

Proiect 1 – Client-Server

Sintea Maria, Suci Andra, 237/1

Cerinta

O sala de spectacole vinde bilete la spectacolele organizate printr-o aplicatie client-server, sala organizand cel mult un spectacol pe zi. Locurile din sala sunt numerotate de la 1 la numarul maxim de locuri 'nr_locuri'. Pentru fiecare spectacol avem urmatoarele informatii: data, titlu, pret_bilet. Sala va mentine permanent o evidenta actualizata pentru:

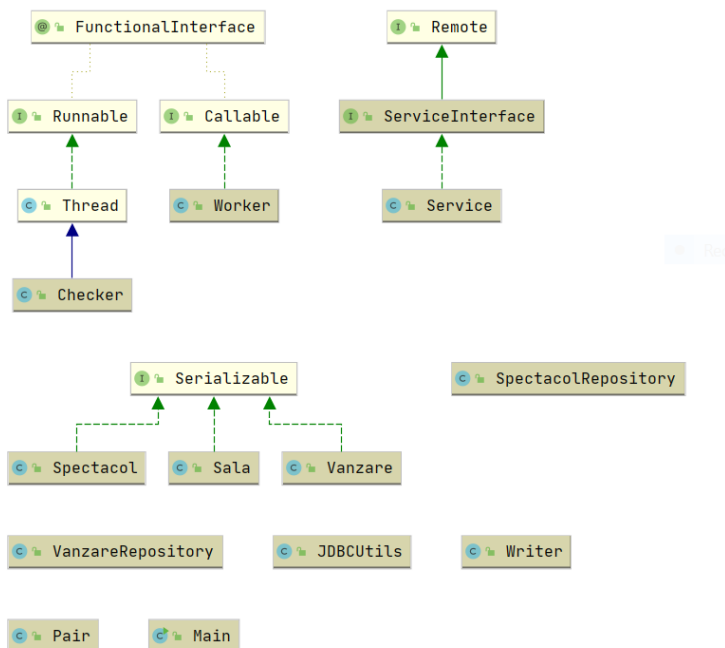
- informatii despre bilete pentru fiecare spectacol - ID_spectacol, lista_locuri_vandute
- vanzarile efectuate - lista de vanzari; pentru o vanzare se retin urmatoarele informatii: data_vanzare, ID_spectacol, numar_bilete, lista_locurilor
- soldul total - suma totala incasata

Periodic sistemul face o verificare a locurilor vandute prin verificarea corespondentei corecte intre locurile libere si vanzarile facute, sumele incasate per vanzare si soldul total. Sistemul foloseste un mecanism de tip Thread-Pool pentru rezolvarea a taskurilor.

Serverul se inchide dupa un interval de timp precizat si notifica clientii activi referitor la inchidere. Implementarea se va face cu executie concurenta, prin apeluri asincrone. Se vor folosi urmatoarele mecanisme: future/promises si thread_pool.

Proiectare

Diagrama de clase:



Model:

- Spectacol (ID_spectacol, data_spectacol, titlu, pret_bilet, lista_locuri_vandute, sold)
- Vanzare (ID_spectacol, data_vanzare, nr_bilete_vandute, lista_locuri_vandute, suma)
- Sala (nr_locuri, Lista<Spectacol>, Lista<Vanzari>)

Am folosit Java RMI pentru implementarea conexiunii intre client si server, aceasta facandu-se prin apeluri Remote care se fac prin interfata ServiceInterface, ce are metoda saveVanzare. Implementarea acestei interfete, clasa Service, se va ocupa de verificarea si adaugarea vanzarii: va adauga in repository, in locuri vandute si sume incasate.

La inceputul executiei programului vom da drumul la un nou Thread numit Checker, ce va verifica la fiecare 5/10 secunde daca locurile vândute erau libere inainte de vanzare si daca toate biletele vandute au sumele aferente incasate. De asemenea, Checker va verifica si daca sumele incasate se aduna, intr-adevar la soldul total.

Testare

Date de intrare pentru testare - nr_locuri = 100 si 3 spectacole (S1, S2, S3):

- pret_bilet S1 = 100
- pret_bilet S2 = 200
- pret_bilet S3 = 150

Serverul lucreaza 2 min, iar verificarea locurilor se face cu doua cazuri de testare: 5 si 10 sec.

Se va considera ca fiecare client initiaza/creeaza la interval de 2 sec o noua cerere de vanzare bilete folosind date generate aleatoriu (nr_de_bilete, locuri).