Tema 5: Lambda Expressions

Student: Campean Bogdan Mihai

Grupa: 30221

**1. Obiectivul temei**

Obiectivul principal al acestei teme este de a utiliza expresiile lambda asupra unor date care se afla intr-un fisier. Aceste date reprezinta actiunile unui om de-a lungul a catorva zile care contine data de inceput, timpul de inceput si data de terminare si timpul de terminare al acestor actiuni, impreuna cu numele actiunii. Programul trebuie sa filtreze toate aceste actiuni produse de o persoana de-a lungul celor 14 zile si de a identifica unele cerinte care par mai suspecte. Acest exemplu de problema se poate utiliza, de exemplu, in cadrul unei personae care utilizeaza un card la momentul x intr-o anumita tara, iar peste un minut, mai exact la momentul x+1, acesta il utilizaza la mii de kilometri distanta, intr-o alta tara, fapt imposibil de realizat in viata reala, fiind un act suspect de frauda, iar banca ii blocheaza cardul.

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

Incepem analiza problemei prin evidentierea cuvintelor cheie utilizate in cerinta, si anume: lambda expressions, lists, hashmaps, etc.

In primul rand observam ca fiecare subpunct al cerintei trebuie realizat prin expresiile lambda pe care limbajul de programare orientat pe obiecte Java, versiunea 8, le are in dotare. Aceste functii trebuie utilizate pentru filtratrea rezultatelor, aplicarea unei functii asupra ei si de a repeta acest ciclu pana cand exista elemente in acea list ape care o iteram.

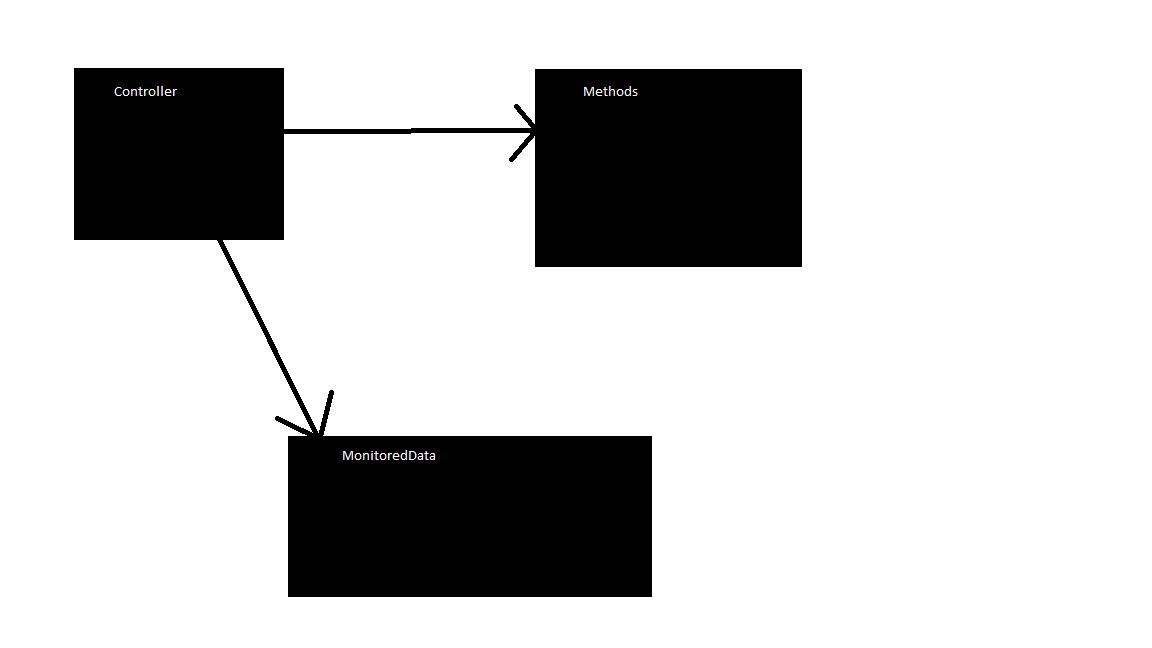
In al doilea rand observam al doilea termen al cerintei, lists, care se refera la tipul de date List care contine toate tipurile de liste: liste inlantuite, liste sub forma de vectori, etc. Aceste liste se folosesc pentru afisarea si salvarea rezultatelor rezultate dupa filtrarea si aplciarea functiilor lambda asupra lor.

In al treilea rand, observam ultimul termen care exista in cerinta, si anume Map. Map este o structura de date care inglobeaza toate Map-urile, cum ar fi: HashMap, HashSet, etc. Aceste Map-uri se folosesc la stocarea temporara a datelor in urmatorul fel: fiecare actiune va avea un numar de aparitii in fiecare zi sau un timp total in care s-a facut acea actiune, insa vom vedea detalii in urmatoarele capitole.

Cazurile de utilizare a aplicatiei sunt foarte putine deoarece cerintele sunt foarte restranse si stricte in cee ace priverte utilizarea aplicatiei, in sensul ca aplicatia are rolul de a obisnui studentul cu utilizarea expresiilor lambda si nu de a fi folosita de utilizatori fara experienta in programare orientate pe obiecte; de asemenea, aplciatia nu implementeaza o interfata grafica pentru usurarea utilizarii de utilizatori obisnuiti, ci doar afisarea in consola si in fisiere a rezultatelor obtinute dupa filtrarea si aplicarea functiilor lambda asupra acestora.

In cazul asumptiilor putem afirma ca nu exista, deoarece exista doar un caz de utilizare si anume acela implicit in care se apeleaza metodele cerute si specificate in cerinta si afisarea rezultatelor in consola sau intr-un fisier de tip text cu extensia .txt.

3. Proiectarea



Dupa cum se observa in diagram de calse UML, care este foarte utila in domeniul IT deoarece simplifica observarea relatiilor dintre clase de catre programatori, putem observa existent a doar 3 clase utilizate in aceasta aplicatie si anume: clasa Montored Data, clasa Controller si clasa Methods.

Aceste clase au fiecare utilitatea lor, si anume: clasa Monitored Data are rolul de a stoca toate informatiile obtinute din fisierul de actiuni, date si timpi; clasa Methods are rolul de a stoca metodele utilizate care filtreaza si aplica expresiile lambda asupra rezultatelor; si clasa Controller are rolul de a uni aceste 2 clase si de a asigura functionarea corecta si in conditii de siguranta a aplicatiei.

4. Implementare

In acest capitol vom prezenta cu lux de detalii amanuntite, inclusive detalii tehnice, construirea si utilizarea claselor si a programului, precum si a metodelor utilizate si rolul fiecareia.

In primul rand observam clasa Monitored Data, care este o clasa definitorie a actiunilor effectuate de o anumita persoana de-a lungul a 14 zile, deoarece aceasta clasa contine toate detaliile despre fiecare actiune intreprinsa de anumita persoana. Aceasta clasa contine campurile end Time, start Time, end Data, sart Data si activity sub forma de String-uri; de asemenea, mai contine si seteri si geteri pentru aceste campuri deoarece variabilele de clasa sunt de tipul private si nu se pot vedea in interiorul altor clase, inclusive in clasa Controller care contine metoda main, metoda ce porneste si asigura buna functionare in conditii de siguranta a aplicatiei.

Urmatoarea clasa pe care o prezentam este clasa Methods care, in mare, contine toate metodele care se vor apela pe parcursul programului pentru a obtine rezultate dorite si cerute, in mod corect. Aceasta clasa contine variabilele de clasa list – o lista de tipul String care contine rezulatele unei cerinte, o constanta de clasa de tipul String care contine calea spre fisierul de activitati ale acelei personae, Activity . txt, un contor de tipul int pentru utilizarea acestuia in interiorul unei functii lambda, si o variabila de clasa hour de tipul Integer pentru monitorizarea orelor petrecute pe fiecare actiune intreprinsa de persoana. De asemenea, aceasta calsa contine un constructor fara parametri in care lista List de String-uri este instantiate ca fiind o lista inlantuita de String-uri.

In cele ce urmeaza vom prezenta metodele folosite si implementate pentru obtinerea rezultatelor cerute in cerinta problemei si anume: metoda read From File, metoda fara parametric care returneaza o lista de tipul array list de tipul Monitored Data; aceasta metoda are rolul de a instantia o lista de tipul array list de tipul monitored data in care se vor obtine, rand pe rand, informatiile citite din fisierul de activitati ale persoanei respective; acest lucru este facut de 2 functii lambda dupa cum urmeaza: prima expresie lambda are rolul de a deschide fisierul dat de calea din constanta de clasa si de a memora toate liniile intr-o lista de tipul lista inlantuita de tipul String; aceasta expresie lambda contine functia collect care are rolul de memorare in list ape care o dam ca si parametru, astfel, dupa aceasta iteratie a expresiei lambda, in lista inlantuita de tipul String, file Line, avem toate randurile fisierului sub forma de String, nedespartite intre ele; a doua expresie lambda are rolul de a desparti in String-uri mai mici fiecare String mare memorat in lista inlantuita de String-uri mari. Aceasta expresie va itera toata lista file Line de String-uri mari si va desparti prin cateva while-uri toate randurile si le transforma in String-uri mai mici, care apoi vor fi transformate intr-un obiect de tipul Monitored Data, care are un constructor cu cei 5 parametri ai sai: end time, start time, nd data, start datas I activity. Acest obiect va fi returnat si bagat in lista cu ajutrul functiei .collect care are ca si parametru colectorul care este de tipul lista de tipul monitored data. A doua metoda utilizata in aceasta clasa este metoda ge distinct activities cu un singur parametru de tipul lista inlantuita de tipul monitored data, pe care am construit-o in metoda anterioara; aceasta metoda returneaza un obiect de tipul hash map de tipul string si integer care au rolul de a adauga intr-o lista de tipul hash map cu semnificatia ca de cate ori apare o actiune in totalul de actiuni effectuate de persoana respective. Integer-ul respective are rolul de contor care este incrementat la fiecare actiune comuna de-a lungul celor 14 zile din activitiati. Aceste activiati sunt filtrate printr-o expresie lambda, iar daca actiunea respective se afla printre hash map, atunci contorul se incrementeasza si se inlocuieste cu + 1. Urmatoarea metoda prezentata rezolva subpunctul 2 al probelemei, si anume contorizarea zilelor distincte dintre toate activitatile effectuate de catre anumita persona; aceasta metoda s-a bazat pe ideea ca exista 2 zile distincte daca si numai daca start date si end date sunt diferite cu o zi, adica diferenta dintre end date si start date sa fie maxim si doar 1. Aceasta metoda itereaza toata lista de monitored data, ia end data si start data, face difereta si verifica daca sunt distincte printr-o valoare 1; daca da, atunci aceasta se adauga la list ape care o construim; daca nu, aceasta nu trece de filtru si nu este adaugata la contor. Urmatoarea metoda pe care o rprezentam este metoda get distinct activities care are un singur parametru si anume o lista de tipul monitored data, cea pe care am luat-o la subpunctul 1, care returneaza o hash map de tipul string si o hash map de tipul string si integer, care are rolul de a salva pe zile fiecare actiune si de cate ori apare ea in activitatile persoanei respective. Aceasta metoda foloseste de asemenea expresii lambda pentru filtrarea fiecarei actiuni pe care persoana respective o face. Daca actiunea apartine unei zile distincte de celelalte, atunci se adauga la primul integer counter + 1 pentru incrementarea zilelor si pentru reinnoirea hash map-ului de care apartine fiecare zi, unde se memoreaza fiecare actiune si de cate ori apare aceasta in ziua respective. Urmatoarea metoda pe care o prezentatm sunt metodele care transforma o ora de tipul string sub formatul hh : mm : ss in tipul integer pentru fiecare dintre: ora, minut si secunda; de asemenea, s-au implementat si metodele care transforma o data de tipul y y y y : m m : d d in integer pt fiecare dintre acestea; aceste metode se vor folsi ulterior in alta metoda care are rolul de a calcula timpul total scurs de o activitate si filtrarea actiunilor ce au mai mult de 10 ore timpul. Urmatoarea metoda pe care o prezentam este metoda get activities care are un singur parametru si anume o lista de tipul array list de tipul moitored data si care are rolul de a aduna toti timpii efectuati de fiecare actiune si de a salva rezultatele intr-un hash map de tipul string si date time; in aceasta metoda se folosesc metodele de transformare a datei si timpului din string in integer pentru usurarea calculelor pe obiecte de tip integer, unde se vor construi obiecte de tipul date time si se vor inlocui cu cele vechi pe pozitia din hash map care contine string-ul respective. In aceasta metoda se vor afisa, de semenea toti timpii efectuati de o actiune pe fiecare rand, adica end time – start time, realizand astfel cat timp dureaza fiecare actiune la un moment dat intr-o zi. Urmatoarea metoda pe care o prezentam este metoda get filtered activities cu un singur parametru de tipul map de tipul string si date time, care returneaza un map de tipul string si date time si care are rolul, prin intermediul expresiilor lambda, de a filtra toate activitatile care dureaza mai mult de 10 ore si de a le afisa apoi in consola dar si intr-un fisier cu extensia . txt. Urmatoarea metoda pe care o prezentatm este metoda filter five minutes cu 2 parametri : unul de tipul lista inlantuita de tipul monitored data si celallt de tipul map de tipul String si Integer, care retine fiecare activitate si cate activitati sun in total de o singura

Metoda filterFiveMinutes are rolul de a filtra dintre toate actiunile de tipul MonitoredData doar pe cele care, 90% din actiuni, au timpul de derulare mai mic de 5 minute. Daca respecta aceasta conditie atunci, prin intermediul expresiilor lambda, se or filtra doar acele activitati care le respecta. Atunci ele vor fi colectate intr-o lista de String-uri doar numele lor, fara nimic altceva. De asemenea, aceasta metoda va scrie intr-un fisier text aceste actiuni filtrate.

5. Testare

6. Rezultate

7. Concluzii

In concluzie, putem afirma ca aceasta aplicatie este o aplicatie realizata in scop didactic pentru invatare si nu pentru utilizarea acesteia de catre utilizatori obisnuiti care nu au experienta in programare orientate pe obiect, deoarece nu se folosesc aplicatii de acest gen in viata reala, deci ar fi inutila cnstruirea unei aplicatii care afiseaza doar mesaje pentru filtrarea unor rezultate. Deci, aceasta applicatie a fost construita pentru a invata studentii cum sa utilizeze in mod corect expresiile lambda pe care limbajul de programare orientate pe obiect Java, versiunea 8, le are in dotare.