DOCUMENTATIE

TEMA *2*

NUME STUDENT: …Avram Darius

GRUPA: …………5………

# CUPRINS

[1. Obiectivul temei 3](file:///C:\foldere\Darius\PT2023_5_Avram_Darius_Assignment_1\PT2023_Documentatie_RO.doc#_Toc95297885)

[2. Proiectare 3](file:///C:\foldere\Darius\PT2023_5_Avram_Darius_Assignment_1\PT2023_Documentatie_RO.doc#_Toc95297887)

[3. Implementare 4](file:///C:\foldere\Darius\PT2023_5_Avram_Darius_Assignment_1\PT2023_Documentatie_RO.doc#_Toc95297888)

[4. Concluzii 4](file:///C:\foldere\Darius\PT2023_5_Avram_Darius_Assignment_1\PT2023_Documentatie_RO.doc#_Toc95297890)

[5. Bibliografie 5](file:///C:\foldere\Darius\PT2023_5_Avram_Darius_Assignment_1\PT2023_Documentatie_RO.doc#_Toc95297891)

# Obiectivul temei

Obiectivul acestui proiect este de a proiecta și implementa o aplicație de gestionare a cozilor care atribuie clienților cozi astfel încât timpul de așteptare să fie minimizat. Aplicația simulează un număr N de clienți care sosesc pentru serviciu, intră în cozi, așteaptă, sunt serviți și în cele din urmă părăsesc cozile. Fiecare client este generat la începutul simulării și este caracterizat de trei parametri: ID (un număr între 1 și N), timpul de sosire (momentul în care sunt gata să intre în coadă în timpul simulării) și timpul de servire (intervalul de timp necesar pentru a servi clientul; adică timpul de așteptare atunci când clientul este în fața cozii). Aplicația urmărește timpul total petrecut de fiecare client în cozi și calculează timpul mediu de așteptare. Fiecare client este adăugat în coada cu cel mai mic timp de așteptare atunci când timpul său de sosire este mai mare sau egal cu timpul de simulare (timpul de sosire ≥ timpul simulării).

# Proiectare

# 

# Implementare

**ConcreteStrategyQueue** implementează interfața Strategy și reprezintă o strategie de adăugare a unei sarcini (Task) într-unul dintre servere (Server). În metoda addTask, strategia caută coada de așteptare cea mai mică din lista de servere și adaugă sarcina în acea coadă.

**ConcreteStrategyTime** implementează interfața Strategy și reprezintă o altă strategie de adăugare a unei sarcini într-unul dintre servere, bazată pe timpul de servire al sarcinilor. În metoda addTask, strategia caută serverul cu cel mai mic timp de servire și adaugă sarcina acolo.

**Scheduler** gestionează lista de servere și implementează logica de schimbare a strategiei. Ea inițializează lista de servere în constructor și creează un fir de execuție pentru fiecare server. Metoda changeStrategy schimbă strategia curentă în funcție de politica de selecție primită ca argument.

**SimulationManager** primește parametrii pentru configurarea simulării (numărul de clienți, numărul de servere, intervalul de simulare, timpul de sosire și timpul de servire) și creează o instanță de Scheduler. Simularea rulează într-un buclă while care se execută până la atingerea limită de timp sau până când nu mai sunt clienți în cozi. În fiecare pas al buclei, se adaugă clienți în funcție de timpul de sosire, se afișează starea clienților și a cozilor și se actualizează timpul.

**RealTimeQueue** este un JFrame care afișează outputul din terminal în timp real. Acesta redirecționează System.out și System.err către un obiect PrintStream personalizat, care afișează mesajele într-un obiect JTextArea. De asemenea, creează un fir de execuție separat care citește în mod continuu din System.out și System.err și afișează mesajele în JTextArea.

**SimulationFrame** reprezintă fereastra principală a aplicației GUI. Aici se afișează elementele de interfață utilizate pentru introducerea parametrilor simulării. La apăsarea butonului "Submit", se creează o instanță de SimulationManager și se porneste un fir de execuție pentru simulare.

# Concluzii

În urma realizării acestei teme am învățat să creez o interfață simplă, funcțională, care face mai ușoară introducerea datelor.

De asemenea, am învățat să gestionez resursele în mod inteligent pentru a oferi o experiență cât mai optimă posibilă pentru clienți.

# Bibliografie

<https://stackoverflow.com/questions/tagged/java>

<https://www.geeksforgeeks.org/java>   
<https://www.javatpoint.com/java-tutorial>  
<https://www.javacodegeeks.com>