

Documentatie

Tema 2

NUME STUDENT: Fekete Borz Victor

GRUPA: 30225

CUPRINS

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

- **Obiectivul temei**

Obiectivul principal al temei 2 este realizarea si simularea unei sistemului client-coada de la un magazin tinand cont de un timp de asteptare

Clasele utilizate in cadrul proiectului:

- Realizarea unei clase pentru crearea unui client random.
- Realizarea unei clase pentru procesarea clientului (clients,queues,arrivalMin, arrivalMax, processMin,processMax).
- Realizarea unei clase pentru stocarea datelor unui client.
- Realizarea unei clase pentru realizarea unei simulari .
- Realizarea unei clase pentru compararea timpilor de ajungere a clientilor si modificarea cozii (thread-uri) .
- Realizarea unei interfete grafice.

- Analiza problemei, modelare, scenarii, cazuri de utilizare

Simulation Settings

Number of clients:

Number of queues:

Minimum arrival time:

Maximum arrival time:

Minimum time of processing:

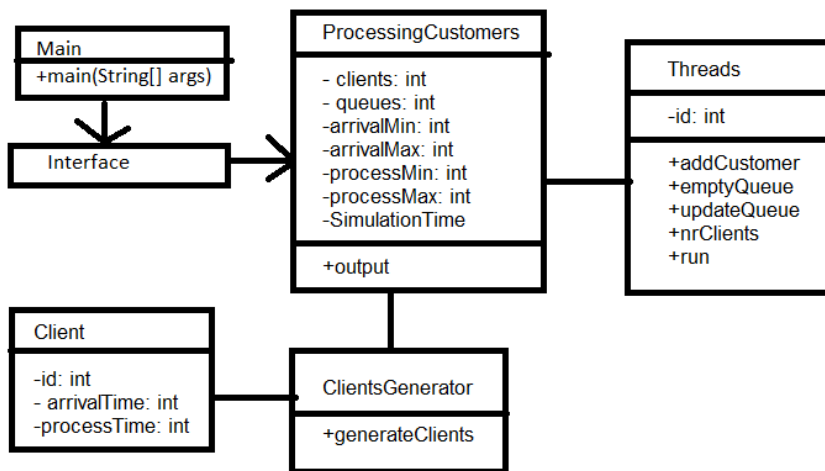
Maximum time of processing:

Simulation time:

Start Simulation

Input data

- Proiectare



- **Implementare**

Acest proiect contine 6 clase :

Pachetul Options este alcatuit din 3 clase:

Prima clasa : “**Customer**”

In aceasta clasa, producem clienti care au urmatoarele specificatii : ID, arrivalTime(timpul la care ajunge) si processingTime(cat timp dureaza sa fie servit) . De asemenea, in aceasta clasa regasim gettere si settere .

A doua clasa : “**ClientsGenerator**”

In aceasta clasa, generam clientii cu functia Random si le atribuim valori la intamplare.

A treia clasa: “**ProcessingCustomer**”

In aceasta clasa, tinem evidenta unei liste de cozi, aici sunt create de asemenea cozile si thread-urile aferente pentru fiecare coada. Avem o metoda care se ocupa de alegerea cozii in care va fi adaugat un client, tinand cont de cel mai mic timp de asteptare. Aceasta se ocupa de Avem o metoda ce calculeaza “PeakHour” si “serviceTime” pentru fiecare coada la fiecare moment de timp si regasim gettere si settere. Aceasta clasa se ocupa de afisarea simularii in timp real in fisierul txt “Log4”.

Pachetul Server contine o clasa:

“**Threads**”

Această clasă implementează interfața **Runnable** și definește un fir de execuție (thread) care poate fi rulat în paralel cu alte fire de execuție. Clasa definește o coadă de obiecte **Customer** folosind o coadă blocantă cu priorități (**PriorityBlockingQueue**) și oferă metode pentru adăugarea unui client la coadă (**addCustomer**), verificarea timpului de așteptare (**getWaitingTime**), numărul de clienți din coadă (**nrClients**) și actualizarea stării cozii (**updateQueue**).

Pachetul Main contine o clasa

“**Main**”

Această clasă conține metoda **main** și este punctul de intrare în aplicație. Aceasta creează o instanță a clasei **Interface**, care reprezintă interfața grafică a utilizatorului și este responsabilă pentru afișarea informațiilor și interacțiunea cu utilizatorul. Astfel, atunci când aplicația este pornită, interfața grafică va fi deschisă.

Pachetul GUI contine o clasa

“Interface”

Această clasă definește o interfață grafică pentru introducerea datelor de intrare pentru o simulare de procesare a clienților. Utilizatorul poate introduce numărul de clienți, numărul de cozi, timpul minim și maxim de sosire al clienților, timpul minim și maxim de procesare a clienților și timpul total de simulare. După ce utilizatorul introduce datele de intrare, poate apăsa butonul "Start Simulation" pentru a începe simularea. Dacă datele introduse sunt corecte, clasa va crea un obiect ProcessingCustomers și va afișa rezultatele simulării.

- **Rezultate**

Rezultatele celor 3 teste vor fi puse pe GitHub.

- **Concluzii**

Din aceasta tema am invatat sa lucrez cu thread-uri.

- **Bibliografie**

[*Untitled Diagram - diagrams.net*](#)

[*\(68\) How to create UML Class Diagrams with IntelliJ Ultimate macOS – YouTube*](#)

[*PT2023_A2.pdf \(dsrl.eu\)*](#)

[*FUNDAMENTAL PROGRAMMING TECHNIQUES \(dsrl.eu\)*](#)

[*FUNDAMENTAL PROGRAMMING TECHNIQUES \(dsrl.eu\)*](#)