**Exercitiu FIS**

-

Laboratorul 1

Intr-un pseudo-sistem de stocare a informatiei se considera urmatoarele ierarhii de clase:

Clasa **Document** contine un atribut secventa de siruri de caractere (modelata fie ca tablou fie ca lista) corespunzator textului integral stocat in document impreuna cu o metoda corespunzatoare pentru afisarea acestui text. De asemenea, clasa contine o metoda analizaeaza care returneaza o noua secventa de siruri de caractere si a carei implementare depinde de tipul concret al documentului. Documentele pot fi de doua tipuri:

1. **XML**. In cadrul acestui tip de document informatia este reprezentata ca tag-uri, fiecare sir de caractere al secventei fiind fie un tag de inceput (ex. "<nume\_tag>"), fie o valoare sir de caractere, fie un tag de sfarsit (ex. "</nume\_tag>"). Metoda analizeaza returneaza o secventa de siruri de caractere obtinuta prin eliminarea tag-urilor (de inceput si de sfarsit). Metoda corespunzatoare pentru afisare va returna "XML" urmat de textul integral dintr-un document (incluzand tag-uri). Exemplu: <tag1> valoare1 </tag1> <tag2> </tag2> <tag3> valoare2 </tag3>. Metoda analizeaza va returna pentru acest exemplu secventa valoare1, valoare2. Se considera ca secventa de siruri de caractere data initial corespunde tot timpul formatului indicat.

2. **JSON**. In cadrul acestui tip de document elementele secventei sunt fie nume (ex. "nume") fie o valoare sir de caractere. Metoda analizeaza returneaza o secventa de siruri de caractere obtinuta prin eliminarea tuturor numelor din secventa initiala. Metoda corespunzatoare pentru afisare v-a retuna "JSON" urmata de textul integral dintr-un document (incluzand numele). Exemplu: JSON nume1: valoare1 nume2: valoare2. Dupa elementele nume exista un :, iar pentru exemplul dat metoda analizeaza intoarce valoare1, valoare2. Se considera ca secventa de siruri de caractere data initial corespunde tot timpul formatului indicat.

In cadrul sistemului orice document poate fi procesat utilizand un **Procesor**. Se considera urmatoarele tipuri de procesoare:

1. **ProcesorCautare**. Acest procesor contine un sir de caractere initializat prin constructor care corespunde textului ce urmeaza sa fie cautat. Metoda proceseaza primeste ca parametru o secventa de documente si functioneaza in felul urmator: pentru fiecare text obtinut prin analizarea unui document (deci fara tag-uri ori nume) se numara toate aparitiile sirului cautat in textul respectiv. Metoda returneaza numarul total de aparitii ale sirului cautat in toate documentele secventei de documente.

2. **ProcesorCompus**. Acest procesor contine o secventa de procesoare (atat de cautare cat si alte procesoare compuse). Metoda proceseaza primeste ca parametru o secventa de documente si functioneaza in felul urmator: returneaza suma rezultatelor obtinute prin aplicarea fiecarui procesor din cadrul secventei de procesoare pe documentele primite ca parametru de metoda proceseaza.

Unde e cazul, secventele se initializaeaza fie din constructor fie folosind o metoda de adaugare de elemente corspunzatoare.

**Cerinte:**

- sa se implementeze clasele descrise anterior impreuna cu alte clase considerate necesare folosind IntelliJ IDEA (obligatoriu); se considera urmatoarea impartire pe pachete: clasele ierarhiei de documente vor face parte din pachetul ***loose.oose.fis.documents,*** clasele ierarhiei de procesoare vor face parte din pachetul ***loose.oose.fis.processors***, iar clasa Main va apartine pachetului radacina (***loose.oose.fis***)

- sa se genereze in mod automat perechi de metode getter/setter pentru atributele clasei **Document** precum si metoda de afisare (toString)

- sa se demonstreze functionarea sistemului descris anterior printr-o clasa Main in care:

1) se instantiaza cel putin doua documente (unul din fiecare) pe baza textelor primite ca model;

2) se instantiaza trei procesoare de cautare, un procesor compus care contine primele doua procesoare de cautare si un procesor compus care contine procesorul compus precedent si al treilea procesor de cautare;

3) se apeleaza metoda proceseaza pe ultimul procesor compus cu o secventa de documente care contine obiectele instantiate la punctul 1

4) sa se intre cu debugger-ul in apelul metodei ***proceseaza*** de pe obiectul mentionat anterior

- sa se refactorizeze in mod automat numele atributului de tip sir de caractere din cadrul clasei **ProcesorCautare**