

## 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”. În gestionarea bazată pe coadă sistemul este interesat să minimizeze timpul în care „clienții” lor așteaptă la coadă înainte ca ei să fie serviți. 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 distribuiți 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
- $t_{simulation}MAX$  timpul maxim al simularii
- $t_{arrival} MIN, t_{arrival} MAX$  arrival time ( $t_{arrival} MIN \leq t_{arrival} \leq t_{arrival} MAX$ ), intervalul de timp in care un client poate sa mearga la una dintre cozi
- service time ( $t_{service} MIN \leq t_{service} \leq t_{service} MAX$ ), 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

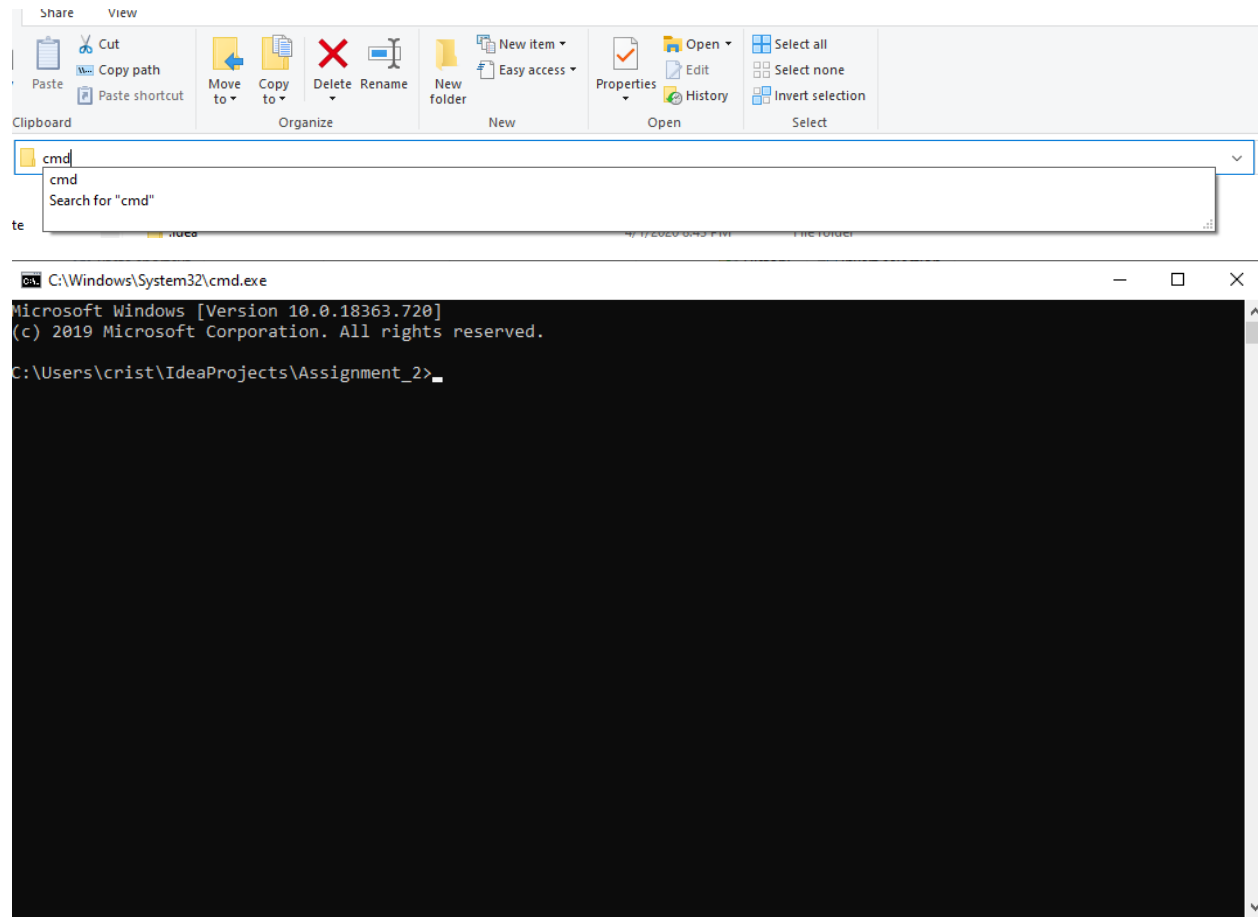
Queue 2: closed

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.

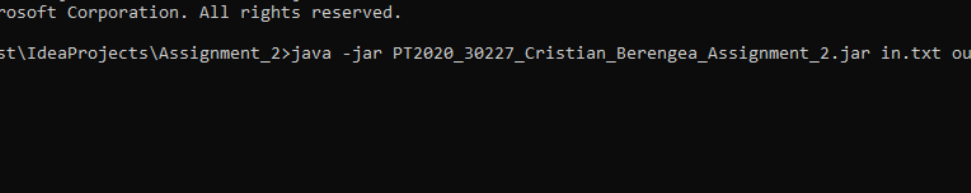
## ***Analiza problemei***

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



Utilizatorul trebuie sa tasteze comanda urmatoare:

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

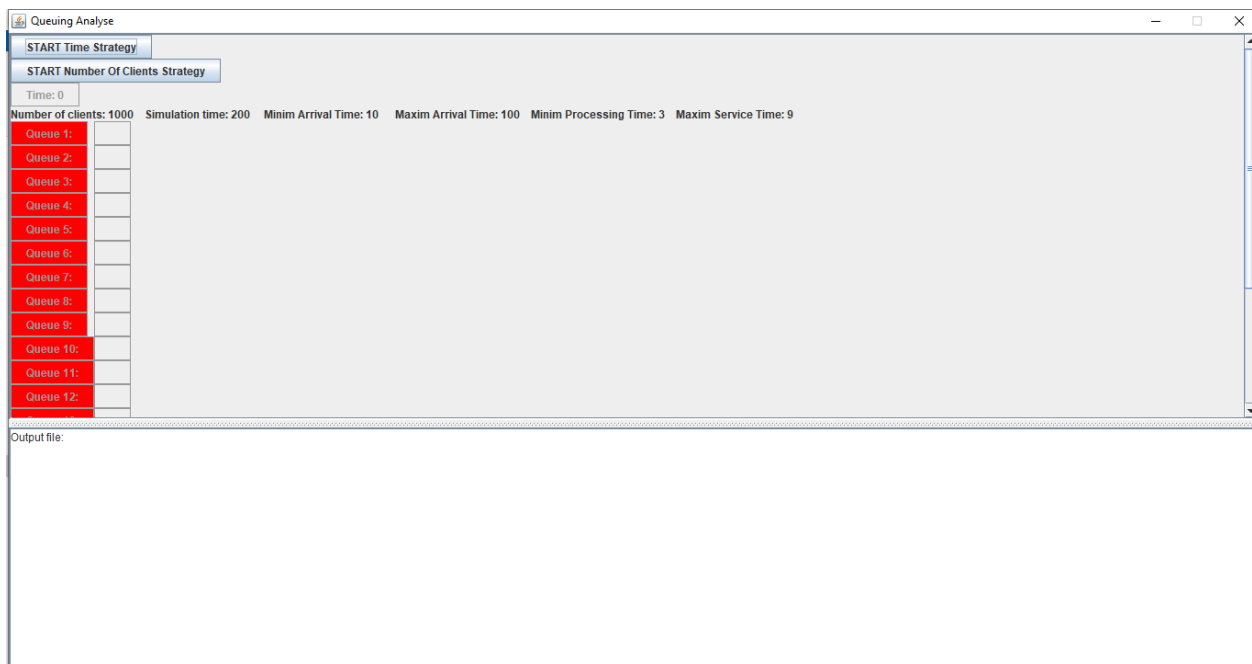


The screenshot shows a Windows command prompt window with the title bar "C:\Windows\System32\cmd.exe". The window contains the following text:

```
Microsoft Windows [Version 10.0.18363.720]  
(c) 2019 Microsoft Corporation. All rights reserved.  
  
C:\Users\cris\IdeaProjects\Assignment_2>java -jar PT2020_30227_Cristian_Berengeas_Assignment_2.jar in.txt out.txt
```

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 fereastră care contine informatiile citite din fisierul de intrare.



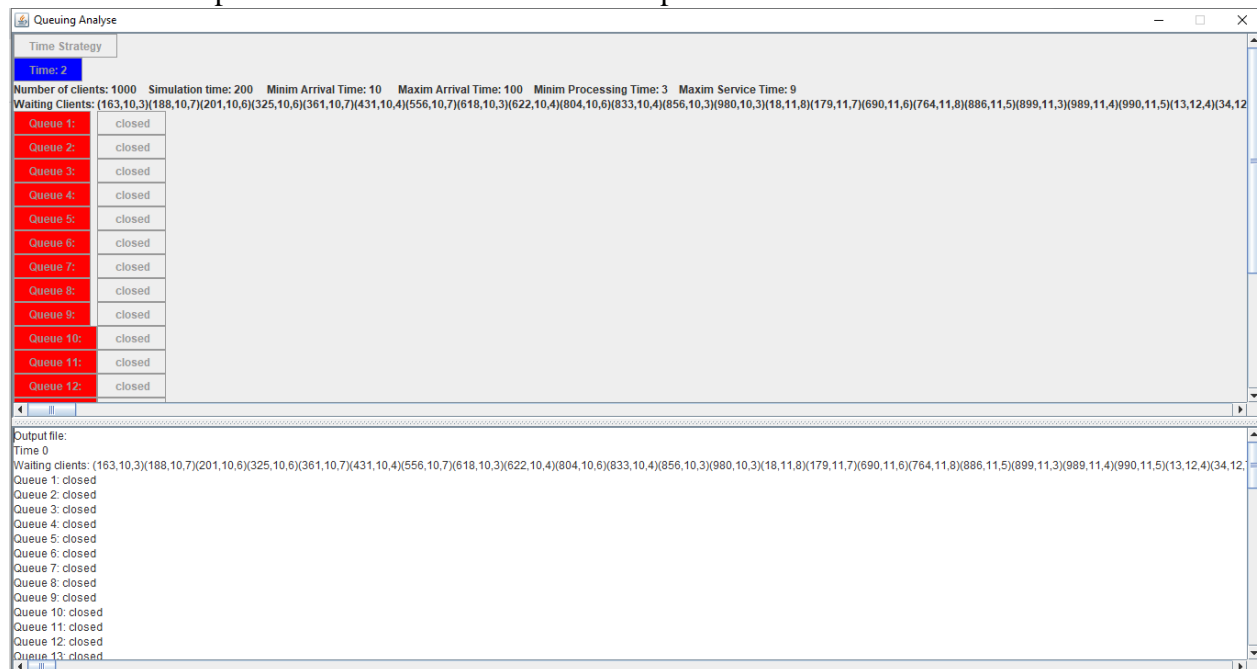
Mai este doar un pas pe care utilizatorul trebuie sa il parcurga pentru a incepe simularea, acesta trebuie sa aleaga ce strategie sa foloseasca strategia printr-un click pe unul dintre cele 2 butoane . Exista doua strategii posibile:

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 vizualizare in timp real a simularii.

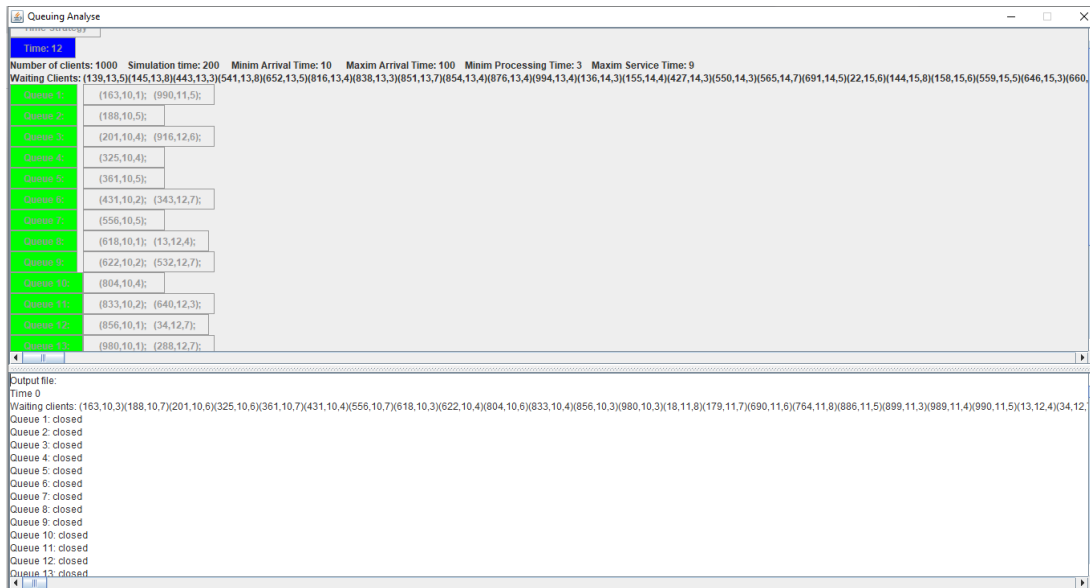
Dupa ce utilizatorul apasa un buton, alegand strategia, simularea incepe. Se afiseaza la fiecare moment de timp situatia actualizata a cozilor. Timul de servire este decrementat cu fiecare secunda pentru clientul care este in curs de procesare.



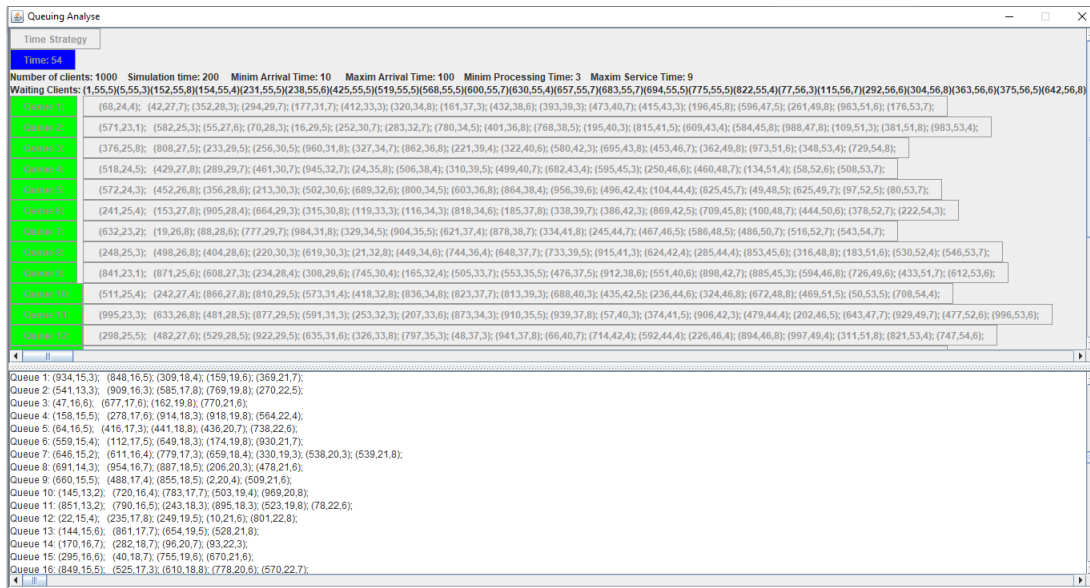
Atunci cand cozile nu contin clienti acestea sunt inchise. Culoarea butonului este rosie.

Daca in coada sunt clienti, angajatul proceseaza cererile clientilor.

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

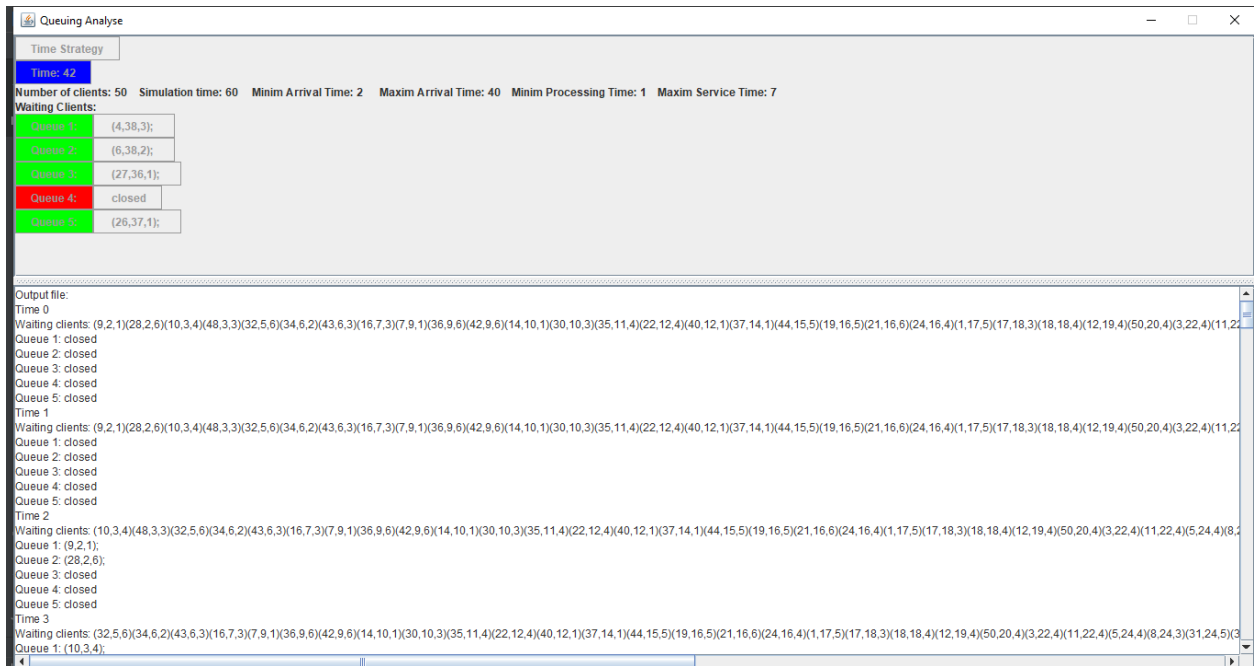


In partea de jos este afisat continutul disierului de iesire.



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 clientii 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

Queuing Analyse

Time Strategy

Time: 200

Average waiting time: 67.540504 s   Number of clients served: 679/1000

Number of clients: 1000   Simulation time: 200   Minim Arrival Time: 10   Maxim Arrival Time: 100   Minim Processing Time: 3   Maxim Service Time: 9

Waiting Clients:

Queue 1:

(4,71,1); (713,71,6); (167,73,7); (948,75,8); (554,78,4); (710,79,5); (700,81,5); (831,83,7); (19,86,7); (198,88,4); (73,90,7); (110,92,8); (750,94,6); (256,97,4); (190,98,8);

Queue 2:

(14,71,2); (843,71,5); (265,73,4); (722,74,7); (138,77,3); (351,78,3); (903,78,3); (937,79,5); (612,82,3); (402,83,5); (185,85,3); (94,86,7); (269,88,7); (324,90,8); (401,93,4); (804,94,6); (362,97,5); (504,98,6);

Queue 3:

(451,70,2); (895,71,6); (414,73,8); (672,76,8); (920,78,8); (655,82,4); (832,83,3); (652,84,8); (95,87,4); (590,88,3); (122,90,6); (640,91,5); (429,93,4); (994,94,3); (524,96,6); (950,97,8);

Queue 4:

(191,71,6); (923,72,3); (807,73,5); (24,76,4); (225,77,6); (993,78,6); (755,80,8); (113,84,5); (164,86,6); (808,87,4); (257,89,6); (67,91,5); (239,93,6); (716,95,4); (75,97,5); (245,98,4); (396,99,8);

Queue 5:

(913,70,3); (93,72,6); (864,73,6); (471,76,5); (242,78,8); (353,80,8); (839,83,6); (813,85,3); (409,86,5); (290,88,3); (260,89,7); (69,92,3); (51,93,5); (638,94,4); (549,96,8); (654,98,7);

Queue 6:

(609,69,1); (772,71,7); (721,73,4); (380,75,4); (678,76,8); (33,79,6); (403,81,4); (25,83,4); (186,84,7); (622,86,8); (312,89,4); (482,90,4); (278,92,6); (972,93,7); (136,97,8); (65,99,5);

Queue 7:

(680,71,7); (326,73,4); (2,75,8); (744,77,4); (819,78,6); (473,80,6); (596,83,8); (173,86,3); (42,87,3); (865,87,3); (966,88,7); (193,91,7); (597,93,7); (829,96,8); (703,98,7);

Queue 8:

(541,71,8); (428,74,4); (232,76,4); (250,77,3); (439,78,4); (701,79,4); (494,80,5); (83,83,7); (330,85,8); (867,87,4); (370,89,6); (410,91,3); (574,92,7); (200,95,8); (764,97,6); (461,99,5);

Queue 9:

(536,70,2); (939,71,7); (957,73,3); (558,75,4); (706,76,8); (311,79,8); (771,82,8); (266,85,7); (382,87,6); (602,89,8); (454,92,3); (317,93,8); (687,96,4); (518,97,5); (739,98,5);

Queue 10:

(435,71,3); (116,72,6); (958,73,6); (625,76,8); (821,78,8); (816,81,8); (769,84,7); (54,87,7); (709,89,6); (525,91,6); (467,93,7); (780,96,5); (837,97,8); (822,99,8);

Queue 11:

(181,70,3); (218,72,7); (437,74,3); (943,75,5); (457,77,3); (480,78,3); (487,79,6); (571,81,4); (171,83,7); (374,85,4); (708,86,8); (809,89,5); (812,90,4); (878,92,5); (997,93,4); (898,95,8); (325,98,3); (145,99,6);

Queue 12:

(908,69,2); (981,71,4); (975,72,8); (275,76,5); (801,77,5); (587,79,3); (215,80,4); (879,81,3); (301,83,4); (306,84,4); (870,85,3); (738,86,3); (592,87,3); (881,88,8); (530,91,3); (5,93,5); (139,94,8); (420,97,6); (884,98,7);

Queue 8: (541,71,2); (912,71,8); (428,74,4); (232,76,4); (250,77,3); (439,78,4); (701,79,4); (494,80,5); (83,83,7); (330,85,8); (867,87,4); (370,89,6); (410,91,3); (574,92,7); (200,95,8); (764,97,6); (461,99,5);

Queue 9: (536,70,2); (939,71,7); (957,73,3); (558,75,4); (706,76,8); (311,79,8); (771,82,8); (266,85,7); (382,87,6); (602,89,8); (454,92,3); (317,93,8); (687,96,4); (518,97,5); (739,98,5);

Queue 10: (435,71,3); (116,72,6); (958,73,6); (625,76,8); (821,78,8); (816,81,8); (769,84,7); (54,87,7); (709,89,6); (525,91,6); (467,93,7); (780,96,5); (837,97,8); (822,99,8);

Queue 11: (181,70,3); (218,72,7); (437,74,3); (943,75,5); (457,77,3); (480,78,3); (487,79,6); (571,81,4); (171,83,7); (374,85,4); (708,86,8); (809,89,5); (812,90,4); (878,92,5); (997,93,4); (898,95,8); (325,98,3); (145,99,6);

Queue 12: (908,69,2); (981,71,4); (975,72,8); (275,76,5); (801,77,5); (587,79,3); (215,80,4); (879,81,3); (301,83,4); (306,84,4); (870,85,3); (738,86,3); (592,87,3); (881,88,8); (530,91,3); (5,93,5); (139,94,8); (420,97,6); (884,98,7);

Queue 13: (512,71,5); (575,72,6); (114,75,3); (332,76,7); (531,78,7); (369,80,8); (857,83,8); (336,86,4); (624,87,6); (131,90,5); (547,91,4); (76,93,5); (647,94,8); (588,97,7); (490,99,6);

Queue 14: (930,70,5); (714,72,7); (567,75,7); (74,78,3); (712,78,7); (698,80,4); (825,82,6); (838,84,6); (751,86,5); (348,88,7); (508,90,6); (103,93,8); (176,96,7); (48,98,4); (214,99,8);

Queue 15: (512,71,5); (575,72,6); (114,75,3); (332,76,7); (531,78,7); (369,80,8); (857,83,8); (336,86,4); (624,87,6); (131,90,5); (547,91,4); (76,93,5); (647,94,8); (588,97,7); (490,99,6);

Queue 16: (385,70,1); (817,71,4); (844,72,5); (611,74,4); (366,76,5); (117,78,5); (614,79,7); (3,82,7); (519,84,3); (406,85,8); (58,88,4); (55,90,4); (666,90,8); (496,93,4); (291,95,5); (241,97,5); (463,98,7);

Queue 17: (542,71,6); (40,73,6); (586,75,7); (180,78,8); (230,80,3); (651,81,8); (552,84,4); (952,85,3); (1,87,4); (126,88,3); (212,89,8); (107,92,5); (584,93,5); (725,95,5); (464,97,7); (347,99,6);

Queue 18: (863,70,4); (501,72,8); (784,75,4); (781,76,4); (322,78,7); (343,80,8); (626,83,6); (802,85,7); (665,87,7); (162,90,3); (938,90,5); (182,93,3); (774,93,7); (28,97,3); (668,97,8); (650,99,4);

Queue 19: (889,70,3); (243,72,3); (87,73,3); (22,74,8); (785,76,6); (846,78,8); (223,82,3); (313,83,3); (995,83,7); (376,86,7); (924,88,5); (220,90,6); (39,93,8); (860,95,8); (123,98,7); (942,99,3);

Queue 20: (991,70,3); (399,72,6); (203,74,3); (929,75,6); (688,77,4); (740,78,8); (694,81,6); (858,83,7); (210,86,5); (682,87,4); (251,89,5); (735,90,4); (509,92,5); (965,93,3); (569,95,8); (855,97,8); (978,99,7);

Average waiting time: 67.540504 s   Number of clients served: 679/1000

Queuing Analyse

Time Strategy

Time: 45

Average waiting time: 4.48 s   Number of clients served: 50/50

Number of clients: 50   Simulation time: 60   Minim Arrival Time: 2   Maxim Arrival Time: 40   Minim Processing Time: 1   Maxim Service Time: 7

Waiting Clients:

Queue 1:

closed

Queue 2:

closed

Queue 3:

closed

Queue 4:

closed

Queue 5:

closed

Output file:

Time 0

Waiting clients: (9,2,1)(28,2,6)(10,3,4)(48,3,3)(32,5,6)(34,6,2)(43,6,3)(16,7,3)(7,9,1)(36,9,6)(42,9,6)(14,10,1)(30,10,3)(35,11,4)(22,12,4)(40,12,1)(37,14,1)(44,15,5)(19,16,5)(21,16,6)(24,16,4)(1,17,5)(17,18,3)(18,18,4)(12,19,4)(50,20,4)(3,22,4)(11,22,4)

Queue 1: closed

Queue 2: closed

Queue 3: closed

Queue 4: closed

Queue 5: closed

Time 1

Waiting clients: (9,2,1)(28,2,6)(10,3,4)(48,3,3)(32,5,6)(34,6,2)(43,6,3)(16,7,3)(7,9,1)(36,9,6)(42,9,6)(14,10,1)(30,10,3)(35,11,4)(22,12,4)(40,12,1)(37,14,1)(44,15,5)(19,16,5)(21,16,6)(24,16,4)(1,17,5)(17,18,3)(18,18,4)(12,19,4)(50,20,4)(3,22,4)(11,22,4)

Queue 1: closed

Queue 2: closed

Queue 3: closed

Queue 4: closed

Queue 5: closed

Time 2

Waiting clients: (10,3,4)(48,3,3)(32,5,6)(34,6,2)(43,6,3)(16,7,3)(7,9,1)(36,9,6)(42,9,6)(14,10,1)(30,10,3)(35,11,4)(22,12,4)(40,12,1)(37,14,1)(44,15,5)(19,16,5)(21,16,6)(24,16,4)(1,17,5)(17,18,3)(18,18,4)(12,19,4)(50,20,4)(3,22,4)(11,22,4)(5,24,4)(8,24,4)

Queue 1: (9,2,1);

Queue 2: (28,2,6);

Queue 3: closed

Queue 4: closed

Queue 5: closed

Time 3

Waiting clients: (32,5,6)(34,6,2)(43,6,3)(16,7,3)(7,9,1)(36,9,6)(42,9,6)(14,10,1)(30,10,3)(35,11,4)(22,12,4)(40,12,1)(37,14,1)(44,15,5)(19,16,5)(21,16,6)(24,16,4)(1,17,5)(17,18,3)(18,18,4)(12,19,4)(50,20,4)(3,22,4)(11,22,4)(5,24,4)(8,24,3)(31,24,5)(32,24,5)

Queue 1: (10,3,4);

# ***Proiectare***

## ***Proiectare clase***

Am proiectat clasele / interfetele:

- Client - inglobeaza informatii reprezentative pentru client
- Server - 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
- TimeStrategy - 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 functie 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 simularii si face un management al cozilor cu ajutorul clasei scheduler
- Ui - acesta clasa contine implementarea interfetei grafice



