Bucșa Ecaterina, gr. 231

**Laborator 6**

**-documentație-**

**Cerința**:

Laborator 4

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.

**Sală concerte**

O sală de concerte vinde bilete la spectacolele organizate printr-o aplicație client-server. Sala are trei categorii de locuri cu prețuri diferite. Pentru fiecare categorie există o listă de locuri care pot fi vândute. Pentru fiecare spectacol avem informații de tip (ID\_spectacol, dată, titlu, descriere). Se pot opera mai multe vânzări simultane !

Permanent se menține o evidență actualizată pentru:

- informații despre bilete pentru fiecare spectacol - (data, ID\_spectacol, lista\_locuri\_ vandute);

- vânzările efectuate: lista de vânzări; vânzare = (data\_vanzare, ID\_spectacol, numar\_bilete, lista\_locurilor) ;

- soldul total (suma totală încasată).

Periodic sistemul face o verificare a locurilor vândute prin verificarea corespondenței corecte între locurile libere și vânzările făcute (de la ultima verificare până în prezent), sumele încasate în aceeași perioada și soldul total. Sistemul folosește un mecanism de tip ‘Thread-Pool’ pentru rezolvarea concurentă a vânzărilor. Pentru testare se va folosi un thread care inițiază/creează la interval de 5 sec o nouă cerere de vânzare bilete folosind date generate aleatoriu. Pentru verificare se cere salvarea pe suport extern a soldului, a listei vanzarilor si a rezultatelor operațiilor de verificare executate periodic.

Limbajul de implementare: la alegere

Descriere soluție:

Am ales să implementez soluția acestei probleme in JAVA. Am folosit mecanismele Thread-Pool, Future și CompletableFuture.

Aplicația server creează un socket și așteaptă pe un anumit port cereri de la clienți. Atunci când un client s-a conectat, se creează un Callable care este executat cu ajutorul unui Thread Pool. Rezultatul este primit într-un obiect de tip Future, pentru ca serverul să nu aștepte deservirea clientului și să poată trece la servirea unei alte cereri, rezultatul fiind trimis clientului într-un thread separat. De asemenea, un thread se ocupă de afișarea detaliilor despre vânzări.

Aplicația client creează un socket și prin intermediul căruia trimite cereri serverului la un interval de 5 secunde, cereri de cumpărare bilete, iar rezultatul primit(detaliile despre biletul cumpărat) este afișat în consolă.

Diagrama de clase:

