Transformarea logaritmica a nivelului de gri in JAVA

**Introducere**

Proiectul "Transformarea logaritmica a nivelului de gri" este o aplicație Java care implementează funcționalitatea de citire, procesare și scriere a imaginilor BMP, aplicând o transformare specifică asupra nivelurilor de gri.

**Descrierea Aplicatiei**

Aplicația primește de la utilizator calea imaginii de procesat și calea destinației pentru imaginile rezultate. Apoi, utilizează un proces de producție-consum pentru a segmenta imaginea și a o converti la nivel de gri.

**Partea Teoretica**

**BMP** este un format de fișier pentru stocarea imaginilor pe calculator, cunoscut și sub numele de fișiere bitmap. Fiecare pixel într-o imagine BMP este reprezentat printr-un număr de biți, iar culorile pot fi reprezentate folosind diferite scheme de culori.

O imagine care conține text, captură de ecran, Font

Descriere generată automat

**Transformarea logaritmica**

Este o tehnică comună utilizată în procesarea imaginilor pentru a îmbunătăți contrastul. În acest proiect, aplicăm o transformare logaritmică asupra nivelurilor de gri ale fiecărui pixel din imagine.

Formula matematica este:

O imagine care conține text, Font, linie, alb

Descriere generată automat

O imagine care conține text, captură de ecran, Font

Descriere generată automat

**Descrierea Implementarii**

**Structura proiectului**

Proiectul este împărțit în mai multe clase, fiecare având o funcționalitate specifică.

Acestea includ:

O imagine care conține text, captură de ecran, Font, software

Descriere generată automat

**Descrierea modulelor**

* Main: Clasa principală pentru testarea întregului proces.
* ImageThread: Implementează metodele comune utilizate de ambele fire de execuție.
* Forma: Clasă abstractă cu metode pentru calculul ariei în funcție de diferite configurări.
* FormaMostenire: Clasă care moștenește funcționalitățile clasei Forma și furnizează o implementare specifică pentru calculul ariei.
* Pipe: Gestionarea comunicării între thread-uri.
* ImageProcessor: Definirea interfeței pentru procesarea imaginilor.
* Mama: Creează și gestionează un fir de execuție, executând un număr fix de pași.
* Tata: Extinde clasa Thread și rulează în mod repetat, afișând numărul de pași și indicând stadiul execuției.
* ProducerThread și ConsumerThread: Implementarea producer-consumer pentru procesarea datelor în paralel.

**Descrierea implementarii**

* Implementarea proiectului folosește limbajul de programare Java și include două fire de execuție (Producer și Consumer) pentru a realiza segmentarea și conversia la nivel de gri.
* Comunicarea între aceste fire de execuție se face prin intermediul unei conducte (Pipe).

**Descrierea functionala:**

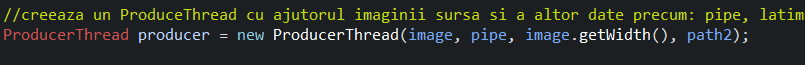
**Main**

-Primește de la utilizator calea imaginii de procesat și calea destinației.

O imagine care conține text, captură de ecran, Font

Descriere generată automat

-Creează un obiect ProducerThread pentru procesarea imaginii.



-Afisează timpul necesar pentru citire și instantiere, evaluând astfel performanța programului.

****

**ImageProcessor**

-Furnizează metodele Produce și Consume pentru producerea și consumul segmentelor de imagine.

**ProducerThread**

-Primește datele imaginilor de la modulul de citire, procesează imaginile și furnizează datele procesate către modulul de consum.

**ConsumerThread**

-Primește datele procesate de la modulul de producție și le scrie într-un fișier.

**ImageThread**

-Clasă abstractă pentru firele de execuție care procesează imaginea.

**Forma si FormaMostenire**

-Modelează conceptul de formă geometrică și furnizează o implementare specifică prin moștenire.

**Pipe**

-Asigură comunicarea între ProducerThread și ConsumerThread.

**Mama si Tata**

-Implementează fire de execuție (Runnable și Thread) pentru clasele Mama și Tata.





**Evaluarea performanței:**

**Timpul Total de Execuție**

-Măsoară timpul total necesar pentru a citi, procesa și scrie întreaga imagine.



O imagine care conține text, Font, captură de ecran, Grafică

Descriere generată automat

**Timpul Alocat Fiecărui Sfert de Imagine**

-Monitorizează timpul alocat pentru fiecare sfert de imagine procesat individual.

O imagine care conține text, captură de ecran, Font, software

Descriere generată automat

**Gestionarea Resurselor si Comunicarea**

-Evaluează eficiența comunicării între ProducerThread și ConsumerThread prin intermediul clasei Pipe. Acest aspect este realizat prin afisarea in consola a unor mesaje semnificative a comunicării între ProducerThread și ConsumerThread.

**Sincronizarea Firului de Execuție**

- Verifică modul în care firul de execuție ProducerThread interacționează cu ConsumerThread.

**Eficiența Segmentării și Procesării Paralele**

- Eficiența segmentării și procesării paralele în codul tău este evidențiată de modul în care imaginea este împărțită în segmente și procesată în paralel de către mai multe fire de execuție. Aceasta se referă în special la funcționalitatea implementată în ProducerThread și ConsumerThread.

**Concluzii:**

Implementarea segmentării imaginii și procesarea în paralel a segmentelor prin utilizarea firelor de execuție (ProducerThread și ConsumerThread) arată o abordare de optimizare pentru procesarea eficientă a imaginilor mari, exemplu fiind imaginile cu extensie BMP.

Implementarea gestionării excepțiilor pentru cazurile specifice, cum ar fi mărimea incorectă a segmentelor (IllegalArgumentException), arată preocupare pentru robustețea programului.

O imagine care conține text, captură de ecran, software, Software multimedia

Descriere generată automat

Integrarea interacțiunii utilizatorului prin citirea căilor imaginilor de la tastatură adaugă o dimensiune interactivă la program.

**Biblioteci**

O imagine care conține text, captură de ecran, Font

Descriere generată automat O imagine care conține text, Font, captură de ecran, tipografie

Descriere generată automat