Documentatie

Assigment 2

NUME STUDENT: Neacă Radu-Sabin

GRUPA: 30222

Cuprins

[1. Obiectivul temei 3](file:///C:\Downloads\FACULTATE\Tehnici%20de%20programare\PT2021-2022_Documentation_Template.doc#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](file:///C:\Downloads\FACULTATE\Tehnici%20de%20programare\PT2021-2022_Documentation_Template.doc#_Toc95297886)

[3. Proiectare 3](file:///C:\Downloads\FACULTATE\Tehnici%20de%20programare\PT2021-2022_Documentation_Template.doc#_Toc95297887)

[4. Implementare 3](file:///C:\Downloads\FACULTATE\Tehnici%20de%20programare\PT2021-2022_Documentation_Template.doc#_Toc95297888)

[5. Rezultate 3](file:///C:\Downloads\FACULTATE\Tehnici%20de%20programare\PT2021-2022_Documentation_Template.doc#_Toc95297889)

[6. Concluzii 3](file:///C:\Downloads\FACULTATE\Tehnici%20de%20programare\PT2021-2022_Documentation_Template.doc#_Toc95297890)

[7. Bibliografie 3](file:///C:\Downloads\FACULTATE\Tehnici%20de%20programare\PT2021-2022_Documentation_Template.doc#_Toc95297891)

Obiectivul temei

Obiectiv principal:

Scopul lucrarii consta in crearea unei aplicatii ce calculeaza timpul mediu de asteptare al unui client in momentul in care acesta s-a pus la coada. Acest lucru il voi realiza folosindu-ma de o lista de Thread-uri, fiecare Thread fiind o coada de clienti.

Obiective secundare:

* Analiza problemei si identificarea cerintelor.
* Organizarea codului corespunzaor: Precum structura MVC (Model-View-Controller). Fiecare clasa creata nu trebuie sa depaseasca o dimensiune mai mare de 300 de linii, cu exceptia clasei Controller. Fiecare metoda trebuie sa cuprinda maxim 30 de linii. Clasele trebuie impartite in pachete astfel incat sa existe o vizibilitate si organizare a proiectului mult mai buna.
* Testarea aplicatiei, verificarea codului astfel incat metodele folosite pe parcursul programului sa fie corect implementate, fara a prezenta cazuri de eroare.
* Implementarea si simularea unei interfete ce sa dispuna si de o fereastra ce afiseaza rezultatul dorit.
* Folosirea thread-urilor si a mecanicilor de sincronizare pentru a trimite in paralel cu procesul principal, mai multe procese, fiecare reprezentand o coada de clienti.

Analiza problemei, modelare, scenarii, cazuri de utilizare

Primul pas inainte de implementarea proiectului ar fi analiza problemei si modul cum ar trebui rezolvata aceasta. Asadar, am ales sa tratez problema client-coada, ca fiind o legatura task-server. Fiecare client reprezinta un task pe care server-ul trebuie sa il adauge, iar mai apoi, sa il execute conform cerintei. Astfel, server-ul reprezinta o coada de task-uri ce se bazeaza pe principiul FIFO (First in first out). Fiecare client are un timp cand de sosire in fata cozilor, si alt timp ce sugereaza timpul pe care il petrece in fata cozii, pe primul lor. Astfel, task-ul va continu un ID ce semnifica numele clientului, un numar TimeArrival ce semnifica momentul cand ajunge la coada si un numar TimeService, ce reprezinta durata de timp pe care o are de asteptat in momentul cand ajunge pe prima pozitie din coada. Aceasta aplicatie poate sa contina mai multe servere pentru a imbunatati sistemul, marind viteza de parcurgere a Task-urilor, in cazul in care numarul acestora ar fi prea mare pentru a fi rezolvate cateva servere.

# Obiectivul temei

*Se vor prezenta: (i) obiectivul principal al temei printr-o fraza, si (ii) obiectivele secundare sub forma tabelara sau ca lista. Obiectivele secundare reprezinta pasii care trebuie urmati pentru indeplinirea obiectivului principal. Fiecare obiectiv secundar va fi descris si se va indica in care capitol al documentatiei va fi detaliat.*

# Analiza problemei, modelare, scenarii, cazuri de utilizare

*Se va prezenta cadrul de cerinte functionale formalizat si cazurile de utilizare ca si diagrame si descrieri de use-case. Descrierile use-case-urilor se vor face sub forma unui flow-chart ori sub forma unei liste continand pasii executiei fiecarui use-case.*

# Proiectare

*Se va prezenta proiectarea OOP a aplicatiei, diagramele UML de clase si de pachete, structurile de date folosite, interfetele definite si algoritmii folositi (*daca e cazul)

# Implementare

*Se va descrie fiecare clasa cu campuri si metodele importante. Se va descrie implemantarea interfetei utilizator.*

# Rezultate

*Se vor prezenta scenariile pentru testare cu Junit sau alt framework de testare.*

# Concluzii

*Se vor prezenta concluziile, ce s-a invatat din tema, dezvoltari ulterioare.*

# Bibliografie