Grameni Stere-Alexandru 323CC

Grad de dificultate: 9/10

Tema a fost rezolvata in 10 zile, cu media de 4-5 ore pe zi

Implementare:

Radacina are numele home si este create la deschiderea aplicatiei. Utilizatorul root are parola rootpass.

Am creat o interfata Node(in enuntul temei Repository) si 3 clase care implementeaza interfata: Fisier, Director si MachineId. MachineId este folosit pentru serviciul de cloud. Atunci cand stochez un director in cloud, toti succesorii lui vor fi inlocuiti cu obiecte de tip MachineId, ce curinde id-ul statiei in care a fost stocat succesorul si numele succesorului.

Am creat o interfata Command care este implementata de 3 clase abstracte: ReadCommand, WriteCommand si SystemCommand, clase care sunt derivate de comenzi.

Clasa CommandFactory are o metoda getCommand care primeste un string si in functie de stringul primit, intoarce un obiect comanda.

Toate comenzile primesc directorul current, unele si directorul radacina.

Comanda Cat afiseaza un vector de 0 si 1, ce reprezinta datele din fisier.

Parametrii posibili: numele unui fisier.

Comanda cd imparte un sir ce caractere in noduri, delimitatorul fiind “/”.

Parametrii: calea catre un nod.

Comanda echo afiseaza un mesaj primit ca parametru.

Parametrii: un String si parametrul –POO(optional)

Comanda Login verifica numele si parola si, in caz favorabil, trimite update catre un observatory. La fel si comanda Logout.

Parametrii: login: username, parola

Logout: fara parametrii

Comanda Ls trateaza 2 cazuri: daca primeste ca argument un nod sau nu, care se impart in mai multe cazuri, in functie de parametrii pt recursivitate sau detalii.

Parametrii: numele unui director, -ar, -POO, toti parametrii sunt optionali

Comanda Mkdir creaza un director nou.

Parametrii: numele directorului, permisiunile(optional). Daca nu primeste permisiuni, toti utilizatorii vor avea drepturi asupre lui.

Comanda Touch creaza un fisier nou.

Parametrii: numele, dimensiunea(optional), permisiunile(optional)

Comanda newuser creaza un utilizator nou si il pune intr-o lista.

Parametrii: username, parola, nume si prenume.

Comanda pwd afiseaza calea absoluta a unui nod. Arunca si o exceptie, daca lungimea esta mai mare de 255 caractere.

Fara parametrii

Comanda Rm sterge un nod. Imparte un string in noduri, separatorul fiind “/” si il elimina din lista nodului predecesor.

Parametrii: -r pentru recursivitate(optional), numele

Comanda sync trateaza 2 cazuti: daca primeste parametru sau nu. In ambele cazuti este apelata metoda sync din clasa CloudService.

La fel si comanda upload.

Comanda userinfo afiseaza datele despre utilizatorul current. Nu primeste parametrii.

Am creat o interfata Subject care este implementata de clasele exceptii si login si logout. Acestea trimit update catre un observator in constructori.

Clasa CloudService contine 3 statii de stocare si implementeaza upload si sync. Comanda upload parcurge arborele si introduce nodurile intr-o lista, dupa care le stocheaza pe cloud. Sync ia nodurile din cloud si recreaza ierarhia.

Am o clasa User pentru utilizatori si o clasa Logger, implementata cu singleton pattern care tine evigenta userului logat.

Am creat doua fisiere txt pentru testare, similare. Primul creaza o ierarhie mai mica de 10kb, iar al doilea mai mare.