Proiect Java

-Converting Color Image to GrayScale Image-

-Average method-

Student: Buzatu Rares Tudor

Grupa: 332AA

Cuprins

1. Introducere…………………………………………………..…..3

2. Descrierea aplicatiei cerute……………………………....3

3. Partea teoretica…………………………………………….…..3

4. Implementare……………………………………………….…..4

4.1. Structura(arhitectura) …………..………………..4

4.2. Functionalitate………………………………………..5

4.3. Concepte POO aplicate……………………………6

5. Evaluare performante……………………..………………..8

6. Concluzii……………………………………………….…………..9

7. Bibliografie………………………………..……………………..9

1.Introducere

Acest proiect reprezinta tema de casa la materia Aplicatii Web cu suport Java. El se axeaza pe partea de Java si are ca si scop prelucrarea de imagini si demonstrarea principiilor de baza ale programarii orientate obiect.

2. Descrierea aplicatiei cerute

Asa cum spune si titlul acestei lucrari, aplicatia are rolul de a face conversia de la o imagine RGB la una GrayScale. Imaginea trebuie sa aibe formatul .bmp, iar dimensiunea unui pixel sa fie de 24 de biti(8 biti x 3 canale). Conversia din color in alb-negru va trebui sa se realizeze cu ajutorul metodei mediei aritmetice. Aplicatia trebuie sa fie insotita de un document in care sa se explice functionalitatile sale. Deasemenea ea trebuie sa inregistreze timpul fiecarei operatii pe care o exectuta: citire, procesare, scriere.

3. Partea teoretica

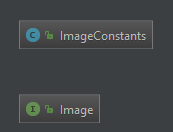
Metoda amintita mai sus ia valoarea de rosu, verde si albastru pentru fiecare pixel, le aduna, iar apoi le imparte la 3. Rezultatul va fi salvat pe toate cele 3 canale ale pixelului. Din punct de vedere al implementarii, proiectul se va realiza in Java fie cu Eclipse fie cu IntelliJ si va trebui sa contina elementele de baza ale programarii orientate obiect: mostenire, interfata, polimorfism, incapsulare, abstractizare.

4. Implementare:

4.1. Structura(arhitectura):

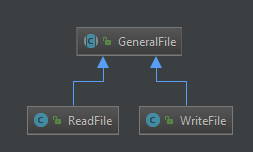
Modul in care am organizat functionalitatile aplicatiei este urmatorul: structura este impartita in 3 pachete + clasa principala “Main”. Cele 3 pachete sunt:

1. interfacesAndConstants:



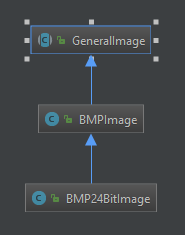
Acest pachet contine toate constantele de care aplicatia are nevoie, plus interfata Image.

1. fileImplementations:



Acest pachet contine cele doua clase (ReadFile si WriteFile) care mostenesc clasa abstracta GeneralFile. Clasele au rolul de a citi si de a scrie din fisier informatia.

1. Implementations:



Cele 3 clase au rolul de a descrie imagine ace urmeaza a fi procesata. Tot aici se gasesc si metodele de procesare a ei. Aceasta reprezinta o mostenire pe 3 niveluri. Clasa abstracta GeneralImage implementeaza o parte din metodele interfetei Image, iar clasele BMPImage si BMP24BitImage au rolul de a particularize si adauga metode fata de cele deja existente in GeneralImage.

4.2. Functionalitate:

Urmeaza sa descriu pas cu pas functionalitatea aplicatiei:

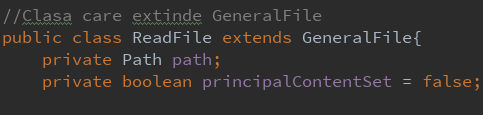
In primul rand, imaginile RGB sunt intr-un folder Input, iar cele alb-negru intr-un folder Output. Programul primeste ca si parametrii din linie de comanda path-urile catre imaginiea de input si cea de output dorite.

Apoi, cu ReadFile se citeste din imaginea de input header-ul si blocul de informatii, care sunt retinute intr-un vector de bytes, vectorul fiind unul din campurile clasei BMP24BitImage. Apoi se apeleaza metoda colorToGrayScale pentru a face conversia, iar rezultatul este salvat in alt obiect BMP24BitImage.

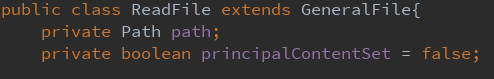
Acest obiect din urma este trimis ca si argument constructorului unui obiect WriteFile, obiect care are rolul de a scrie in fisierul de Output informatia(header + bloc de informatii). Dupa fiecare parte a programului, se afiseaza timpul necesar rularii. Timpul este afisat doar daca utilizatorul a selectat aceasta optiune din linie de comanda la inceputul exectutiei programului.

4.3. Concepte POO aplicate:

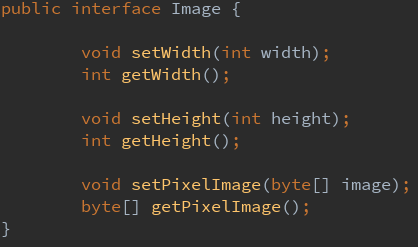
1. Mostenire: Rezulta din clasele de mai sus



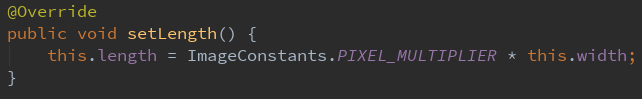
1. Incapsulare:



1. Abstractizare:



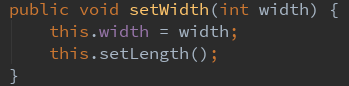
1. Polimorfism:



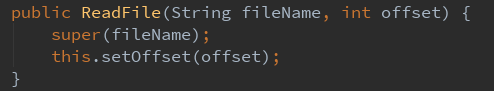


Alte tehnici aplicate:

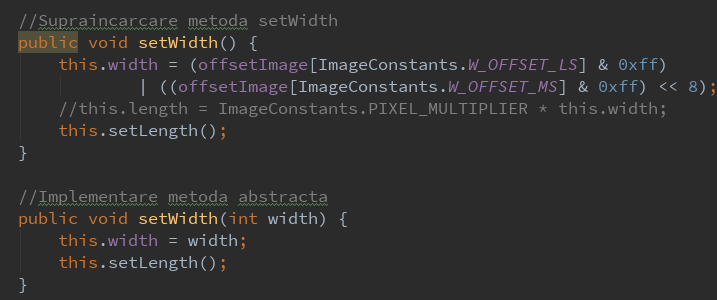
1. Apelarea lui “this”:



1. Apelarea lui “super”:



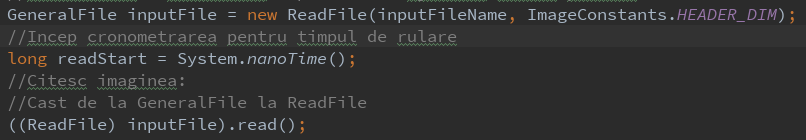
1. Supraincarcare:



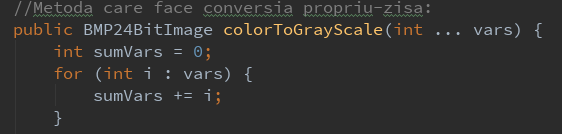
1. Initializarea unei clase parinte cu tipul clasei copil:



1. Downcast:



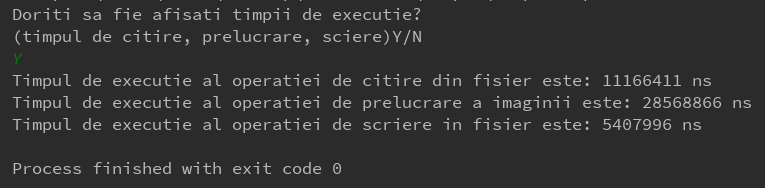
1. Varargs:



5. Evaluare performante

Deoarece proiectul are ca si scop, pe langa conversia propriu-zisa, exemplificarea conceptelor POO, aplicatia din punct de vedere al eficientei poate fi imbunatatita.

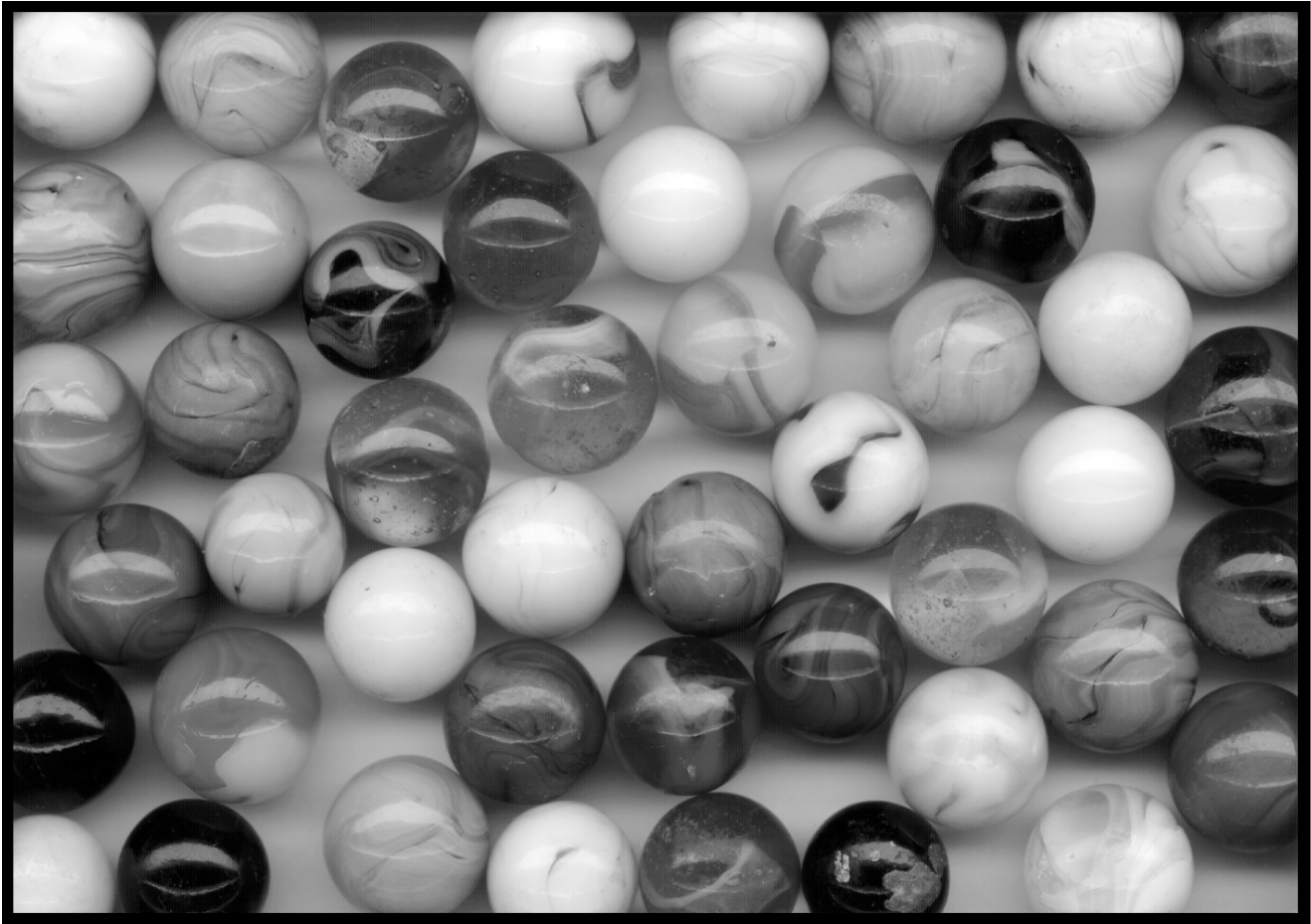
Cu toate acestea, timpul necesar prelucrarii unei imagini de dimensiune 1401x1000 sunt:



Prin acest screenshot am exemplificat si modul de luare a informatiei de la tastatura de catre utilizator.

Aplicatia merge doar pentru imagini BMP 24bit, dar poate fi foarte usor modificata pentru a accepta si alte formate. Diferenta este facuta de header-ul fiecarui format in parte.

In ceea ce priveste conversia imaginii, se poate observa ca imaginea este convertita complet din color in alb-negru:

🡺 

6. Concluzii:

In concluzie, aplicatia respecta toate cerintele impuse si este complet functionala. Prin aceasta aplicatie am reusit sa convertesc o imagine color intr-una alb-negru, folosind metoda mediei aritmetice, si sa exemplific conceptele de baza ale programarii orientate obiect.

7. Bibliografie:

a) Arhiva “Materiale Java” din cadrul site-ului acs.curs.pub.ro

Site-uri pentru diverse comenzi si biblioteci Java:

b) https://docs.oracle.com/javase/7/docs/api/java/nio/file/Path.html

c) https://docs.oracle.com/javase/7/docs/api/java/nio/file/Files.html

d) https://docs.oracle.com/javase/7/docs/api/java/io/FileOutputStream.html

e) https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html

f) https://docs.oracle.com/javase/7/docs/api/java/nio/file/Paths.html