LABORATOR 4

Documentatie

Analiza cerintelor

Tema:

- Folosirea executiei concurente prin apeluri asincrone.
- Folosirea mecanismelor: future, promises si thread_pool.
- Analiza imbunatatirii performantei executiei unei aplicatii (de tip business) prin programare concurenta.

Sala concerte

O sala de concerte vinde bilete la spectacolele organizate printr-o aplicatie client-server. Sala are trei categorii de locuri cu preturi diferite. Pentru fiecare categorie exista o lista de locuri care pot fi vandute. Pentru fiecare spectacol avem informatii de tip (ID_spectacol, data, titlu, descriere). Se pot opera mai multe vanzari simultane !

Permanent se mentine o evidenta actualizata pentru:

- informatii despre bilete pentru fiecare spectacol (data, ID_spectacol, lista_locuri_ vandute);
- vanzarile efectuate: lista de vanzari; vanzare = (data_vanzare, ID_spectacol, numar_bilete, lista_locurilor) ;
- soldul total (suma totala incasata).

Periodic sistemul face o verificare a locurilor vandute prin verificarea corespondentei corecte intre locurile libere si vanzarile facute (de la ultima verificare pana in prezent), sumele incasate in aceeasi perioada si soldul total.

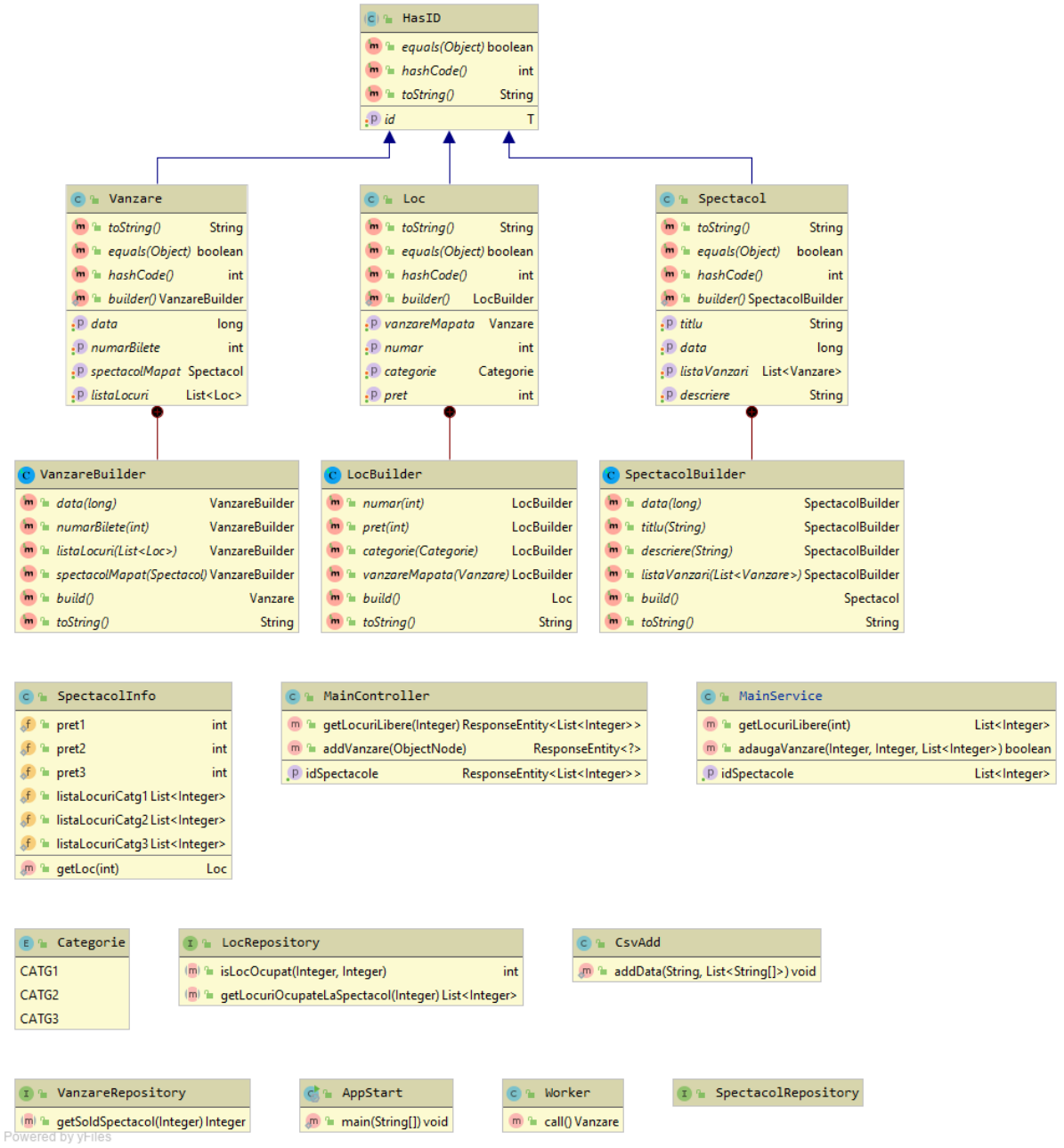
Sistemul foloseste un mecanism de tip ‘Thread-Pool’ pentru rezolvarea concurenta a vanzarilor.

Pentru a testare se va folosi un thread care initiaza/creeaza la interval de 5 sec o noua cerere de vanzare bilete folosind date generate aleatoriu. Pentru verificare se cere salvarea pe suport extern a soldului, a listei vanzarilor si a rezultatelor operatiilor de verificare executate periodic.

Proiectare

Pentru proiectarea aplicatiei s-au definit clasele Vanzare, Spectacol, Loc, HasId, SpectacolInfo si enum-ul Categorie. De asemenea, s-au definit clasele SpectacolRepository, LocRepository, VanzareRepository, MainService si MainController, care se ocupa de logica aplicatiei.

Atat clientul cat si serverul au fost scrise in Java. Pe partea de server s-a folosit Hibernate, persistenta detelor facandu-se intr-o baza de date MySQL.



Detalii de implementare

Aplicatia foloseste servicii REST pentru comunicarea intre client si server. S-au definit urmatoarele rute:

localhost:3000/spectacole (GET) => Lista id-urilor spectacolelor

localhost:3000/locuri/{id} (GET) => Lista locurilor libere la spectacolului cu id-ul “id”

localhost:3000/vanzare (POST) => Se face o vanzare la un anumit spectacol identificat prin id-ul sau.

Body-ul metodei contine o entitate de timp VanzareClient (id: int, numar: int, locuri: int[]), unde field-urile au urmatoarele semnificatii:

Id => id-ul spectacolului la care se doreste sa se cumpere bilete

Numar => numarul biletelor care se doresc a fi cumparate

Locuri => lista de intregi ce reprezinta numarul unic de identificare a unui scaun in sala

In cazul in care in timp ce un client cumpara un bilet pe un anumit loc, iar alt client tocmai a facut tranzactia pe acelasi loc, tranzactia primului client e anulata.

Pentru a putea deservi mai multi clienti in acelasi timp, serverul a fost implementat folosind programarea concurenta. S-a utilizat *synchronized* pentru ca un singur client sa poata accesa la un moment dat o regiune critica.

```
public List<Integer> getIdSpectacole() {
    List<Integer> lista = new ArrayList<>();
    synchronized (spectacolRepository) {
        for (Spectacol s : spectacolRepository.findAll()) {
            lista.add(s.getId());
        }
    }
    return lista;
}
```

De asemenea, cererea fiecarui client este executata de cate un thread dintr-un thread pool de dimensiune NTHREADS. Rezultatul threadurilor este retinut intr-o lista, iar cand aceasta lista a ajuns la o anumita capacitate, se realizeaza un raport cu toate datele informatiile spectacole, dupa care se goleste lista.

```
private static final int NTHREDS = 8;
private ExecutorService executor = Executors.newFixedThreadPool(NTHREDS);
private List<Future<Vanzare>> list = new ArrayList<>();
```

...

```
Callable<Vanzare> worker = new Worker(vanzareRepository, locRepository, vanzare, locuriSpectacol);
Future<Vanzare> submit = executor.submit(worker);



boolean isAdded = submit.get() != null;

synchronized (list) {
    if (isAdded) {
        list.add(submit);
    }
}

CompletableFuture
        .runAsync(this::workerRaports)
        .join();
```

Pe partea de server s-au folosit mecanismele Future, CompletableFuture si ThreadPool. Clasa future a fost folosita pentru a obtine rezultatul executiei unui thread. CompletableFuture a fost facuta pentru a crea rapoartele, iar ThreadPool a fost utilizata pentru a limita numarul de threaduri care se folosesc si, de asemenea, pentru a le reutiliza.

Cazuri de testare

Pentru testarea aplicatiei se fac rapoarte periodice. Aceste rapoarte se salveaza in fisiere CSV.

Dupa 4 tranzactii facute de 2 clienti:

Solduri.csv:

```
"ID_spectacol","sold"
"1","60"
"2","10"
"3","10"
```

Vanzari.csv:

```
"data_vanzare","ID_spectacol","numar_bilete","lista_locurilor"

"18-12-2019 10:35:55","2","1","25"
"18-12-2019 10:36:01","1","3","275"
"18-12-2019 10:36:01","1","3","268"
"18-12-2019 10:36:01","1","3","265"
"18-12-2019 10:36:06","3","1","202"
"18-12-2019 10:36:08","1","3","195"
"18-12-2019 10:36:08","1","3","59"
"18-12-2019 10:36:08","1","3","107"
```

Bilete.csv:

```
"data","ID_spectacol","lista_locuri_vandute"
"19-01-1970 07:53:51","1","275"
"19-01-1970 07:53:51","1","268"
```

```
"19-01-1970 07:53:51","1","265"
"19-01-1970 07:53:51","1","195"
"19-01-1970 07:53:51","1","59"
"19-01-1970 07:53:51","1","107"
"19-01-1970 07:53:51","2","25"
"19-01-1970 07:53:51","3","202"
```

Rezultat:

Spectacol 1: 1 bilet 10 lei

Spectacol 2: 6 bilete 10 lei

Spectacol 3 : 1 bilet 10 lei

Dupa 29 de tranzactii facute de 3 clienti:

Solduri.csv:

```
"ID_spectacol","sold"
"1","200"
"2","150"
"3","190"
```

Vanzari.csv:

```
"data_vanzare","ID_spectacol","numar_bilete","lista_locurilor"
"18-12-2019 10:35:55","2","1","25"
"18-12-2019 10:36:01","1","3","275"
"18-12-2019 10:36:01","1","3","268"
"18-12-2019 10:36:01","1","3","265"
"18-12-2019 10:36:06","3","1","202"
"18-12-2019 10:36:08","1","3","195"
"18-12-2019 10:36:08","1","3","59"
"18-12-2019 10:36:08","1","3","107"
"18-12-2019 10:42:07","1","1","121"
"18-12-2019 10:42:13","1","2","41"
"18-12-2019 10:42:13","1","2","249"
"18-12-2019 10:42:19","3","1","165"
"18-12-2019 10:42:25","3","3","189"
"18-12-2019 10:42:25","3","3","271"
"18-12-2019 10:42:25","3","3","197"
"18-12-2019 10:42:30","2","1","116"
"18-12-2019 10:42:32","3","3","142"
"18-12-2019 10:42:32","3","3","243"
"18-12-2019 10:42:32","3","3","183"
"18-12-2019 10:42:35","1","1","234"
"18-12-2019 10:42:36","1","3","187"
"18-12-2019 10:42:36","1","3","122"
"18-12-2019 10:42:36","1","3","164"
"18-12-2019 10:42:38","1","1","243"
```

```
"18-12-2019 10:42:41","1","1","193"
"18-12-2019 10:42:43","2","3","184"
"18-12-2019 10:42:43","2","3","274"
"18-12-2019 10:42:43","2","3","237"
"18-12-2019 10:42:45","3","1","156"
"18-12-2019 10:42:46","2","2","139"
"18-12-2019 10:42:46","2","2","276"
"18-12-2019 10:42:48","3","1","157"
"18-12-2019 10:42:51","3","3","164"
"18-12-2019 10:42:51","3","3","255"
"18-12-2019 10:42:51","3","3","220"
"18-12-2019 10:42:52","1","3","161"
"18-12-2019 10:42:52","1","3","150"
"18-12-2019 10:42:52","1","3","105"
"18-12-2019 10:42:53","2","3","145"
"18-12-2019 10:42:53","2","3","281"
"18-12-2019 10:42:53","2","3","220"
"18-12-2019 10:42:56","2","2","118"
"18-12-2019 10:42:56","2","2","287"
"18-12-2019 10:42:57","1","1","252"
"18-12-2019 10:42:59","3","1","122"
"18-12-2019 10:43:01","3","2","214"
"18-12-2019 10:43:01","3","2","145"
"18-12-2019 10:43:02","3","3","136"
"18-12-2019 10:43:02","3","3","151"
"18-12-2019 10:43:02","3","3","199"
"18-12-2019 10:43:04","2","1","232"
"18-12-2019 10:43:07","1","1","138"
```

Bilete.csv:

```
"data","ID_spectacol","lista_locuri_vandute"
"19-01-1970 07:53:51","1","275"
"19-01-1970 07:53:51","1","268"
"19-01-1970 07:53:51","1","265"
"19-01-1970 07:53:51","1","195"
"19-01-1970 07:53:51","1","59"
"19-01-1970 07:53:51","1","107"
"19-01-1970 07:53:51","1","121"
"19-01-1970 07:53:51","1","41"
"19-01-1970 07:53:51","1","249"
"19-01-1970 07:53:51","1","234"
"19-01-1970 07:53:51","1","187"
"19-01-1970 07:53:51","1","122"
"19-01-1970 07:53:51","1","164"
"19-01-1970 07:53:51","1","243"
"19-01-1970 07:53:51","1","193"
"19-01-1970 07:53:51","1","161"
"19-01-1970 07:53:51","1","150"
"19-01-1970 07:53:51","1","105"
"19-01-1970 07:53:51","1","252"
"19-01-1970 07:53:51","1","138"
"19-01-1970 07:53:51","2","25"
"19-01-1970 07:53:51","2","116"
```

```
"19-01-1970 07:53:51","2","184"
"19-01-1970 07:53:51","2","274"
"19-01-1970 07:53:51","2","237"
"19-01-1970 07:53:51","2","139"
"19-01-1970 07:53:51","2","276"
"19-01-1970 07:53:51","2","145"
"19-01-1970 07:53:51","2","281"
"19-01-1970 07:53:51","2","220"
"19-01-1970 07:53:51","2","118"
"19-01-1970 07:53:51","2","287"
"19-01-1970 07:53:51","2","232"
"19-01-1970 07:53:51","3","202"
"19-01-1970 07:53:51","3","165"
"19-01-1970 07:53:51","3","189"
"19-01-1970 07:53:51","3","271"
"19-01-1970 07:53:51","3","197"
"19-01-1970 07:53:51","3","142"
"19-01-1970 07:53:51","3","243"
"19-01-1970 07:53:51","3","183"
"19-01-1970 07:53:51","3","156"
"19-01-1970 07:53:51","3","157"
"19-01-1970 07:53:51","3","164"
"19-01-1970 07:53:51","3","255"
"19-01-1970 07:53:51","3","220"
"19-01-1970 07:53:51","3","122"
"19-01-1970 07:53:51","3","214"
"19-01-1970 07:53:51","3","145"
"19-01-1970 07:53:51","3","136"
"19-01-1970 07:53:51","3","151"
"19-01-1970 07:53:51","3","199"
```

Rezultat:

Spectacol 1: 20 bilete 10 lei

Spectacol 2: 15 bilete 10 lei

Spectacol 3 : 19 bilete 10 lei