**QUEUES SIMULATOR**

Cozile sunt utilizate în mod obișnuit pentru modelarea domeniilor din lumea reală. Principalul obiectiv al unei cozi este să furnizeze un loc pentru ca un „client” să aștepte înainte de a primi un „serviciu”. In gestionarea bazată pe coadă sistemul este interesat să minimizeze timpul în care „clienții” lor așteaptă la coadă înainte ca ei sa fie serviti. Un mod de a minimiza timpul de așteptare este să adăugați mai multe servere, adică mai multe cozi în sistemul (fiecare coadă este considerată ca având un procesor asociat), dar această abordare crește costurile furnizorului de servicii. Când se adaugă un server nou, clienții în așteptare vor fi uniform distribuiti la toate cozile disponibile curente.

Se citesc dintr-un fisier urmatoarele date de intrare:

* N reprezinta numarul de clienti generati pentru simulare
* Q numarul de cozi
* 𝑡𝑠𝑖𝑚𝑢𝑙𝑎𝑡𝑖𝑜𝑛𝑀𝐴𝑋 timpul maxim al simulari
* 𝑡𝑎𝑟𝑟𝑖𝑣𝑎𝑙 𝑀𝐼𝑁, 𝑡𝑎𝑟𝑟𝑖𝑣𝑎𝑙 𝑀𝐴𝑋 arrival time (𝑡𝑎𝑟𝑟𝑖𝑣𝑎𝑙 𝑀𝐼𝑁 ≤ 𝑡𝑎𝑟𝑟𝑖𝑣𝑎𝑙 ≤ 𝑡𝑎𝑟𝑟𝑖𝑣𝑎𝑙 𝑀𝐴𝑋 ), intervalul de timp in care un client poate sa mearga la una dintre cozi
* service time (𝑡𝑠𝑒𝑟𝑣𝑖𝑐𝑒 𝑀𝐼𝑁 ≤ 𝑡𝑠𝑒𝑟𝑣𝑖𝑐𝑒 ≤ 𝑡𝑠𝑒𝑟𝑣𝑖𝑐𝑒 𝑀𝐴𝑋 ), intervalul de timp in care trebuie sa se incadreze timpul de servire a unui client din momentul in care ajunge la casa, in varful cozii.

Date de iesire trebuie stocate intr-un fiser sub forma:

Time 0

Waiting clients: (1,2,2); (2,3,3); (3,4,3); (4,10,2)

Queue 1: closed

Queue 2: closed

Time 1

Waiting clients: (1,2,2); (2,3,3); (3,4,3); (4,10,2)

Queue 1: closed

Queue 2: closed

Time N

Queue 1: closed

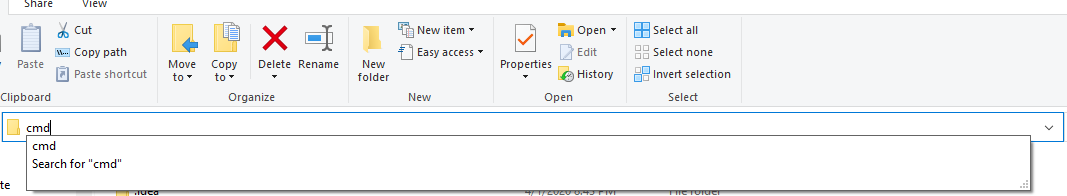
Queue 2: closed

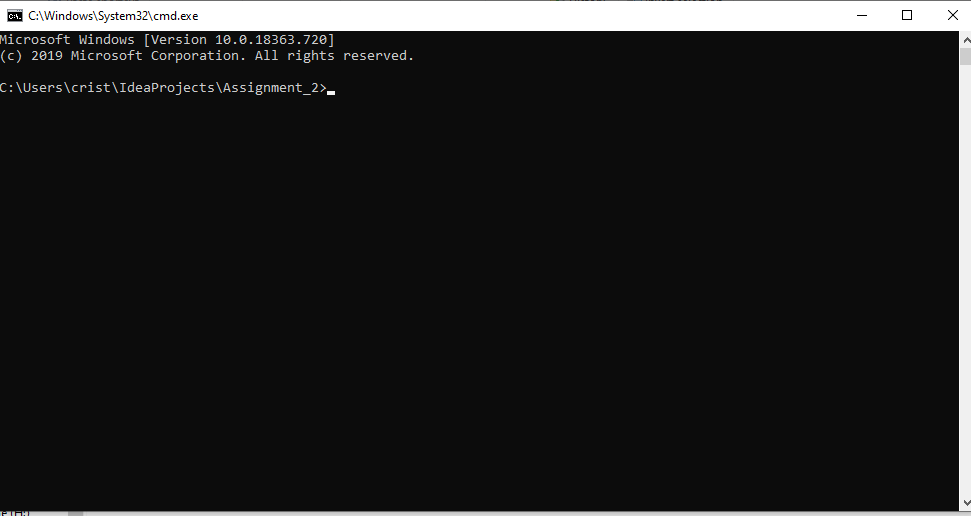
Average waiting time: 2.5 - timpul mediu de asteptare pentru clientii procesatii pana la time N

Aplicatia trebuie sa primeasca ca argumente fisierul de intrare si fisierul de iesire.

***Analiza problemei***

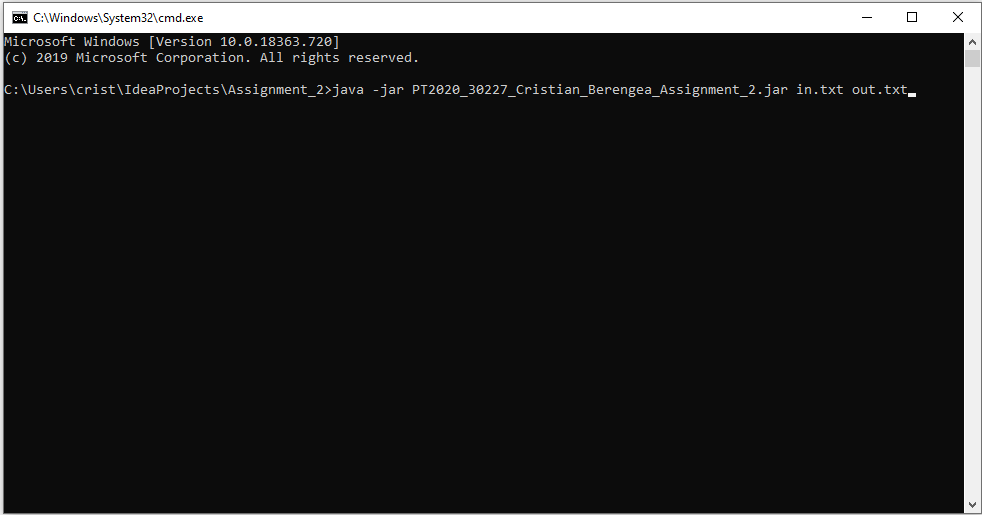
Utilizatorul trevuiesa deschida Command Prompt (Consola), scriind in campul care contine calea cmd , apoi trebuie sa apese Enter.





Utilizatorul trebuie sa tasteze comada urmatoare:

QueingAnalyse.jar <fisierul 1> <fisierul 2>.



Programul ruleaza, sau afiseaza un mesaj de eroare in cazul in care comanda nu are formatul corect sau in cazul in care fisierele introduse ca argumente nu pot deschise.

Cand aplicatia porneste este afisata o fereastra care contine informatiile citite din fisierul de intrare.

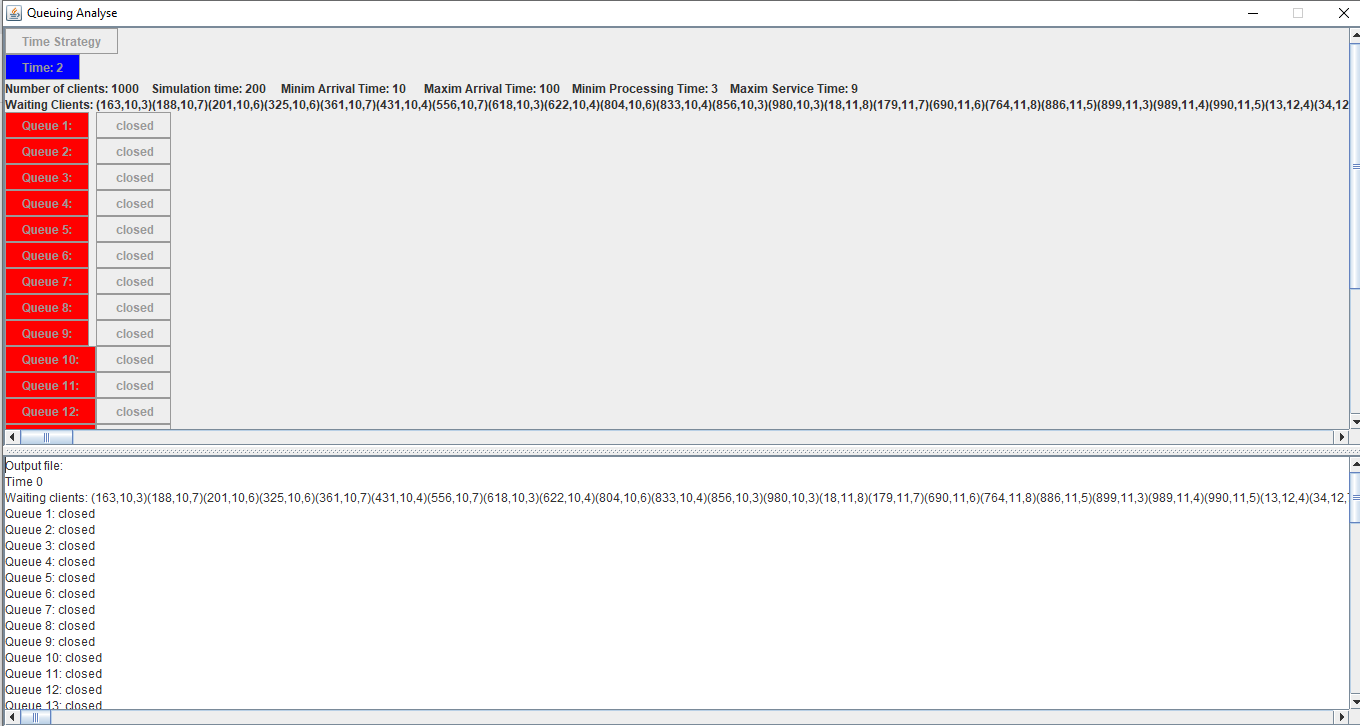


Mai este doar un pas pe care utilizatorul trebuie sa il parcurga pentru a incepe simularea, acesta trebuie sa aleaga ce strategie sa foloseasca strategia printr-un click pe unul dintre cele 2 butoane . Exista doua strategi posibile:

1. Time Strategy Clientul este repartizat in coada care are perioada de asteptare minima
2. Number Of Clients Strategy Clientul este repartizat in coada care contine numarul minim de clienti.

Dupa rulare fisierul de iesire o sa contina datele de iesire corespunzatoare.

Interfata ofera o vizualisare in timp real a simularii.

Dupa ce utilizatorul apasa un buton, alegand strategia, simularea incepe. Se afiseaza la fiecare moment de timp situatia actualizata a cozilor. Timul de servire este decrementat cu fiecare secunda pentru clientul care este in curs de procesare. 

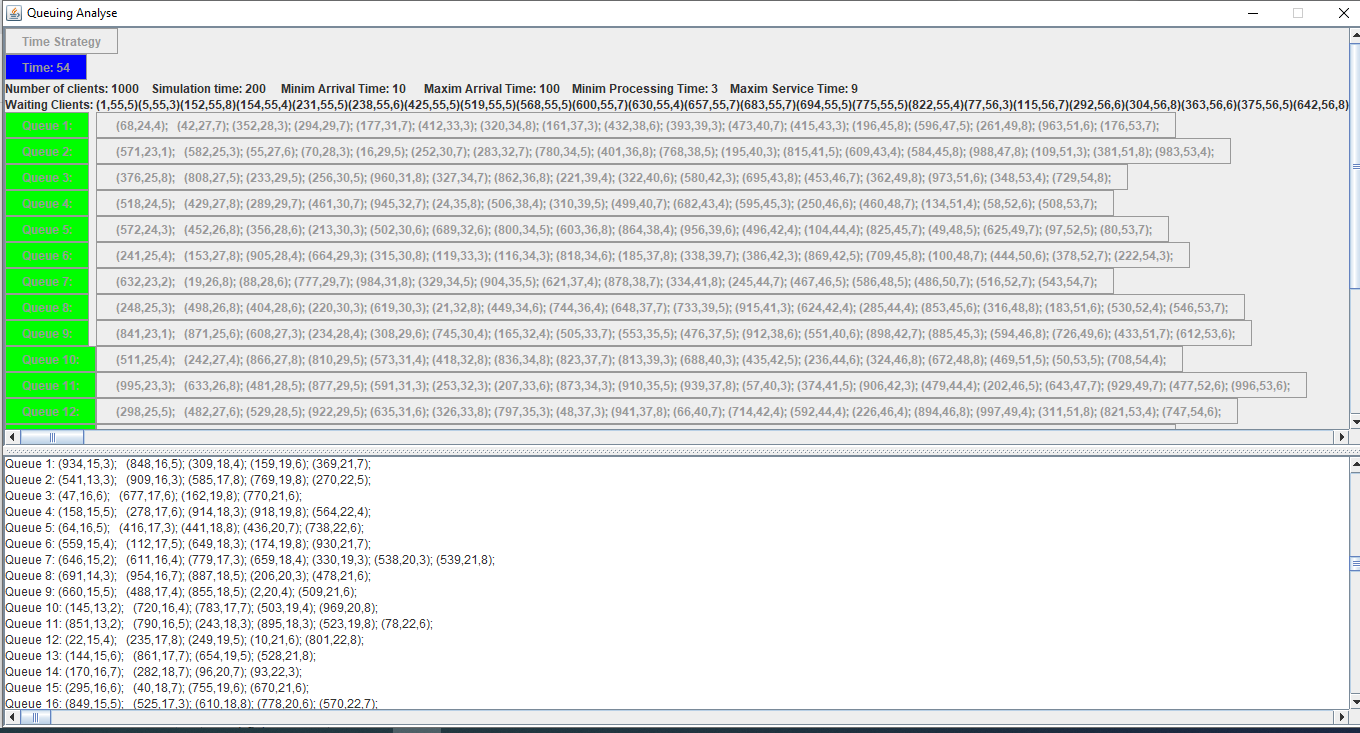
Atunci cand cozile nu cotin clienti acestea sunt inchise. Culoarea butonului este rosie.

Daca in coada sunt clienti, angajatul proceseza cererile clientilor.

De asemenea sunt afisati clientii care sunt inca in magazin, si care la timpul potrivit o sa se indrepte spre cozi.

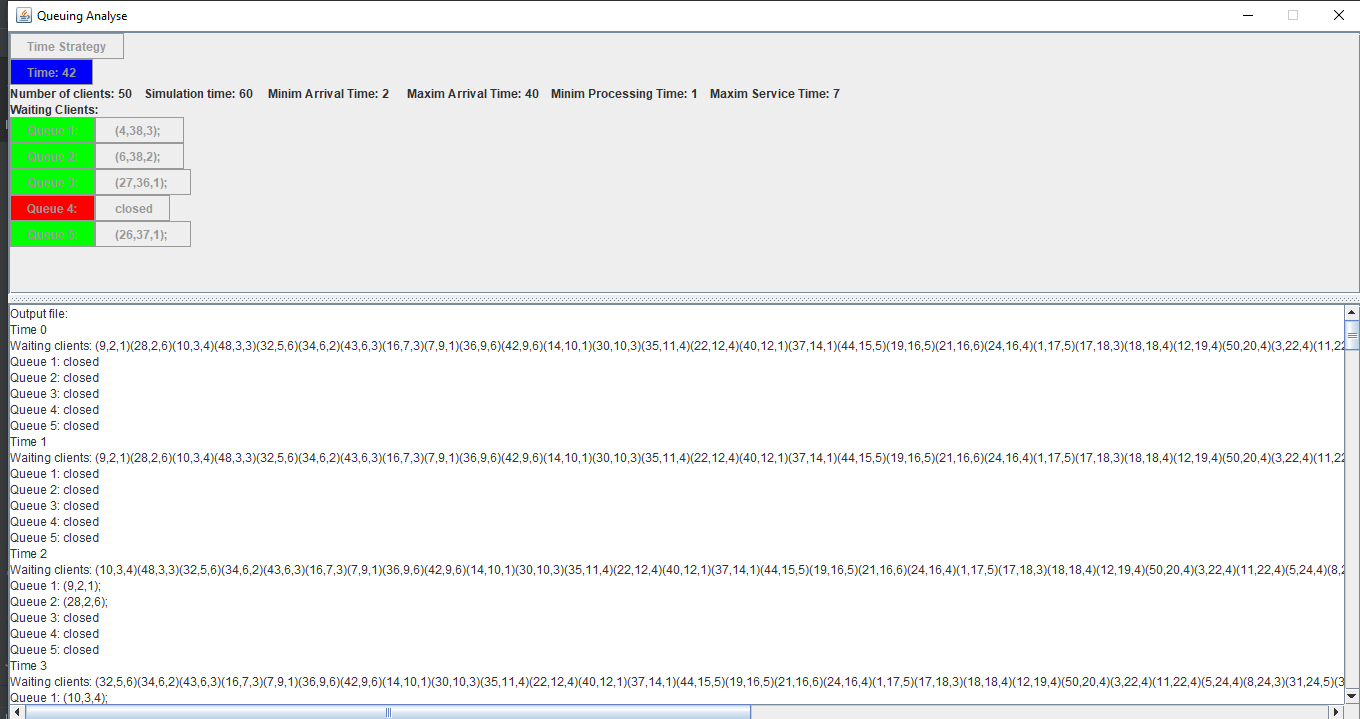


In partea de jos este afisat continutul disierului de iesire.



Butonul albastru in care este afisat timpul palpaie atunci cand contorul timpului trece la secunda urmatoare.

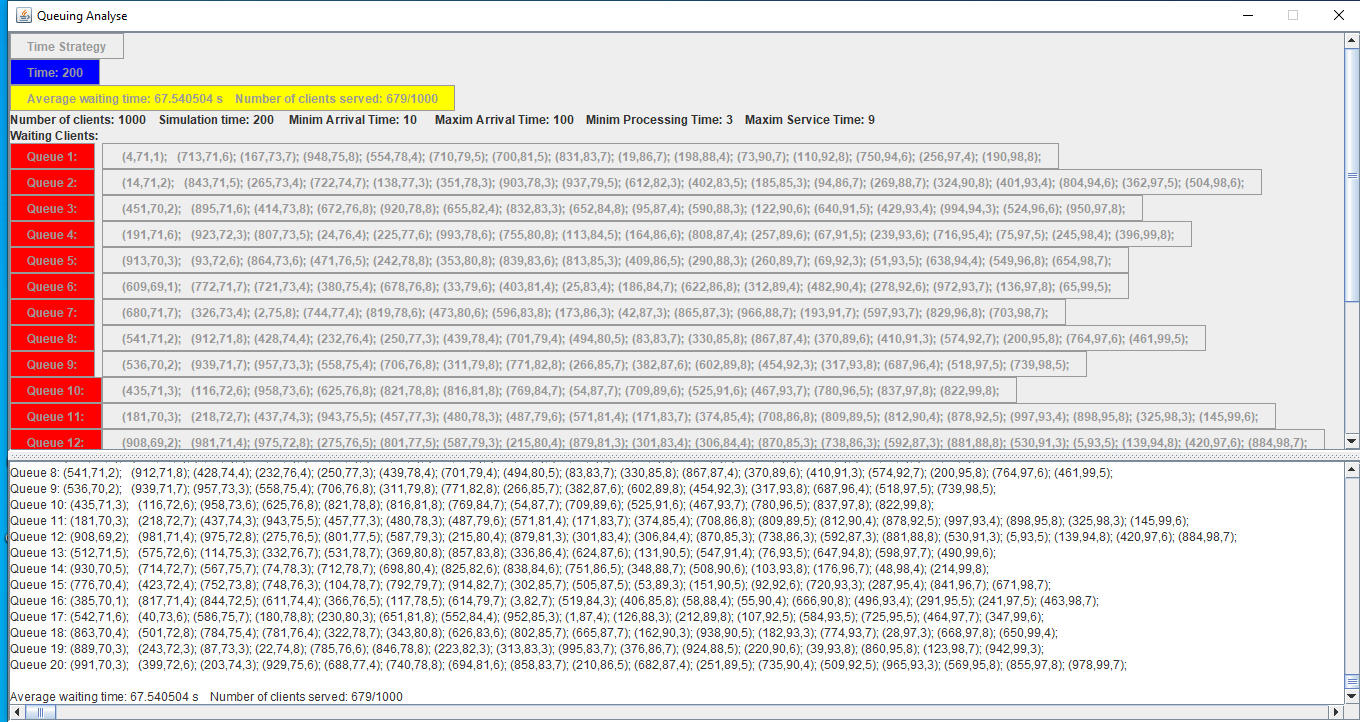
Cozile care au clienti sunt verzi, atunci cand nu mai exista clentii butonul devine rosu, coada fiind inchisa.

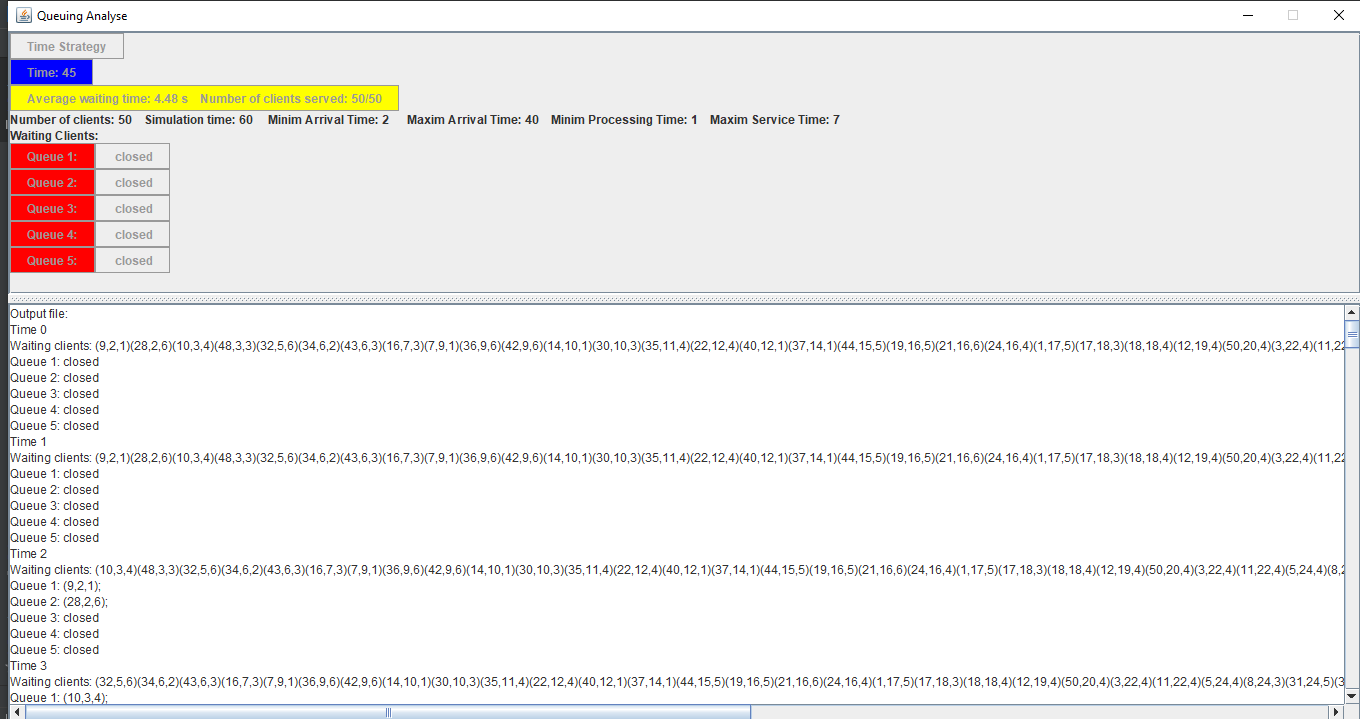


Cand simularea ajunge in momentu de timp egal cu 𝑡𝑠𝑖𝑚𝑢𝑙𝑎𝑡𝑖𝑜𝑛𝑀𝐴𝑋 ( timpul maxim al simulari ) sau nu mai exista clienti in magazin, se afiseaza un cam care contine rezultatele calculate :

- timpul mediu de asteptare

- numarul de clienti procesati, este posibil sa nu fie procesati toti clientii, unii pot sa ramana inca in cozii la finalul simularii





***Proiectare***

***Proiectare clase***

Am proiectat clasele / interfetele:

* Client - inglbeaza informatii reprezentative pentru client
* Serer – contine o lista de clienti si are asociat un thread, threadul ruleaza atunci cand in coada sunt clienti, daca coada ramane fara clienti threadul “ moare ”, la aparitia unui client este instantiat un thread nou, iar apoi este apelata metoda run.
* interfata Strategy - contine anetul unei singure metode, aceasta metoda o sa adauge un client intr-o coada
* TimeStraegy – implementeaza interfata Strategy, aceasta strategie adauga clientul in coada cu cel mai mic timp total de procesare ( in general este cea mai buna strategie )
* NumberStrategy – implementeaza interfata Strategy, aceasta strategie adauga clientul in coada cu cel mai mic numar de clienti care se aflain coada
* Scheduler - aceasta clasa este responsabile cu repartizarea clientilor in cozi, in funtie de strategia aleasa
* Simulation manager – initializeaza clasa scheduler si interfata gui, citeste / afiseaza datele de intrare / iesire in fisiere, prin intermediul unui thread tine timpul simulari si face un managemen al cozilor cu ajutorul clasei sceduler
* Ui – acesta clasa contine implementarea interfetei grafice

