DOCUMENTATIE TEMA NUMARUI DOI

QUEUES MANAGEMENT APPLICATION USING THREADS AND SYNCHRONIZATION MECHANISMS

NUME STUDENT:Sulea Alexandra-Maria GRUPA: 30222

CUPRINS

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

1. Obiectivul temei

Scopul principal al acestei teme este de a crea o aplicație de gestionare a cozilor care să atribuie clienții la cozi astfel încât timpul de așteptare să fie minimizat. Aplicația simulează sosirea și servirea unei serii de clienți, ținând cont de timpul lor de așteptare în cozi și timpul necesar pentru servire. Scopul final este de a calcula timpul mediu de așteptare al clienților și de a optimiza gestionarea cozilor pentru a reduce acest timp.

Objectivele secundare sunt:

- Alegerea strategiei de distributie prin 2 metode:
 - 1. Timpul cel mai scurt de asteptare
 - 2. Coada cu cei mai putini clienti
- Afisarea in timp real intr-o interfata grafica intreaga simulare

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

Cerinte functionale:

- 1. Simularea sosirii si servirii unei serii de client.
- 2. Atribuirea clientilor la cozi astfel incat timpulde asteptare sa fie minimizat.
- 3. Calculul timpului mediu de asteptare al clientilor.
- 4. Optimizarea gestionarii cozilor pentru reducerea timpului de asteptare.

Cerinte non-functionale:

1. Afisarea in timp real a simularii intr-o interfata.

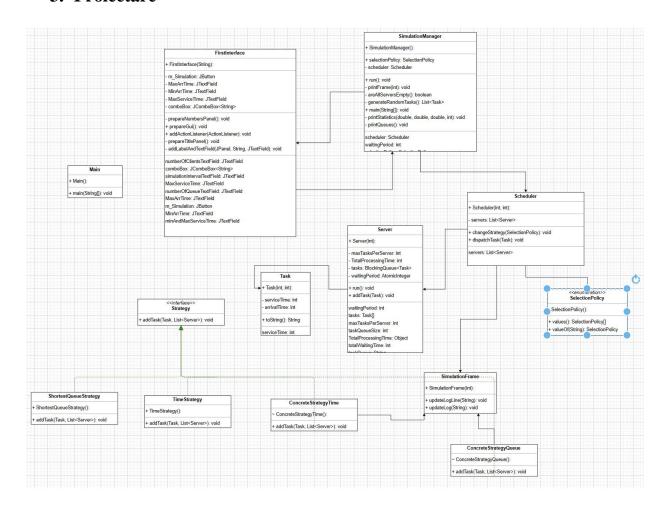
Mod de utilizare:

1. Utilizatorul introduce parametrii pentru simulare si alege strategia de distrubuire dorita.

Queues Management Application		-		×	
QUEUES MANAGEMENT APPLICATION USING THREADS AND SYNCHRONIZATION MECHANISMS					
Number Of Clients:					
Number Of Queue:					
Simulation Interval:					
Minimum Arrival Time:					
Maximum Arrival Time:					
Minimum Service Time:					
Maximum Service Time:					
Choose an option: Shortest queue ▼					
Simu	lation				

2. Dupa apasarea butonului de simulare, se va deschide o alta interfata cu adaugarea clientilor in cozi in functie de datele introduse initial.

3. Proiectare



4. Implementare

Clasele sunt:

FirstInterface:Cu ajutorul primei interfate, utilizatorul introduce datele pentru simulare.Prin apasarea butonului de pe aceasta interfata se initializeaza a doua interfata grafica, cu simularea.

Scheduler: de aici se gestioneaza cozile si distrinuirea clientilor.

SelectionPolicy: reprezinta cele 2 modalitati de distribuire.

SimulationFrame:o interfata grafica pentru afisarea starii de simulare.

SimulationManager:coordoneaza simularea intregului process.

Server: Fiecare coada are o lista de client de tipul task, de aici fiecare thread pornit avand ca parametru primul client din lista, este oprit pentru perioada de procesare a acestuia.

Task: Fiecare client este reprezentat de un obiect de tip Task.

5. Rezultate

Pentru verificarea corectitudinii aplicatiei, am rulat codul, rezultatele fiind in fisiere test1.txt, test2.txt si test3.txt.

6. Concluzii

Prin intermediul acestei teme am invatat utilizarea Thread-urilor si sincronizarea lor cu ajutorul DeblockingDeque pentru stocarea cozilor. Posibile dezvoltari ulterioare ar fi utilizarea atributului waitingPeriod din clasa Server pentru a verifica timpul de asteptare curent, necesar la calcularea timpului mediu de asteptare si in distribuirea clientilor din clasa ConcreteStrategyTime.

7. Bibliografie

https://dsrl.eu/courses/pt/materials/PT_2024_A2_S1.pdf https://dsrl.eu/courses/pt/materials/PT_2024_A2_S2.pdf

https://www.w3schools.com/java/java_threads.asp