Proiect 1 – Client-Server

Obiectiv:
- 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 spectacole
O sala de spectacole vinde bilete la spectacolele organizate printr-o aplicatie client-server.
Sala organizeaza cel mult un spectacol pe zi.
Sala de concerte are un numar maxim - 'nr_locuri' - de locuri numerotate de la 1 la 'nr_locuri'.
Pentru fiecare spectacol avem informatii de tip (data, titlu, pret_bilet).

Permanent sala mentine o evidenta actualizata pentru:
- informatii despre bilete pentru fiecare spectacol - (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 (2 cazuri testare: 5, 10 secunde) 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.

Pentru testare 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) si se primeste de la server o notificare – vanzare reusita sau vanzare nereusita. Nu este necesara interfata grafica!

Pentru verificare se cere salvarea pe suport extern (fisier text) a rezultatelor operatiilor de verificare executate periodic: data, ora, sold_per spectacol, lista vanzarilor per spectacol, 'corect/incorect'.

Serverul se inchide dupa un interval de timp precizat si notifica clientii activi referitor la inchidere.

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>)

Taskuri posibile
- Vanzare bilete
- Verificare

Limbajul de implementare: la alegere!!!

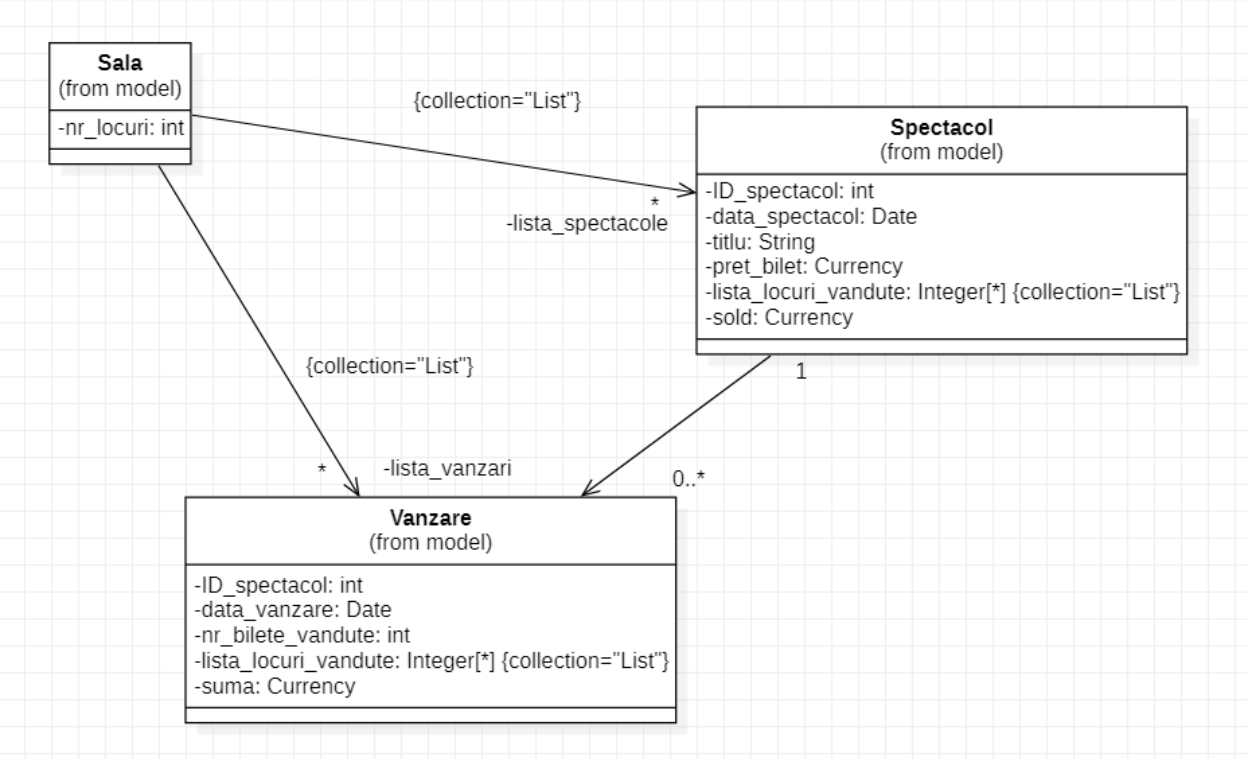
Pentru implementarea conexiunii intre client si server se pot folosi oricare dintre tehnologiile folosite la cursurile anterioare!!!

Testare:

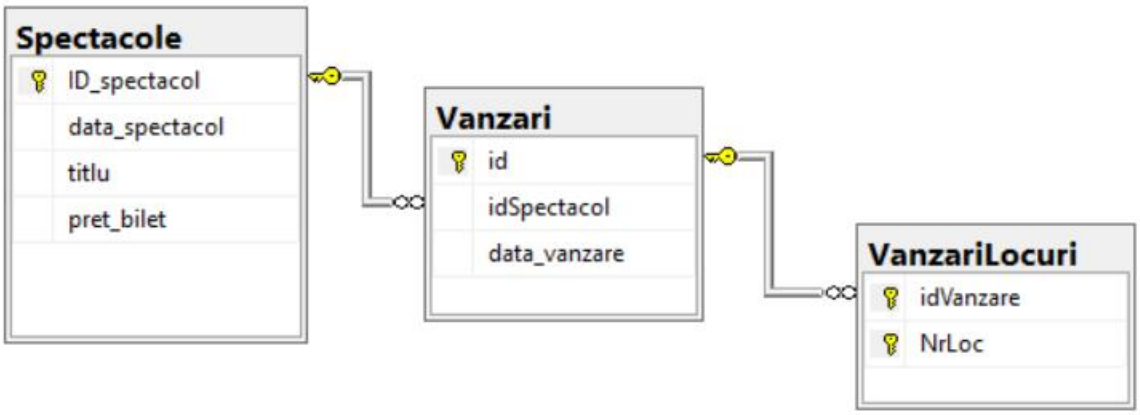
Nr_locuri =100;
3 spectacole (S1, S2, S3)
S1 pret_bilet=100;
S2 pret_bilet=200;
S3 pret_bilet=150;
Serverul lucreaza 2 minute.

Deadline: saptamana 13

Indicatii:
Un model posibil este cel din fig. de mai jos:



Salvarea datelor poate fi facuta intr-o baza de date de tipul:



Sau in fisiere text:

Spectacol.txt:

ID_spectacol1, data_spectacol1, titlu1, pret_bilet,
1,5,6,7,3 ## lista_locuri_vandute
sold;

ID_spectacol2, data_spectacol2, titlu2, pret_bilet,
1,2,3,6,7,8
sold;
.
.

Vanzare.txt:

ID_spectacol1, data_vanzare1, nr_bilete_vandute,
1,5 ## lista_locuri_vandute
suma;

ID_spectacol1, data_vanzare2, nr_bilete_vandute,
6,7,3
suma;

ID_spectacol2, data_vanzare1, nr_bilete_vandute,
1,2,3
suma;
.
.

Sala.txt: nr_locuri,
ID_spectacol1, ID_spectacol2, ID_spectacol3,…etc ## lista_spectacole
ID_spectacol1, data_vanzare1; ID_spectacol1, data_vanzare2, …etc ## lista_vanzari
ID_spectacol2, data_vanzare1; ID_spectacol2, data_vanzare2, …etc