Laborator 4

Stefan Sebastian 235

Analiza cerintelor

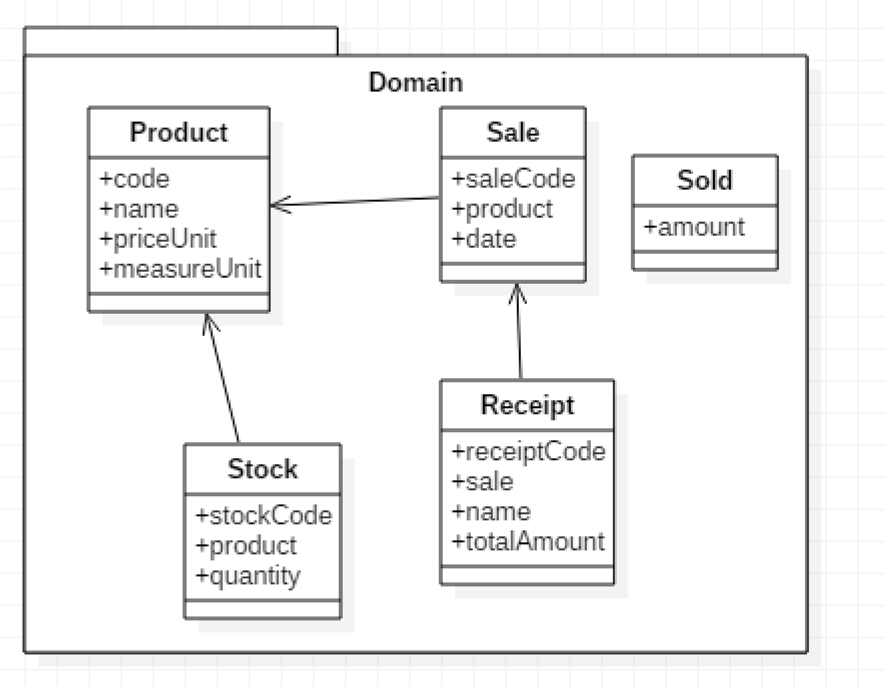
Sa se implementeze un magazin virtual care poate realiza vanzari de produse in paralel. Programul retine o lista de produse (nume, cod, pret, unitate de masura), stocuri (cod_produs, cantitate), vanzari (data, produs, cantitate), facturi (nume, vanzare, suma_totala) si soldul total. Periodic, sistemul face o verificare a stocurilor, produselor vandute si a sumei incasate de la ultima verificare. Programul va contine o UI minimala pentru initierea vanzarilor.

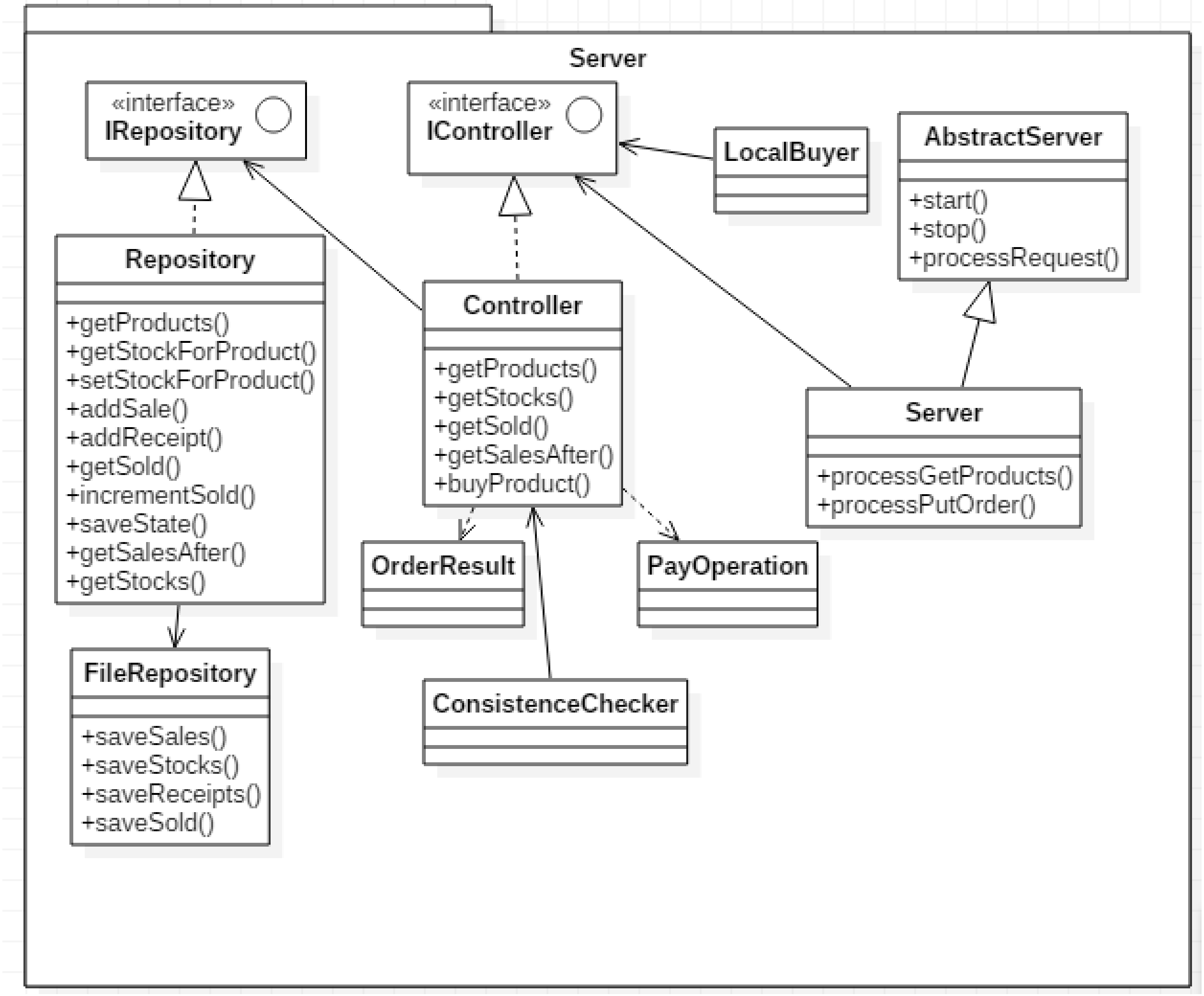
Restrictii

- sistemul se foloseste de un mecanism de tip Thread-Pool pentru rezolvarea concurenta a vanzarilor
- programul sa implementeze o arhitectura client-server

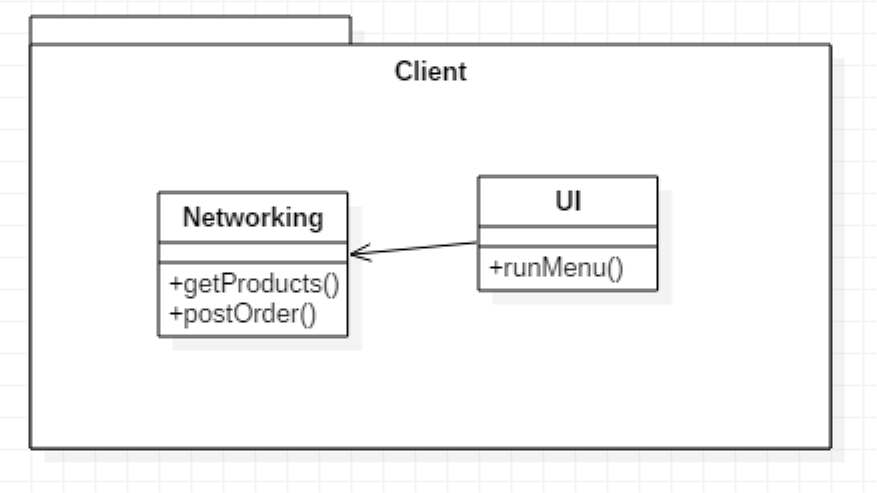
Proiectare

Clasele din domeniul problemei:





Partea de server contine un strat de persistenta (Repository) care mentine in memorie produsele, stocurile, vanzarile, facturile si le salveaza periodic in fisiere folosind metodele din FileRepository. Controllerul inglobeaza logica vanzarilor, LocalBuyer implementeaza un Runnable si executa comenzi aleatorii, ConsistenceChecker verifica integritatea si corectitudinea datelor, OrderResult inglobeaza rezultatul unei comenzi iar PayOperation implementeaza un Callable care contine o comanda. Server se ocupa de comunicarea prin retea si apeleaza metodele corespunzatoare cererilor.



Pachetul de client contine o clasa care se ocupa cu apelurile in retea (Networking) si o interfata utilizator simplista (UI).

Detalii de implementare

Fiecare operatie de comanda a unui produs este inglobata in PayOperation, clasa care implementeaza Callable, si trimisa unui ExecutorService. Pentru a garanta integritatea stocurilor am folosit cate un lock pentru fiecare produs. Astfel se pot face comenzi pentru produse diferite in paralel. PayOperation obtine lock pe produs si de asemenea un shared lock mentinut de de controller (controller.readWriteLock). Obtinerea unui readlock inseamna ca mai multe operatii pot fi executate in paralel. Threadul care verifica consistenta datelor obtine un exclusive lock pentru a garanta capturarea unei stari valide (evita problema in care alte threaduri reusesc sa actualizeze doar o parte din vanzari, facturi, stocuri). Rezultatul returnat de PayOperation este un Future care, la nivel de server, este pus intr-un thread separat sa astepte rezultatul si sa comunice cu clientul care a facut comanda.

Corectitudinea datelor din stocuri este garantata de faptul ca obtinem lockuri pe produse (nu se pot face actualizari de stoc pentru acelasi produs simultan). Accesul la lista de vanzari, facturi si la sold se face in mod sincronizat.

ConsistenceChecker implementeaza Runnable si determina comportamentul threadului care verifica periodic corectitudinea datelor. Acesta obtine exclusive lock pe controller.readWriteLock, deci nu se poate efectua in paralel cu operatii de vanzare. La fiecare iteratie verifica stocul produselor in comparatie cu cel vechi verificand numarul de produse vandute si soldul total verificand facturile.

Metode de testare

Rulare server cu un thread care face comenzi odata la 4 secunde si 2 clienti.

Comenzi client1

- nume: viorel, produs: 1, cantitate: 3

Comenzi client2

- nume: gheorghe, produs: 1, cantiate: 1
- nume: mircea, produs: 2, cantiate: 1

Comenzi thread local

- produs: 2, cantitate: 4
- produs: 2, cantiate: 5
- produs: 3, cantiate: 1
- produs: 1, cantitate: 1
- produs: 3, cantiate: 4

Logul de la server

```
Waiting for clients
Trying to buy : 2 , 4 , Thread-1
Receipt{name='Thread-1', sale=shop.domain.Sale@1606061, totalAmount=56.0}
Trying to buy : 2 , 5 , Thread-1
Receipt{name='Thread-1', sale=shop.domain.Sale@1fda537, totalAmount=70.0}
Client connected
Started processing
Reading data
Order: 1 1 gheorghe
Waiting for clients
Client connected
Started processing
Reading data
Order: 2 1 mircea
Waiting for clients
Trying to buy : 3 , 1 , Thread-1
Receipt{name='Thread-1', sale=shop.domain.Sale@10dae0a, totalAmount=20.0}
Trying to buy : 1 , 1 , Thread-1
```

```
Receipt{name='Thread-1', sale=shop.domain.Sale@7bccc5, totalAmount=10.0}
Client connected
Started processing
Reading data
Order: 1 3 viorel
Waiting for clients
Trying to buy : 3 , 4 , Thread-1
Not enough products on stock
```

Logul de la ConsistenceChecker

```
Correct values
The following sales were added
Correct values
The following sales were added
Masinax3 at Mon Nov 27 19:10:42 EET 2017
Correct values
The following sales were added
Masinax1 at Mon Nov 27 19:10:44 EET 2017
Correct values
The following sales were added
Bicicletax2 at Mon Nov 27 19:10:51 EET 2017
Correct values
The following sales were added
Bicicletax5 at Mon Nov 27 19:10:54 EET 2017
Trenx1 at Mon Nov 27 19:10:58 EET 2017
Correct values
The following sales were added
Masinax1 at Mon Nov 27 19:11:01 EET 2017
Bicicletax4 at Mon Nov 27 19:11:02 EET 2017
Correct values
The following sales were added
Masinax3 at Mon Nov 27 19:11:06 EET 2017
Trenx1 at Mon Nov 27 19:11:07 EET 2017
Correct values
The following sales were added
Correct values
The following sales were added
Bicicletax4 at Mon Nov 27 19:13:54 EET 2017
Correct values
The following sales were added
Bicicletax5 at Mon Nov 27 19:13:58 EET 2017
Masinax1 at Mon Nov 27 19:13:59 EET 2017
Correct values
The following sales were added
Bicicletax1 at Mon Nov 27 19:14:02 EET 2017
Trenx1 at Mon Nov 27 19:14:02 EET 2017
Correct values
The following sales were added
Masinax1 at Mon Nov 27 19:14:06 EET 2017
Masinax3 at Mon Nov 27 19:14:07 EET 2017
```

Facturi

```
Name: Thread-1, Total value: 56.0 , Sale id1
Name: Thread-1, Total value: 70.0 , Sale id2
Name: gheorghe, Total value: 10.0 , Sale id3
Name: mircea, Total value: 14.0 , Sale id4
Name: Thread-1, Total value: 20.0 , Sale id5
Name: Thread-1, Total value: 10.0 , Sale id6
Name: viorel, Total value: 30.0 , Sale id7
```

Vanzari

Code 1, Product code: 2, Quantity4, TimestampMon Nov 27 19:13:54 EET 2017

Code 2, Product code: 2, Quantity5, TimestampMon Nov 27 19:13:58 EET 2017

Code 3, Product code: 1, Quantity1, TimestampMon Nov 27 19:13:59 EET 2017

Code 4, Product code: 2, Quantity1, TimestampMon Nov 27 19:14:02 EET 2017

Code 5, Product code: 3, Quantity1, TimestampMon Nov 27 19:14:02 EET 2017

Code 6, Product code: 1, Quantity1, TimestampMon Nov 27 19:14:06 EET 2017

Code 7, Product code: 1, Quantity3, TimestampMon Nov 27 19:14:07 EET 2017

Sold : 210

Stocuri initiale

product code:1, Quantity: 10

product code:2, Quantity: 40

product code:3, Quantity: 2

Stocuri finale

product code:1, Quantity: 5

product code:2, Quantity: 30

product code:3, Quantity: 1