

Operații binare între două imagini
Aplicații web cu suport java

Proiect realizat de:
Sătmar Elena-Elisabeta
Grupa 332AA

2023-2024

Introducere:

În lumea programării Java, manipularea imaginilor devine o abilitate esențială în dezvoltarea aplicațiilor vizuale și de prelucrare a datelor. Cu ajutorul unor operații simple, de bază, precum AND, OR și XOR se pot crea noi imagini din unele deja existente.

Descrierea aplicației cerute:

Aplicația cerută dorește implementarea cu ajutorul programării orientate pe obiecte a unui program care realizează construirea unei noi imagini prin îmbinarea a două imagini prin condițiile: AND, OR, XOR.

Aplicația va utiliza în totalitate conceptele POO: încapsulare, moștenire, polimorfism și abstractizare.

Imaginea sursă va respecta formatul .bmp.

Codul sursă respectă "Coding standards".

Se va efectua lucru cu fișiere.

Aplicația trebuie să fie multimodulară.

Partea teoretică

Logica binară reprezintă cea mai fundamentală și răspândită formă de organizare a operațiilor logice. Aceasta se bazează pe ideea că o afirmație poate să aibă doar una dintre cele două rezultate posibile: fie este adevărată, fie este falsă. Dintr-o perspectivă abstractă, se utilizează valorile binare 1 și 0 pentru a simboliza aceste două alternative.

Pentru a lucra cu logica binară, există trei mari operații fundamentale cu care vom lucra: AND (și), OR (sau) și XOR (sau exclusiv). Modul de lucru al acestora este prezentat în tabelul de mai jos:

| Operație | Simbol logic | Simbol programare (logică) | Simbol programare (binar) | Operator 1 | Operator 2 | Rezultat |
|----------|--------------|----------------------------|---------------------------|------------|------------|----------|
| AND (și) | \wedge | && | & | 0 | 0 | 0 |
| | | | | 0 | 1 | 0 |
| | | | | 1 | 0 | 0 |
| | | | | 1 | 1 | 1 |
| OR (sau) | \vee | | | 0 | 0 | 0 |
| | | | | 0 | 1 | 1 |

| | | | | | | |
|---------------------------------|----------|----------|----------|---|---|---|
| | | | | 1 | 0 | 1 |
| | | | | 1 | 1 | 1 |
| XOR (sau exclusiv) | \oplus | \wedge | \wedge | 0 | 0 | 0 |
| | | | | 0 | 1 | 1 |
| | | | | 1 | 0 | 1 |
| | | | | 1 | 1 | 0 |

Descrierea aplicației:

Aplicația este creată cu ajutorul programelor Java jdk 8 și Eclipse Mars 2. Aceasta este structurată în mai multe clase care conțin toate elementele conceptuale din programarea orientată pe obiecte: încapsulare, moștenire, polimorfism și abstractizare.

Aceasta are câteva constrângeri:

- Imaginile trebuie să se regăsească în același folder cu proiectul.
- Imaginile trebuie să aibă unul dintre următoarele formate: .jpeg/.jpg, .png sau .bmp
- Imaginile trebuie să aibă același format

Aplicația este structurată în două package-uri: Main și operation

Package-ul main va conține, așa cum specific și numele acestuia, aplicația principală.

Package-ul operation va conține următoarele clase:

- Operation
- AndOperation
- OrOperation
- XorOperation
- ApplyOperationTask
- ImageProcessor
- ProcessingTask

Algoritm:

Pas 1. Introducerea imaginilor în argumente

Pas 2. Rularea programului

Pas 3. Introducerea operației dorite

Pas 4. Introducerea numărului de thread-uri dorite:

Pas 5. Imaginile vor fi introduse în program și se va calcula minimul de pixeli pe lungime și înălțime pentru ambele poze. Se împarte matricea în blocuri de câte thread-uri se dorește pe care va aplica una din cele trei operații dorite. Dacă imaginea este color, se va face ponderea fiecărui pixel.

Pas 6. Se va returna în același folder cu aplicația imaginea de convoluție dintre cele două imagini de input sub numele de output.png

Descrierea modulelor:

Operation

Este o interfață pentru cele trei operații

AndOperation

Implementează funcția AND între cele două surse

OrOperation

Implementează funcția OR între cele două surse

XorOperation

Implementează funcția XOR între cele două surse

ProcessingTask

Se va introduce abstractizarea. În această parte se va afla timpul de start și de final al rulării aplicației și numărul de cicli.

ApplyOperationTask

Acesta va parcurge matricea și va aplica operația dorită pe fiecare pixel în parte.

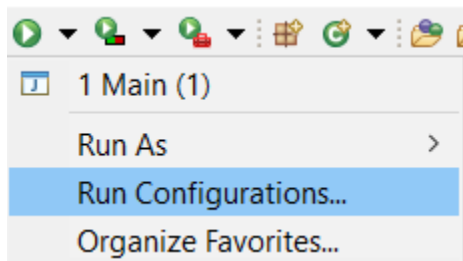
ImageProcessor

Se creează un nou buffer pentru imagine și se parcurge lățimea și înălțimea. De asemenea, în această parte se implementează codul pentru identificarea și ajustarea pixelilor. Se verifică dacă sunt valide coordonatele introduse și returnează lățimea și înălțimea rezultatului final.

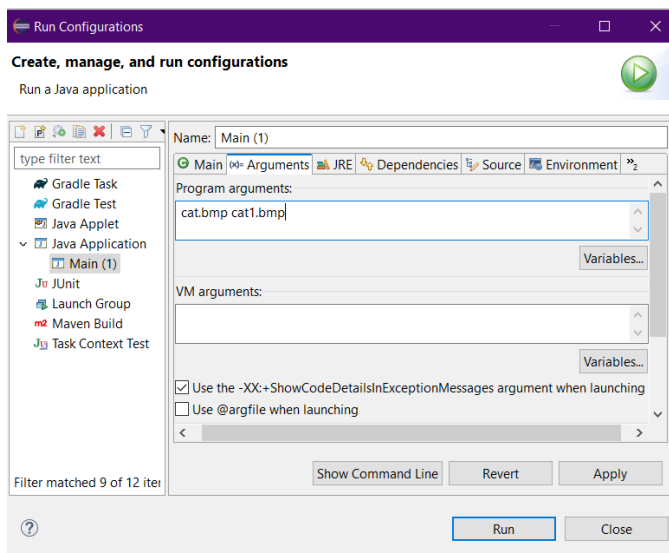
Rularea aplicației:

Pentru rularea aplicației se va introduce proiectul în Eclipse Mars 2. Înainte de rularea efectivă a aplicației, se vor introduce cele două imagini care se doresc a fi folosite în modul următor:

Se va accesa din bara de execuție butonul Run Configurations



Se va apăsa pe butonul Arguments și se va introduce numele celor două imagini pe care dorim să le folosim în Program Arguments, așa cum este prezentat și în imaginea de mai jos:



După introducerea argumentelor se va apăsa apply și apoi run.

Evaluarea performanțelor:

Programul funcționează corect, deoarece îndeplinește cerințele date și anume aplicarea operațiilor logice binare AND, OR, XOR pe două imagini. Programul rulează fără întreruperi, dacă constrângerile sunt îndeplinite. De asemenea, timpul de execuție este foarte scurt.

Concluzii:

În concluzie, aplicația va calcula minimul de lățime și lungime pentru cele două imagini în vederea creerii unei noi imagini, va calcula o medie a valorilor RGB a fiecărui pixel și va transpune într-o nouă imagine răspunsul operației alese.