziua de 28 apare de 3 ori
 Activitatea Showering in ziua de 28 apare de 1 ori
 Activitatea Breakfast in ziua de 28 apare de 1 ori
 Activitatea Grooming in ziua de 28 apare de 2 ori
 Activitatea Spare_Time/TV in ziua de 28 apare de 4 ori
 Activitatea Leaving in ziua de 28 apare de 1 ori
 Activitatea Lunch in ziua de 28 apare de 1 ori
 Activitatea Snack in ziua de 28 apare de 1 ori
 Activitatea Spare_Time/TV in ziua de 29 apare de 7 ori
 Activitatea Sleeping in ziua de 29 apare de 1 ori
 Activitatea Toileting in ziua de 29 apare de 4 ori
 Activitatea Grooming in ziua de 29 apare de 3 ori
 Activitatea Showering in ziua de 29 apare de 1 ori
 Activitatea Breakfast in ziua de 29 apare de 1 ori
 Activitatea Snack in ziua de 29 apare de 1 ori
 Activitatea Lunch in ziua de 29 apare de 1 ori
 Activitatea Leaving in ziua de 29 apare de 1 ori
 Activitatea Spare_Time/TV in ziua de 30 apare de 9 ori
 Activitatea Sleeping in ziua de 30 apare de 1 ori
 Activitatea Toileting in ziua de 30 apare de 6 ori
 Activitatea Showering in ziua de 30 apare de 1 ori
 Activitatea Breakfast in ziua de 30 apare de 1 ori
 Activitatea Grooming in ziua de 30 apare de 2 ori
 Activitatea Snack in ziua de 30 apare de 2 ori
 Activitatea Lunch in ziua de 30 apare de 1 ori
 Activitatea Leaving in ziua de 30 apare de 1 ori
 Activitatea Spare_Time/TV in ziua de 1 apare de 7 ori
 Activitatea Sleeping in ziua de 1 apare de 1 ori
 Activitatea Grooming in ziua de 1 apare de 3 ori
 Activitatea Showering in ziua de 1 apare de 1 ori
 Activitatea Breakfast in ziua de 1 apare de 1 ori
 Activitatea Lunch in ziua de 1 apare de 1 ori
 Activitatea Toileting in ziua de 1 apare de 2 ori

Pentru metoda durationOfEachActivityForEachLine()

Activitatea Sleeping care incepe la 2011-11-28 02:27:59 si se termina la 2011-11-28 10:18:11 dureaza PT7H50M12S
 Activitatea Toileting care incepe la 2011-11-28 10:21:24 si se termina la 2011-11-28 10:23:36 dureaza PT2M12S
 Activitatea Showering care incepe la 2011-11-28 10:25:44 si se termina la 2011-11-28 10:33:00 dureaza PT7M16S
 Activitatea Breakfast care incepe la 2011-11-28 10:34:23 si se termina la 2011-11-28 10:43:00 dureaza PT8M37S
 Activitatea Grooming care incepe la 2011-11-28 10:49:48 si se termina la 2011-11-28 10:51:13 dureaza PT1M25S
 Activitatea Spare_Time/TV care incepe la 2011-11-28 10:51:41 si se termina la 2011-11-28 13:05:07 dureaza PT2H13M26S
 Activitatea Toileting care incepe la 2011-11-28 13:06:04 si se termina la 2011-11-28 13:06:31 dureaza PT27S
 Activitatea Leaving care incepe la 2011-11-28 13:09:31 si se termina la 2011-11-28 13:29:09 dureaza PT19M38S
 Activitatea Spare_Time/TV care incepe la 2011-11-28 13:38:40 si se termina la 2011-11-28 14:21:40 dureaza PT43M
 Activitatea Toileting care incepe la 2011-11-28 14:22:38 si se termina la 2011-11-28 14:27:07 dureaza PT4M29S
 Activitatea Lunch care incepe la 2011-11-28 14:27:11 si se termina la 2011-11-28 15:04:00 dureaza PT36M49S
 Activitatea Grooming care incepe la 2011-11-28 15:04:59 si se termina la 2011-11-28 15:06:29 dureaza PT1M30S
 Activitatea Spare_Time/TV care incepe la 2011-11-28 15:07:01 si se termina la 2011-11-28 20:20:00 dureaza PT5H12M59S
 Activitatea Snack care incepe la 2011-11-28 20:20:55 si se termina la 2011-11-28 20:20:59 dureaza PT4S
 Activitatea Spare_Time/TV care incepe la 2011-11-28 20:21:15 si se termina la 2011-11-29 02:06:00 dureaza PT5H44M45S
 Activitatea Sleeping care incepe la 2011-11-29 02:16:00 si se termina la 2011-11-29 11:31:00 dureaza PT9H15M
 Activitatea Toileting care incepe la 2011-11-29 11:31:55 si se termina la 2011-11-29 11:36:55 dureaza PT5M
 Activitatea Grooming care incepe la 2011-11-29 11:37:38 si se termina la 2011-11-29 11:48:54 dureaza PT11M16S
 Activitatea Showering care incepe la 2011-11-29 11:49:57 si se termina la 2011-11-29 11:51:13 dureaza PT1M16S
 Activitatea Breakfast care incepe la 2011-11-29 12:08:28 si se termina la 2011-11-29 12:18:00 dureaza PT9M32S
 Activitatea Grooming care incepe la 2011-11-29 12:19:01 si se termina la 2011-11-29 12:22:00 dureaza PT2M59S
 Activitatea Spare_Time/TV care incepe la 2011-11-29 12:22:38 si se termina la 2011-11-29 12:24:59 dureaza PT2M21S
 Activitatea Snack care incepe la 2011-11-29 13:25:29 si se termina la 2011-11-29 13:25:32 dureaza PT3S
 Activitatea Spare_Time/TV care incepe la 2011-11-29 13:25:38 si se termina la 2011-11-29 15:12:26 dureaza PT1H46M48S
 Activitatea Toileting care incepe la 2011-11-29 15:13:28 si se termina la 2011-11-29 15:13:57 dureaza PT29S
 Activitatea Lunch care incepe la 2011-11-29 15:14:33 si se termina la 2011-11-29 15:45:54 dureaza PT31M21S
 Activitatea Grooming care incepe la 2011-11-29 15:49:51 si se termina la 2011-11-29 15:50:54 dureaza PT1M3S
 Activitatea Spare_Time/TV care incepe la 2011-11-29 15:52:04 si se termina la 2011-11-29 16:17:58 dureaza PT25M54S
 Activitatea Toileting care incepe la 2011-11-29 16:18:00 si se termina la 2011-11-29 16:31:27 dureaza PT13M27S
 Activitatea Spare_Time/TV care incepe la 2011-11-29 16:34:17 si se termina la 2011-11-29 17:08:07 dureaza PT33M50S
 Activitatea Toileting care incepe la 2011-11-29 17:08:58 si se termina la 2011-11-29 17:09:29 dureaza PT31S
 Activitatea Spare_Time/TV care incepe la 2011-11-29 17:43:00 si se termina la 2011-11-29 18:57:22 dureaza PT1H14M22S
 Activitatea Leaving care incepe la 2011-11-29 19:02:15 si se termina la 2011-11-29 20:23:38 dureaza PT1H21M23S
 Activitatea Spare_Time/TV care incepe la 2011-11-29 20:28:00 si se termina la 2011-11-30 01:19:47 dureaza PT4H51M47S
 Activitatea Sleeping care incepe la 2011-11-30 01:22:33 si se termina la 2011-11-30 10:07:31 dureaza PT8H44M58S

Pentru metoda entireDurationForEachActivity()



```

Activitatea Grooming      care incepe la 2011-12-11 15:41:34 si se termina l
Activitatea Spare_Time/TV care incepe la 2011-12-11 15:43:51 si se termina
Durata totala a activitatii Sleeping este PT107H3M31S
Durata totala a activitatii Toileting este PT-9H-39M-26S
Durata totala a activitatii Showering este PT1H34M9S
Durata totala a activitatii Breakfast este PT14H58M8S
Durata totala a activitatii Grooming este PT2H40M42S
Durata totala a activitatii Spare_Time/TV este PT-565H-31M-5S
Durata totala a activitatii Leaving este PT39H44M44S
Durata totala a activitatii Lunch este PT5H13M31S
Durata totala a activitatii Snack este PT6M1S

```

Pentru metoda `activitiesLesssThanFiveMinutes()`

```

Durata totala a activitatii Snack este PT6M1S
Activitatile care au 90% din durata mai putin de 5 min sunt:
Snack

```

6. Concluzii

In concluzie aceasta aplicatie implementeaza corect si eficient operatiile de analiza a fisierului de activitati folosind streamurile si expresiile lambda.

Posibilitati de dezvoltare ulterioara:

- determinarea activitatii care dureaza cel mai mult;
- determinarea activitatii care dureaza cel mai putin;
- calculul duratei medii a fiecarei activitati;
- implementarea acestor operatii pe mai multe fisiere pentru analiza mai multor indivizi.

In urma acestui proiect am invatat si am inteles cum trebuie sa lucrez cu expresiile lambda si cu procesarea de streamuri.

7. Bibliografie

http://coned.utcluj.ro/~salomie/PT_Lic/

<https://stackoverflow.com/>

<https://docs.oracle.com/javase/tutorial/>

<http://www.coned.utcluj.ro/~marcel99/PT/>

<http://cursuri.cs.pub.ro/~poo/wiki/index.php/Input/Output>