JavaZeroZahar: StockExchange

Membrii echipei:

Murariu Marius

Munteanu Mircea-Georgian

Munteanu Daniel

Vasile Ioana

Costean Ionut

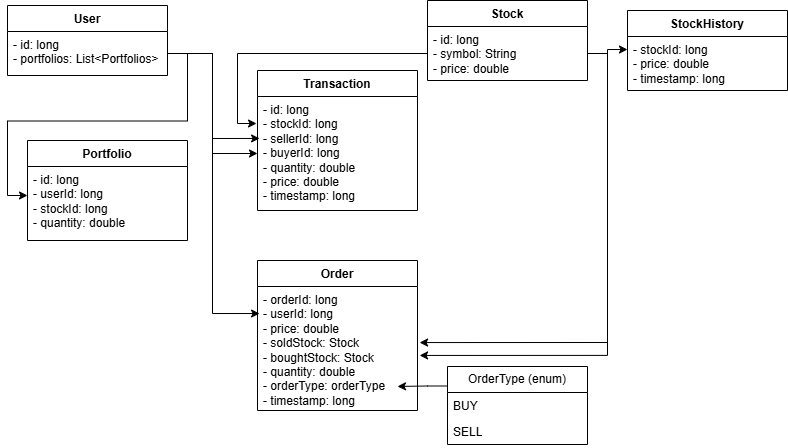
Link repository: <https://github.com/MirceaTT8/StockExchange>

1. **Descrierea proiectului**

Acest proiect reprezinta simularea unei burse de actiuni care functioneaza asemanator cu modul de functionare propus la laborator. Structura este de tip server-clienti, avand urmatoarele roluri:

* Server-ul are ca sarcini:
  + procesarea comenzilor (de cumparare si vanzare);
  + gestionarea actiunilor de tip CRUD (Create Read Update Delete) asupra anumitor obiecte ce vor fi stocate intr-o baza de date;
  + verificarea conditiilor necesare functionarii corecte din punct de vedere concurent (potentiale probleme explicate la punctul 4), dar si din alte puncta de vedere (de exemplu, verificarea ca un comparator are suficient de multi bani pentru a lansa o cerere de cumparare cu suma respectiva).
* Client-ul reprezinta platforma prin care utilizatorii pot interactiona cu Server-ul prin trimiterea de request-uri (folosind un API Http pentru Java), cu ajutorul unui framework pentru frontend (de exemplu Vue.js)

1. **Entitatile proiectului**

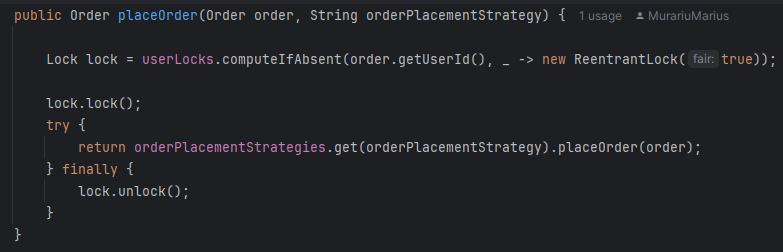
****

In aceasta diagrama sunt reprezentate clasele de obiecte care vor fi introduse intr-o baza de date si dependentele dintre ele (de exemplu, un portofoliu trebuie sa contina ca atribut valoarea id-ului user-ului care il detine).

1. **Posibile probleme de concurenta**

Printre cazurile care ar putea duce la conflicte de concurenta am identificat urmatoarele:

* Adaugarea mai multor comenzi simultan pentru un stoc poate duce la o problema de tipul producer-consumer;
* Cazul in care un user trimite simultan mai multe comenzi de modificare a unui order poate provoca problem atunci cand se verifica disponibilitatea fondurilor pentru vanzare/cumparare de stocuri;

****

Metoda din imagine apartine clasei OrderPlacer, care foloseste metode din biblioteca java.util.concurrent pentru a asigura faptul ca nu vor exista niciodata 2 sau mai multe comenzi care vor fi efectuate simultan de mai multe thread-uri.