



Documentatie

Aplicatie de simulare

Bogdan Cristian Sipos
GRUPA 30225

Cuprins

1. Obiectivul temei.	2
2. Analiza problemei, modelare, scenarii, cazuri de utilizare.....	2
3. Proiectare.....	4
4. Implementare.....	5
5. Rezultate.....	9
6. Concluzii.	534

1. Obiectivul temei

Obiectivul temei a fost sa proiectam in Java un simulator de cozi. Acest simulator primeste ca parametrii numarul de cozi disponibile, numarul de clienti, timpul maxim a simularii, timpul minim si maxim in care pot veni clientii, timpul minim si maxim de procesare a unui client. Va trebui clientii sa fie generati de fiecare data si de fiecare data cand o coada este goala va trebui sa fie inchisa si respectiv deschisa cand e cazul. De asemenea rezultatul simularii, timpul mediu de asteptare, timpul mediu de procesare a clientilor si ora de varf a simularii trebuie sa fie afisat atat in fisier cat si in interfata grafica. Simularea trebuie sa fie realizata cu ajutorul a celor doua strategii: cel mai scurt timp si cea mai scurta coada.

Obiectiv secundar	Descriere	Capitol
Dezvoltare de use case-uri si scenarii	A fost un ajutor in a identifica necesitatile problemei si sa ilustrem interactiunea dintre utilizator si sistem	2
Alegerea structurilor de date	Alegerea anumitor structuri de date a avut scopul de a ne eficientiza programul sau de a ne simplifica metoda de implementare	3
Impartirea pe clase	Impartirea pe clase ne ajuta in: organizarea structurii codului, reutilizarea acestuia, incapsularea si ascunderea detaliilor precum si in partea de intretinere si extindere	3
Dezvoltarea algoritmilor	Dezvoltarea algoritmilor a fost facuta prin implementarea celor doua tipuri de departajare a clientilor	3
Implementarea solutiei	In implementare am divizat fiecare problema in mai multe subprobleme (in mai multe clase, subclase dar si metode) dar am si realizat o interfata grafica cat mai simpla de utilizat	4

2. Analiza problemei, modelare, scenarii, cazuri de utilizare

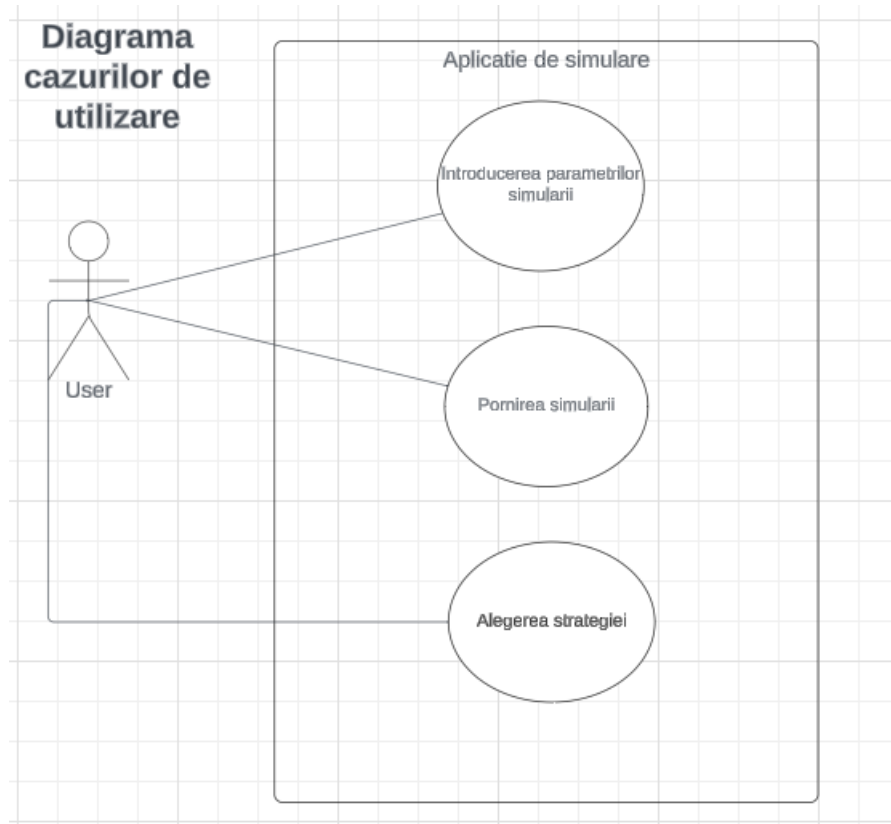
Cerintele functionale ale aplicatiei de simulare sunt:

- Aplicatia va trebui sa isi lase utilizatorii sa isi seteze parametrii simularii
- Aplicatia va trebui sa isi lase utilizatorii sa o porneasca
- Aplicatia va trebui sa afiseze rezultatul final intr-un fisier si interfata grafica

- Aplicatia trebuie sa permita alegerea uneia dintre metode de partajare a clientilor

Cerintele nefunctionale ale aplicatiei sunt:

- Aplicatia trebuie sa fie intuitiva si usor de folosit de utilizator



Use case: setarea simularii

Actor ptincipal: Utilizatorul

Scenariu principal:

1. Utilizatorul introduce valorile pentru: numarul de clienti, numarul de cozi, intervalul simularii, timpul minim si maxim cand a ajuns la coada si timpul minim si maxim de procesare
2. Utilizator alege strategia si incepe simularea

Scenarii alternative:

- Utilizatorul introduce date invalide la parametrii asa ca nu va porni simularea si ne intoarcem la scenariul 1

Modelarea problemei a fost facuta in mare parte dupa exemplul alaturat prezentarii temei, astfel incat cozile sunt modelate ca servere, care primesc task-uri (clienti) pe care trebuie sa le proceseze. Serverele sunt monitorizate si primesc task-uri de la un scheduler, care consulta timpul de asteptare la fiecare server in parte si ia o decizie cu privire la coada careia sa ii asigneze urmatorul client pentru a facilita eficienta. De asemenea scheduler-ul este responsabil si pentru inchiderea si deschiderea serverelor atunci cand este nevoie. Datele de

intrare sunt numarul de clienti, numarul de servere, durata maxima a simularii, timpul minim si maxim cand au ajuns clientii la servere, timpul minim si maxim a duratei serviciului.

3. Proiectare

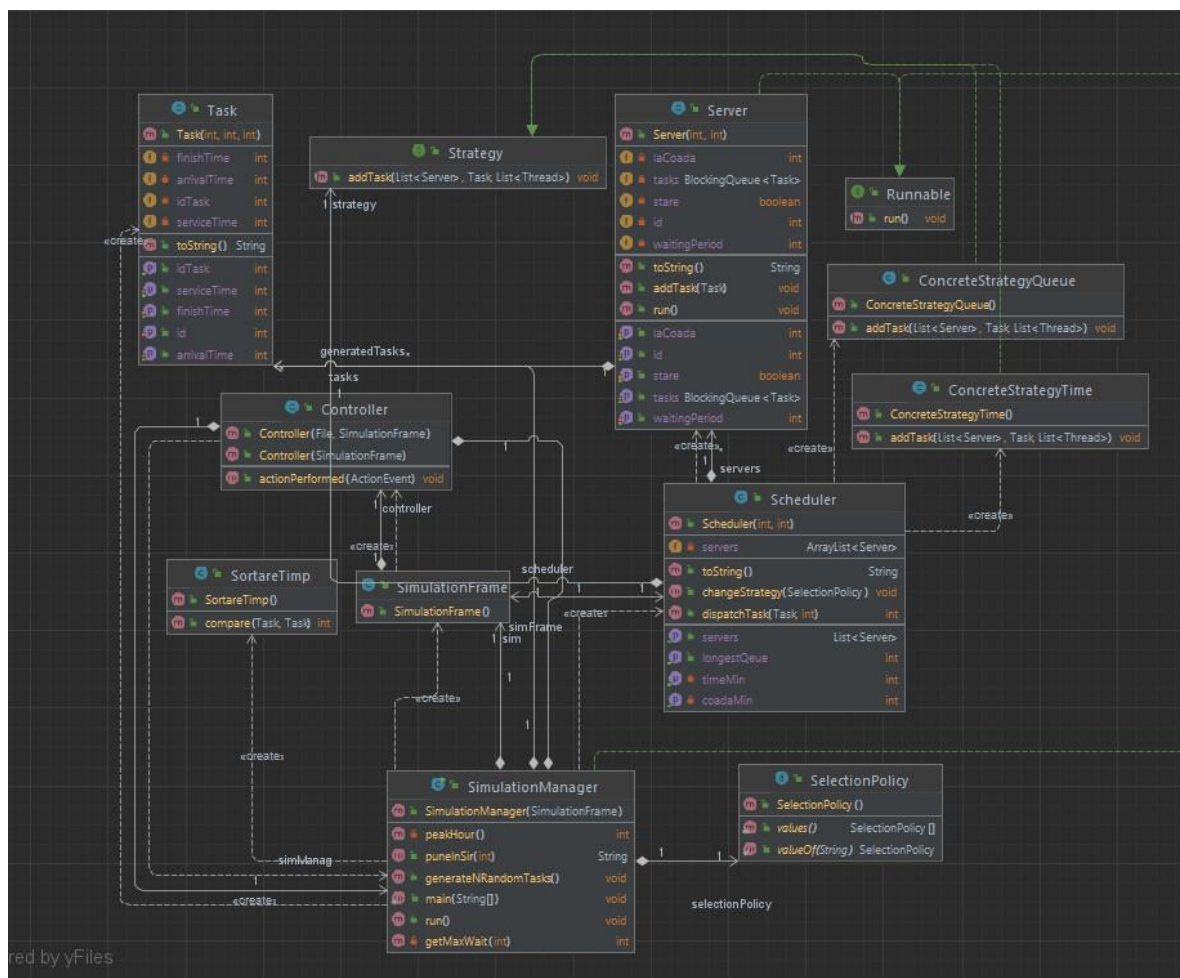
In conceperea proiectului am folosit in mare parte structura pe care am primit-o ca indrumare dar am si adaugat de la mine anumite clase si metode.

Astfel avem urmatoarele clase: Task care reprezinta clientul si actiunea pe care o face, Server care reprezinta coada la care se afla clientii, Scheduler care are rolul de a controla serverele si de a le imparti task-urile, clasa SortareTimp care ne ajuta sa sortam clientii in functie de timpul cand au ajuns, SimulationFrame care se ocupa de partea grafica a programului, Controller care se ocupa de partea interactiva a interfetei grafice si SimulationManager care foloseste toate clasele pentru a avea simularea dorita.

Am folosit o coada de tip BlockingQueue deoarece aceasta ofera un mecanism simplu si sigur pentru sincronizarea firelor de executie, acestea pot astepta pana cand sunt disponibile datele in coada fara a fie nevoie de alte mecanisme de sincronizare complexe, cum ar fi lacate, semafoare sau variabile conditionale.

Am implementat cele doua variante de distribuire a clientilor in servere in functie de cel mai scurt timp de procesare sau cea mai scurta coada de clienti. In ambele cazuri am sortat crescator clientii in functie de timpul in care au ajuns si apoi am vazut care server are cel mai scurt timp de procesare respectiv cea mai scurta coada.

4.



4. Implementare

Clasa Task contine:

a) Atribute:

- arrivalTime: Un întreg care reprezintă momentul de timp când sarcina ajunge pentru procesare.
- serviceTime: Un întreg care indică durata necesară pentru a procesa această sarcină.
- finishTime: Un întreg care indică momentul de timp când sarcina este finalizată. Este setat ulterior, după procesare.
- idTask: Un întreg care reprezintă identificatorul unic al sarcinii.

b) Constructor:

- public Task(int arrivalTime, int serviceTime, int id): Constructorul clasei Task care primește trei parametri: arrivalTime (timpul de sosire), serviceTime (timpul de procesare) și id (identificatorul sarcinii). Inițializează câmpurile clasei cu valorile primite.

c) Metode:

- getFinishTime(): Returnează momentul de timp când sarcina este finalizată.
- setFinishTime(int finishTime): Setează momentul de timp când sarcina este finalizată.
- getIdTask(): Returnează identificatorul unic al sarcinii.
- setId(int idTask): Setează identificatorul unic al sarcinii.
- getArrivalTime(): Returnează timpul de sosire al sarcinii.
- setArrivalTime(int arrivalTime): Setează timpul de sosire al sarcinii.
- getServiceTime(): Returnează timpul de procesare al sarcinii.
- setServiceTime(int serviceTime): Setează timpul de procesare al sarcinii.
- toString(): Suprascrie metoda toString() pentru a oferi o reprezentare sub formă de șir de caractere a sarcinii, care include identificatorul, timpul de sosire și timpul de procesare.

Clasa Server contine:

a) Atribute:

- tasks: O coadă blocantă (BlockingQueue) de sarcini (Task) ce urmează a fi procesate de server.
- waitingPeriod: Un întreg care reprezintă perioada totală de așteptare a sarcinilor în coadă.
- waitingTasks: Un întreg care reprezintă numărul de sarcini care așteaptă în coadă.
- laCoadă: Un întreg care reprezintă numărul de sarcini în prezent în coadă.
- stare: Un boolean care indică dacă serverul este activ sau inactiv.
- id: Un întreg care reprezintă identificatorul unic al serverului.

b) Constructor:

- public Server(int maxload, int id): Constructorul clasei Server care primește două parametri: maxload (încărcătura maximă a serverului) și id (identificatorul serverului). Inițializează coada de sarcini și alte atribute ale serverului.

c) Metode:

- addTask(Task newTask): Adaugă o nouă sarcină în coadă. Actualizează perioada totală de așteptare și numărul de sarcini în coadă.

- `toString()`: Suprascrie metoda `toString()` pentru a oferi o reprezentare sub formă de șir de caractere a serverului, care include starea sa (deschis sau închis) și sarcinile aflate în coadă.
- `run()`: Implementează metoda `run()` din interfața `Runnable`, care reprezintă logica de procesare a sarcinilor de către server. Serverul rulează într-o buclă infinită și procesează sarcinile din coadă până când aceasta este golită. Pentru fiecare sarcină, serverul așteaptă o secundă (simulând procesarea) și actualizează timpul rămas pentru procesare. Dacă timpul de procesare pentru o sarcină devine zero, aceasta este eliminată din coadă.

Clasa Scheduler conține:

- Atribute:**
 - `servers`: O listă de obiecte `Server` care reprezintă serverele disponibile pentru procesare.
 - `threads`: O listă de obiecte `Thread` asociate fiecărui server. Aceste fire de execuție rulează serverele în mod concurent.
 - `maxNoServers`: Un întreg care reprezintă numărul maxim de servere pe care le poate gestiona planificatorul.
 - `policy`: Un întreg care indică politica de selecție a serverului (0 pentru coada cea mai scurtă, 1 pentru timpul cel mai scurt).
 - `maxTasksPerServer`: Un întreg care reprezintă numărul maxim de sarcini pe care un server le poate procesa în același timp.
 - `strategy`: Un obiect de tipul `Strategy` care definește strategia de selecție a serverului pentru atribuirea sarcinilor.
- Constructor:**
 - `public Scheduler(int maxNoServers, int maxTasksPerServer)`: Constructorul clasei `Scheduler` care primește două parametri: `maxNoServers` (numărul maxim de servere) și `maxTasksPerServer` (numărul maxim de sarcini per server). Inițializează listele de servere și fire de execuție și creează obiecte `Server` și `Thread` corespunzătoare pentru fiecare server.
- Metode:**
 - `changeStrategy(SelectionPolicy policy)`: Schimbă strategia de selecție a serverului în funcție de politica specificată. Creează un obiect de strategie corespunzător politicii.
 - `getTimeMin()`: Returnează indexul serverului cu cea mai scurtă perioadă de așteptare.
 - `getCodaMin()`: Returnează indexul serverului cu cea mai mică coadă de așteptare (numărul minim de sarcini în așteptare).
 - `dispatchTask(Task t, int policy)`: Distribuie o sarcină către un server disponibil în funcție de politica specificată. Actualizează starea serverului și porneste un fir de execuție nou dacă este necesar.
 - `getServers()`: Returnează lista de servere.
 - `getLongestQueue()`: Returnează lungimea maximă a cozii de așteptare din toate serverele.
 - `toString()`: Suprascrie metoda `toString()` pentru a oferi o reprezentare sub formă de șir de caractere a stării serverelor și cozilor de așteptare.

Enum **SelectionPolicy**:

Acest enum Java, **SelectionPolicy**, definește două politici posibile de selecție a serverului: **SHORTEST_QUEUE** și **SHORTEST_TIME**. Iată o scurtă descriere a fiecărei politici:

- **SHORTEST_QUEUE**: Această politică indică faptul că serverul selectat pentru a primi o sarcină este cel cu cea mai mică coadă de așteptare. Cu alte cuvinte, sarcina este atribuită serverului care are cea mai mică cantitate de sarcini deja în așteptare.
- **SHORTEST_TIME**: Această politică indică faptul că serverul selectat pentru a primi o sarcină este cel cu cea mai mică perioadă de așteptare totală. Acest lucru înseamnă că sarcina este atribuită serverului care va termina sarcinile existente cel mai rapid, luând în considerare și sarcina nouă.

Clasa SortareTimp contine:

- a) Metoda `compare(Task x, Task y)`:

Această metodă este suprascrisă din interfața **Comparator** și este folosită pentru a compara două obiecte **Task**, `x` și `y`, pe baza timpului lor de sosire. Metoda returnează un număr întreg negativ, zero sau pozitiv în funcție de relația dintre timpurile de sosire ale celor două sarcini:

- Returnează un număr negativ dacă `x` are un timp de sosire mai mic decât `y`.
- Returnează zero dacă timpul de sosire al lui `x` este egal cu timpul de sosire al lui `y`.
- Returnează un număr pozitiv dacă `x` are un timp de sosire mai mare decât `y`.

Clasa SimulationManager contine:

- a) Atribute:

- `sim`: O referință către obiectul **SimulationFrame**, care reprezintă fereastra principală a aplicației GUI.
- Variabile pentru parametrii simulării, cum ar fi `timeLimit`, `maxProcessingTime`, `minProcessingTime`, `maxArrivalTime`, `minArrivalTime`, `numberOfServers`, `numberOfClients`, `simulationTime`, `averageTime`, `peakHour`, `averageServiceTime`, `policy`.
- `scheduler`: Un obiect de tipul **Scheduler** pentru gestionarea serverelor și a sarcinilor.
- `fisierOut`: Un obiect **File** pentru a specifica fișierul de ieșire pentru salvarea rezultatelor simulării.
- `generatedTasks`: O listă de obiecte **Task** reprezentând sarcinile generate pentru simulare.

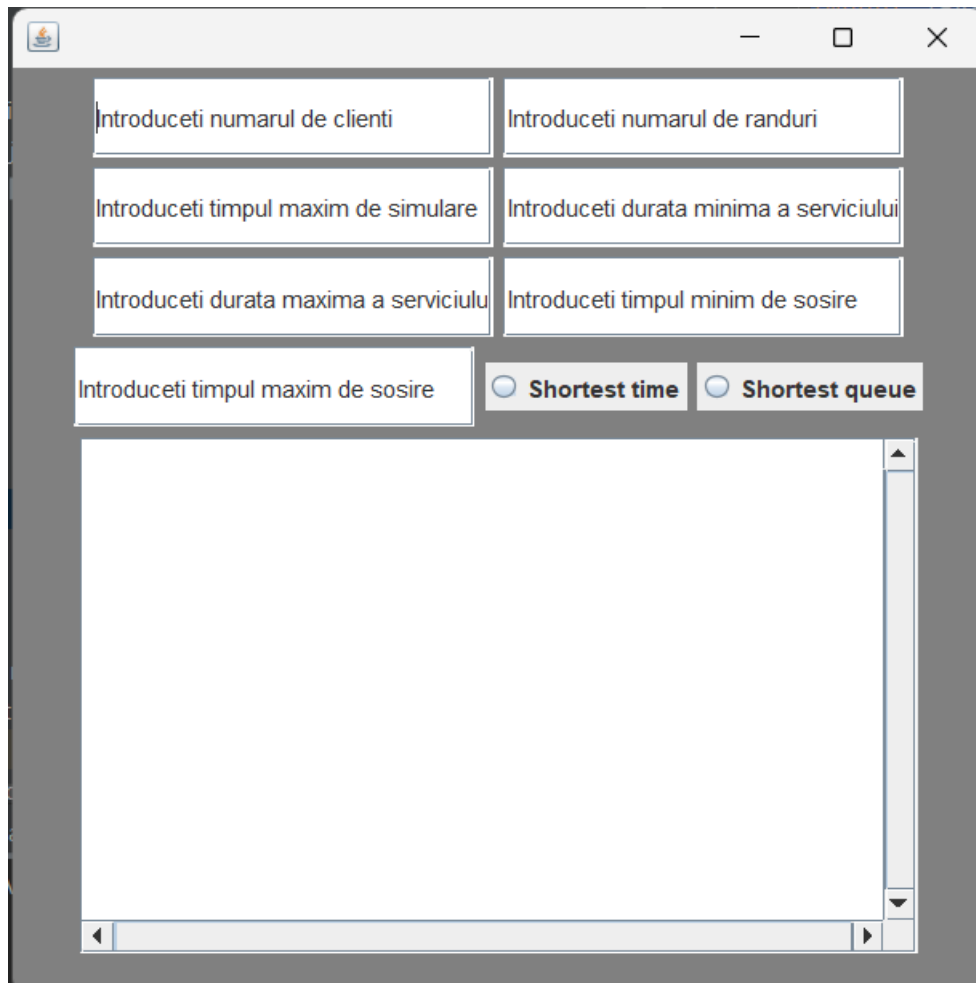
- b) Constructor:

- `public SimulationManager(SimulationFrame sim)`: Constructorul clasei **SimulationManager** care primește o referință către **SimulationFrame**, inițializează parametrii simulării și creează lista de sarcini generate.

- c) Metode:

- `generateNRandomTasks()`: Generează un număr specificat de sarcini cu timpi de sosire și de serviciu aleatorii și le adaugă în lista `generatedTasks`.

- `run()`: Metoda care gestionează desfășurarea simulării. În bucla `while`, se actualizează starea simulării în fiecare secundă și se adaugă sarcinile în cozi, se actualizează starea coziilor, se scriu rezultatele în fișier și se actualizează afișajul în fereastra GUI.
- `puneInSir(int currentTime)`: Creează un șir de caractere care conține starea simulării la un moment dat pentru a fi afișat în fereastra GUI.
- Metoda `main`: Metoda principală care creează un obiect `SimulationFrame` pentru a iniția aplicația GUI.



Clasa `SimulationFrame` conține:

- a) **Atribute:**
 - `panel`: Un panou care conține elementele GUI pentru introducerea parametrilor simulării.
 - `controller`: Un obiect de tipul `Controller`, utilizat pentru gestionarea evenimentelor GUI.
 - `nrCientiPanel`, `nrRanduriPanel`, `maxSimTimePanel`, `minSerPanel`, `maxSerPanel`, `minTimeArrPanel`, `maxTimeArrPanel`: Câmpuri de text pentru introducerea diferitelor parametri ai simulării, cum ar fi numărul de clienți, numărul de rânduri, timpul maxim de simulare etc.

- ShortestTime, ShortestQueue: Butonii radio pentru selectarea politicii de selecție a serverului.
 - afisareRez: Un obiect JTextArea pentru afișarea rezultatelor simulării.
 - scrollPane: Un obiect JScrollPane pentru a permite derularea conținutului afisareRez în cazul în care rezultatele depășesc dimensiunile ferestrei.
 - fisierOut: Un obiect File pentru a specifica fișierul de ieșire pentru salvarea rezultatelor simulării.
- b) Constructor:
- public SimulationFrame(): Constructorul clasei SimulationFrame care inițializează toate elementele GUI, adaugă evenimente și setează aspectul și dimensiunea ferestrei.

Clasa Controller conține:

- a) Atribute:
- simFrame: O referință către obiectul SimulationFrame, care reprezintă fereastra principală a aplicației GUI.
 - simManag: O referință către obiectul SimulationManager, care gestionează simularea sistemului de cozi.
 - fisierOut: Un obiect File pentru a specifica fișierul de ieșire pentru salvarea rezultatelor simulării.
- b) Constructori:
- public Controller(SimulationFrame simFrame): Constructor care primește o referință către SimulationFrame și o utilizează pentru a inițializa atributul simFrame.
 - public Controller(File fisierOut, SimulationFrame simFrame): Constructor care primește un obiect File și un SimulationFrame și le utilizează pentru a inițializa atribuțiile fisierOut și simFrame.
 - Metoda actionPerformed(ActionEvent e): Această metodă este suprascrisă din interfața ActionListener și este apelată atunci când se declanșează un eveniment de acțiune, cum ar fi apăsarea unui buton. Verifică sursa evenimentului și execută acțiuni corespunzătoare:

Dacă sursa evenimentului este butonul ShortestTime din SimulationFrame, setează politica de selecție a serverului pe SHORTEST_TIME și inițiază o nouă instanță de SimulationManager și un fir de execuție pentru a începe simularea.

Dacă sursa evenimentului este butonul ShortestQueue din SimulationFrame, setează politica de selecție a serverului pe SHORTEST_QUEUE și inițiază o nouă instanță de SimulationManager și un fir de execuție pentru a începe simularea.

5. Rezultate

Test 1	Test 2	Test 3
$N = 4$ $Q = 2$ $t_{simulation}^{MAX} = 60 \text{ seconds}$ $[t_{arrival}^{MIN}, t_{arrival}^{MAX}] = [2, 30]$ $[t_{service}^{MIN}, t_{service}^{MAX}] = [2, 4]$	$N = 50$ $Q = 5$ $t_{simulation}^{MAX} = 60 \text{ seconds}$ $[t_{arrival}^{MIN}, t_{arrival}^{MAX}] = [2, 40]$ $[t_{service}^{MIN}, t_{service}^{MAX}] = [1, 7]$	$N = 1000$ $Q = 20$ $t_{simulation}^{MAX} = 200 \text{ seconds}$ $[t_{arrival}^{MIN}, t_{arrival}^{MAX}] = [10, 100]$ $[t_{service}^{MIN}, t_{service}^{MAX}] = [3, 9]$

Rezultatul pentru testul unu este:

Time: 0

Waiting clients: (3,3,8)(2,6,5)(0,16,2)(1,18,2)

Queue 0: closed

Queue 1: closed

Time: 1

Waiting clients: (3,3,8)(2,6,5)(0,16,2)(1,18,2)

Queue 0: closed

Queue 1: closed

Time: 2

Waiting clients: (3,3,8)(2,6,5)(0,16,2)(1,18,2)

Queue 0: closed

Queue 1: closed

Time: 3

Waiting clients: (2,6,5)(0,16,2)(1,18,2)

Queue 0: [(3,3,8)]

Queue 1: closed

Time: 4

Waiting clients: (2,6,5)(0,16,2)(1,18,2)

Queue 0: [(3,3,7)]

Queue 1: closed

Time: 5

Waiting clients: (2,6,5)(0,16,2)(1,18,2)

Queue 0: [(3,3,6)]

Queue 1: closed

Time: 6

Waiting clients: (0,16,2)(1,18,2)

Queue 0: [(3,3,5)]

Queue 1: [(2,6,5)]

Time: 7

Waiting clients: (0,16,2)(1,18,2)

Queue 0: [(3,3,4)]

Queue 1: [(2,6,4)]

Time: 8

Waiting clients: (0,16,2)(1,18,2)

Queue 0: [(3,3,3)]

Queue 1: [(2,6,3)]

Time: 9

Waiting clients: (0,16,2)(1,18,2)

Queue 0: [(3,3,2)]

Queue 1: [(2,6,2)]

Time: 10

Waiting clients: (0,16,2)(1,18,2)

Queue 0: [(3,3,1)]

Queue 1: [(2,6,1)]

Time: 11

Waiting clients: (0,16,2)(1,18,2)

Queue 0: closed

Queue 1: closed

Time: 12

Waiting clients: (0,16,2)(1,18,2)

Queue 0: closed

Queue 1: closed

Time: 13

Waiting clients: (0,16,2)(1,18,2)

Queue 0: closed

Queue 1: closed

Time: 14

Waiting clients: (0,16,2)(1,18,2)

Queue 0: closed

Queue 1: closed

Time: 15

Waiting clients: (0,16,2)(1,18,2)

Queue 0: closed

Queue 1: closed

Time: 16

Waiting clients: (1,18,2)

Queue 0: [(0,16,2)]

Queue 1: closed

Time: 17

Waiting clients: (1,18,2)

Queue 0: [(0,16,1)]

Queue 1: closed

Time: 18

Waiting clients:

Queue 0: closed

Queue 1: [(1,18,2)]

Time: 19

Waiting clients:

Queue 0: closed

Queue 1: [(1,18,1)]

Time: 20

Waiting clients:

Queue 0: closed

Queue 1: closed

Average waiting time: 0.0

Peak hour: 6

Average service time: 4.25

Rezultatul pentru testul 2 este:

Time: 0

Waiting clients:

(49,2,4)(18,3,1)(43,3,3)(7,5,5)(10,6,8)(33,7,3)(36,7,6)(47,7,2)(13,8,3)(29,8,3)(40,8,5)(8,9,8)(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: closed

Queue 1: closed

Queue 2: closed

Queue 3: closed

Queue 4: closed

Time: 1

Waiting clients:

(49,2,4)(18,3,1)(43,3,3)(7,5,5)(10,6,8)(33,7,3)(36,7,6)(47,7,2)(13,8,3)(29,8,3)(40,8,5)(8,9,8)(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: closed

Queue 1: closed

Queue 2: closed

Queue 3: closed

Queue 4: closed

Time: 2

Waiting clients:

(18,3,1)(43,3,3)(7,5,5)(10,6,8)(33,7,3)(36,7,6)(47,7,2)(13,8,3)(29,8,3)(40,8,5)(8,9,8)(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(49,2,4)]

Queue 1: closed

Queue 2: closed

Queue 3: closed

Queue 4: closed

Time: 3

Waiting clients:

(7,5,5)(10,6,8)(33,7,3)(36,7,6)(47,7,2)(13,8,3)(29,8,3)(40,8,5)(8,9,8)(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(2

5,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(49,2,3)]

Queue 1: [(18,3,1)]

Queue 2: [(43,3,3)]

Queue 3: closed

Queue 4: closed

Time: 4

Waiting clients:

(7,5,5)(10,6,8)(33,7,3)(36,7,6)(47,7,2)(13,8,3)(29,8,3)(40,8,5)(8,9,8)(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(49,2,2)]

Queue 1: closed

Queue 2: [(43,3,2)]

Queue 3: closed

Queue 4: closed

Time: 5

Waiting clients:

(10,6,8)(33,7,3)(36,7,6)(47,7,2)(13,8,3)(29,8,3)(40,8,5)(8,9,8)(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(49,2,1)]

Queue 1: [(7,5,5)]

Queue 2: [(43,3,1)]

Queue 3: closed

Queue 4: closed

Time: 6

Waiting clients:

(33,7,3)(36,7,6)(47,7,2)(13,8,3)(29,8,3)(40,8,5)(8,9,8)(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(10,6,8)]

Queue 1: [(7,5,4)]

Queue 2: closed

Queue 3: closed

Queue 4: closed

Time: 7

Waiting clients:

(13,8,3)(29,8,3)(40,8,5)(8,9,8)(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(10,6,7)]

Queue 1: [(7,5,3)]

Queue 2: [(33,7,3)]

Queue 3: [(36,7,6)]

Queue 4: [(47,7,2)]

Time: 8

Waiting clients:

(8,9,8)(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(10,6,6), (13,8,3)]

Queue 1: [(7,5,2), (29,8,3)]

Queue 2: [(33,7,2), (40,8,5)]

Queue 3: [(36,7,5)]

Queue 4: [(47,7,1)]

Time: 9

Waiting clients:

(45,10,8)(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(10,6,5), (13,8,3)]

Queue 1: [(7,5,1), (29,8,3)]

Queue 2: [(33,7,1), (40,8,5)]

Queue 3: [(36,7,4)]

Queue 4: [(8,9,8)]

Time: 10

Waiting clients:

(31,11,6)(46,11,4)(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(10,6,4), (13,8,3)]

Queue 1: [(29,8,3)]

Queue 2: [(40,8,5), (45,10,8)]

Queue 3: [(36,7,3)]

Queue 4: [(8,9,7)]

Time: 11

Waiting clients:

(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,

5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(10,6,3), (13,8,3)]

Queue 1: [(29,8,2), (31,11,6)]

Queue 2: [(40,8,4), (45,10,8)]

Queue 3: [(36,7,2), (46,11,4)]

Queue 4: [(8,9,6)]

Time: 12

Waiting clients:

(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(10,6,2), (13,8,3)]

Queue 1: [(29,8,1), (31,11,6)]

Queue 2: [(40,8,3), (45,10,8)]

Queue 3: [(36,7,1), (46,11,4)]

Queue 4: [(8,9,5)]

Time: 13

Waiting clients:

(22,14,3)(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(10,6,1), (13,8,3)]

Queue 1: [(31,11,6)]

Queue 2: [(40,8,2), (45,10,8)]

Queue 3: [(46,11,4)]

Queue 4: [(8,9,4)]

Time: 14

Waiting clients:

(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: closed

Queue 1: [(31,11,5), (22,14,3)]

Queue 2: [(40,8,1), (45,10,8)]

Queue 3: [(46,11,3)]

Queue 4: [(8,9,3)]

Time: 15

Waiting clients:

(26,16,6)(38,16,4)(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(13,8,2)]

Queue 1: [(31,11,4), (22,14,3)]

Queue 2: [(45,10,8)]

Queue 3: [(46,11,2)]

Queue 4: [(8,9,2)]

Time: 16

Waiting clients:

(3,17,2)(11,17,6)(15,17,9)(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(13,8,1), (26,16,6)]

Queue 1: [(31,11,3), (22,14,3)]

Queue 2: [(45,10,7), (38,16,4)]

Queue 3: [(46,11,1)]

Queue 4: [(8,9,1)]

Time: 17

Waiting clients:

(16,18,2)(27,18,8)(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(4,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(26,16,6), (15,17,9)]

Queue 1: [(31,11,2), (22,14,3)]

Queue 2: [(45,10,6), (38,16,4)]

Queue 3: [(3,17,2)]

Queue 4: [(11,17,6)]

Time: 18

Waiting clients:

(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(26,16,5), (15,17,9)]

Queue 1: [(31,11,1), (22,14,3)]

Queue 2: [(45,10,5), (38,16,4)]

Queue 3: [(3,17,1), (16,18,2)]

Queue 4: [(11,17,5), (27,18,8)]

Time: 19

Waiting clients:

(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(26,16,4), (15,17,9)]

Queue 1: [(22,14,3)]

Queue 2: [(45,10,4), (38,16,4)]

Queue 3: [(16,18,2)]

Queue 4: [(11,17,4), (27,18,8)]

Time: 20

Waiting clients:

(4,21,10)(19,21,9)(25,21,4)(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(26,16,3), (15,17,9)]

Queue 1: [(22,14,2)]

Queue 2: [(45,10,3), (38,16,4)]

Queue 3: [(16,18,1)]

Queue 4: [(11,17,3), (27,18,8)]

Time: 21

Waiting clients:

(41,22,10)(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(26,16,2), (15,17,9)]

Queue 1: [(22,14,1), (4,21,10)]

Queue 2: [(45,10,2), (38,16,4)]

Queue 3: [(19,21,9), (25,21,4)]

Queue 4: [(11,17,2), (27,18,8)]

Time: 22

Waiting clients:

(1,23,3)(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(26,16,1), (15,17,9)]

Queue 1: [(4,21,10), (41,22,10)]

Queue 2: [(45,10,1), (38,16,4)]

Queue 3: [(19,21,8), (25,21,4)]

Queue 4: [(11,17,1), (27,18,8)]

Time: 23

Waiting clients:

(39,24,3)(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(15,17,9), (1,23,3)]

Queue 1: [(4,21,9), (41,22,10)]

Queue 2: [(38,16,4)]

Queue 3: [(19,21,7), (25,21,4)]

Queue 4: [(27,18,8)]

Time: 24

Waiting clients:

(12,25,6)(17,25,3)(44,25,3)(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(15,17,8), (1,23,3)]

Queue 1: [(4,21,8), (41,22,10)]

Queue 2: [(38,16,3), (39,24,3)]

Queue 3: [(19,21,6), (25,21,4)]

Queue 4: [(27,18,7)]

Time: 25

Waiting clients:

(23,26,6)(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(15,17,7), (1,23,3), (17,25,3)]

Queue 1: [(4,21,7), (41,22,10), (44,25,3)]

Queue 2: [(38,16,2), (39,24,3)]

Queue 3: [(19,21,5), (25,21,4)]

Queue 4: [(27,18,6), (12,25,6)]

Time: 26

Waiting clients:

(14,27,3)(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(15,17,6), (1,23,3), (17,25,3)]

Queue 1: [(4,21,6), (41,22,10), (44,25,3)]

Queue 2: [(38,16,1), (39,24,3), (23,26,6)]

Queue 3: [(19,21,4), (25,21,4)]

Queue 4: [(27,18,5), (12,25,6)]

Time: 27

Waiting clients:

(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(15,17,5), (1,23,3), (17,25,3)]

Queue 1: [(4,21,5), (41,22,10), (44,25,3)]

Queue 2: [(39,24,3), (23,26,6), (14,27,3)]

Queue 3: [(19,21,3), (25,21,4)]

Queue 4: [(27,18,4), (12,25,6)]

Time: 28

Waiting clients:

(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(15,17,4), (1,23,3), (17,25,3)]

Queue 1: [(4,21,4), (41,22,10), (44,25,3)]

Queue 2: [(39,24,2), (23,26,6), (14,27,3)]

Queue 3: [(19,21,2), (25,21,4)]

Queue 4: [(27,18,3), (12,25,6)]

Time: 29

Waiting clients:

(9,30,8)(37,30,5)(42,30,3)(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(15,17,3), (1,23,3), (17,25,3)]

Queue 1: [(4,21,3), (41,22,10), (44,25,3)]

Queue 2: [(39,24,1), (23,26,6), (14,27,3)]

Queue 3: [(19,21,1), (25,21,4)]

Queue 4: [(27,18,2), (12,25,6)]

Time: 30

Waiting clients:

(5,31,7)(24,31,10)(32,31,8)(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(15,17,2), (1,23,3), (17,25,3)]

Queue 1: [(4,21,2), (41,22,10), (44,25,3)]

Queue 2: [(23,26,6), (14,27,3), (37,30,5)]

Queue 3: [(25,21,4), (9,30,8), (42,30,3)]

Queue 4: [(27,18,1), (12,25,6)]

Time: 31

Waiting clients:

(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(15,17,1), (1,23,3), (17,25,3), (24,31,10)]

Queue 1: [(4,21,1), (41,22,10), (44,25,3), (32,31,8)]

Queue 2: [(23,26,5), (14,27,3), (37,30,5)]

Queue 3: [(25,21,3), (9,30,8), (42,30,3)]

Queue 4: [(12,25,6), (5,31,7)]

Time: 32

Waiting clients:

(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(1,23,3), (17,25,3), (24,31,10)]

Queue 1: [(41,22,10), (44,25,3), (32,31,8)]

Queue 2: [(23,26,4), (14,27,3), (37,30,5)]

Queue 3: [(25,21,2), (9,30,8), (42,30,3)]

Queue 4: [(12,25,5), (5,31,7)]

Time: 33

Waiting clients:

(0,34,10)(2,34,3)(20,34,5)(21,34,2)(6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(1,23,2), (17,25,3), (24,31,10)]

Queue 1: [(41,22,9), (44,25,3), (32,31,8)]

Queue 2: [(23,26,3), (14,27,3), (37,30,5)]

Queue 3: [(25,21,1), (9,30,8), (42,30,3)]

Queue 4: [(12,25,4), (5,31,7)]

Time: 34

Waiting clients: (6,35,2)(48,35,7)(34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(1,23,1), (17,25,3), (24,31,10), (20,34,5)]

Queue 1: [(41,22,8), (44,25,3), (32,31,8), (21,34,2)]

Queue 2: [(23,26,2), (14,27,3), (37,30,5)]

Queue 3: [(9,30,8), (42,30,3), (2,34,3)]

Queue 4: [(12,25,3), (5,31,7), (0,34,10)]

Time: 35

Waiting clients: (34,36,6)(35,36,8)(28,37,10)(30,40,9)

Queue 0: [(17,25,3), (24,31,10), (20,34,5), (6,35,2)]

Queue 1: [(41,22,7), (44,25,3), (32,31,8), (21,34,2)]

Queue 2: [(23,26,1), (14,27,3), (37,30,5), (48,35,7)]

Queue 3: [(9,30,7), (42,30,3), (2,34,3)]

Queue 4: [(12,25,2), (5,31,7), (0,34,10)]

Time: 36

Waiting clients: (28,37,10)(30,40,9)

Queue 0: [(17,25,2), (24,31,10), (20,34,5), (6,35,2)]

Queue 1: [(41,22,6), (44,25,3), (32,31,8), (21,34,2)]

Queue 2: [(14,27,3), (37,30,5), (48,35,7), (34,36,6)]

Queue 3: [(9,30,6), (42,30,3), (2,34,3), (35,36,8)]

Queue 4: [(12,25,1), (5,31,7), (0,34,10)]

Time: 37

Waiting clients: (30,40,9)

Queue 0: [(17,25,1), (24,31,10), (20,34,5), (6,35,2)]

Queue 1: [(41,22,5), (44,25,3), (32,31,8), (21,34,2)]

Queue 2: [(14,27,2), (37,30,5), (48,35,7), (34,36,6)]

Queue 3: [(9,30,5), (42,30,3), (2,34,3), (35,36,8)]

Queue 4: [(5,31,7), (0,34,10), (28,37,10)]

Time: 38

Waiting clients: (30,40,9)

Queue 0: [(24,31,10), (20,34,5), (6,35,2)]

Queue 1: [(41,22,4), (44,25,3), (32,31,8), (21,34,2)]

Queue 2: [(14,27,1), (37,30,5), (48,35,7), (34,36,6)]

Queue 3: [(9,30,4), (42,30,3), (2,34,3), (35,36,8)]

Queue 4: [(5,31,6), (0,34,10), (28,37,10)]

Time: 39

Waiting clients: (30,40,9)

Queue 0: [(24,31,10), (20,34,5), (6,35,2)]

Queue 1: [(41,22,3), (44,25,3), (32,31,8), (21,34,2)]

Queue 2: [(37,30,5), (48,35,7), (34,36,6)]

Queue 3: [(9,30,3), (42,30,3), (2,34,3), (35,36,8)]

Queue 4: [(5,31,5), (0,34,10), (28,37,10)]

Time: 40

Waiting clients:

Queue 0: [(24,31,8), (20,34,5), (6,35,2), (30,40,9)]

Queue 1: [(41,22,2), (44,25,3), (32,31,8), (21,34,2)]

Queue 2: [(37,30,4), (48,35,7), (34,36,6)]

Queue 3: [(9,30,2), (42,30,3), (2,34,3), (35,36,8)]

Queue 4: [(5,31,4), (0,34,10), (28,37,10)]

Time: 41

Waiting clients:

Queue 0: [(24,31,8), (20,34,5), (6,35,2), (30,40,9)]

Queue 1: [(41,22,1), (44,25,3), (32,31,8), (21,34,2)]

Queue 2: [(37,30,3), (48,35,7), (34,36,6)]

Queue 3: [(9,30,1), (42,30,3), (2,34,3), (35,36,8)]

Queue 4: [(5,31,3), (0,34,10), (28,37,10)]

Time: 42

Waiting clients:

Queue 0: [(24,31,6), (20,34,5), (6,35,2), (30,40,9)]

Queue 1: [(44,25,3), (32,31,8), (21,34,2)]

Queue 2: [(37,30,2), (48,35,7), (34,36,6)]

Queue 3: [(42,30,3), (2,34,3), (35,36,8)]

Queue 4: [(5,31,2), (0,34,10), (28,37,10)]

Time: 43

Waiting clients:

Queue 0: [(24,31,5), (20,34,5), (6,35,2), (30,40,9)]

Queue 1: [(44,25,2), (32,31,8), (21,34,2)]

Queue 2: [(37,30,1), (48,35,7), (34,36,6)]

Queue 3: [(42,30,2), (2,34,3), (35,36,8)]

Queue 4: [(5,31,1), (0,34,10), (28,37,10)]

Time: 44

Waiting clients:

Queue 0: [(24,31,4), (20,34,5), (6,35,2), (30,40,9)]

Queue 1: [(44,25,1), (32,31,8), (21,34,2)]

Queue 2: [(48,35,7), (34,36,6)]

Queue 3: [(42,30,1), (2,34,3), (35,36,8)]

Queue 4: [(0,34,10), (28,37,10)]

Time: 45

Waiting clients:

Queue 0: [(24,31,3), (20,34,5), (6,35,2), (30,40,9)]

Queue 1: [(32,31,8), (21,34,2)]

Queue 2: [(48,35,6), (34,36,6)]

Queue 3: [(2,34,3), (35,36,8)]

Queue 4: [(0,34,9), (28,37,10)]

Time: 46

Waiting clients:

Queue 0: [(24,31,2), (20,34,5), (6,35,2), (30,40,9)]

Queue 1: [(32,31,7), (21,34,2)]

Queue 2: [(48,35,5), (34,36,6)]

Queue 3: [(2,34,2), (35,36,8)]

Queue 4: [(0,34,8), (28,37,10)]

Time: 47

Waiting clients:

Queue 0: [(24,31,1), (20,34,5), (6,35,2), (30,40,9)]

Queue 1: [(32,31,6), (21,34,2)]

Queue 2: [(48,35,4), (34,36,6)]

Queue 3: [(2,34,1), (35,36,8)]

Queue 4: [(0,34,7), (28,37,10)]

Time: 48

Waiting clients:

Queue 0: [(20,34,5), (6,35,2), (30,40,9)]

Queue 1: [(32,31,5), (21,34,2)]

Queue 2: [(48,35,3), (34,36,6)]

Queue 3: [(35,36,8)]

Queue 4: [(0,34,6), (28,37,10)]

Time: 49

Waiting clients:

Queue 0: [(20,34,4), (6,35,2), (30,40,9)]

Queue 1: [(32,31,4), (21,34,2)]

Queue 2: [(48,35,2), (34,36,6)]

Queue 3: [(35,36,7)]

Queue 4: [(0,34,5), (28,37,10)]

Time: 50

Waiting clients:

Queue 0: [(20,34,3), (6,35,2), (30,40,9)]

Queue 1: [(32,31,3), (21,34,2)]

Queue 2: [(48,35,1), (34,36,6)]

Queue 3: [(35,36,6)]

Queue 4: [(0,34,4), (28,37,10)]

Time: 51

Waiting clients:

Queue 0: [(20,34,2), (6,35,2), (30,40,9)]

Queue 1: [(32,31,2), (21,34,2)]

Queue 2: [(34,36,6)]

Queue 3: [(35,36,5)]

Queue 4: [(0,34,3), (28,37,10)]

Time: 52

Waiting clients:

Queue 0: [(20,34,1), (6,35,2), (30,40,9)]

Queue 1: [(32,31,1), (21,34,2)]

Queue 2: [(34,36,5)]

Queue 3: [(35,36,4)]

Queue 4: [(0,34,2), (28,37,10)]

Time: 53

Waiting clients:

Queue 0: [(6,35,2), (30,40,9)]

Queue 1: [(21,34,2)]

Queue 2: [(34,36,4)]

Queue 3: [(35,36,3)]

Queue 4: [(0,34,1), (28,37,10)]

Time: 54

Waiting clients:

Queue 0: [(6,35,1), (30,40,9)]

Queue 1: [(21,34,1)]

Queue 2: [(34,36,3)]

Queue 3: [(35,36,2)]

Queue 4: [(28,37,10)]

Time: 55

Waiting clients:

Queue 0: [(30,40,9)]

Queue 1: closed

Queue 2: [(34,36,2)]

Queue 3: [(35,36,1)]

Queue 4: [(28,37,9)]

Time: 56

Waiting clients:

Queue 0: [(30,40,8)]

Queue 1: closed

Queue 2: [(34,36,1)]

Queue 3: closed

Queue 4: [(28,37,8)]

Time: 57

Waiting clients:

Queue 0: [(30,40,7)]

Queue 1: closed

Queue 2: closed

Queue 3: closed

Queue 4: [(28,37,7)]

Time: 58

Waiting clients:

Queue 0: [(30,40,6)]

Queue 1: closed

Queue 2: closed

Queue 3: closed

Queue 4: [(28,37,6)]

Time: 59

Waiting clients:

Queue 0: [(30,40,5)]

Queue 1: closed

Queue 2: closed

Queue 3: closed

Queue 4: [(28,37,5)]

Average waiting time: 6.42

Peak hour: 37

Average service time: 5.44

Rezultatul pentru testul trei este:

Average waiting time: 114.858

Peak hour: 100

Average service time: 6.51

6. Concluzii

In concluzie, sunt de parere ca acest proiect mi-a aprofundat cunostiintele in tot ce inseamna limbajul Java, implementarea paradigmelor OOP si creerea unui program cu o interfata grafica. De asemenea m-au ajutat sa imi reamintesc tehnicile de programare invatate semestrul trecut. Aplicatia de simulare o putem folosi pentru a observa comportamentul fluxului clientilor unui magazin si cu ajutorul unor mici imbunatatiri se poate si eficientiza acesta prin observarea timpului de varf si adaugarea unor cozi.