

## DOCUMENTATIE

Am avut de realizat o aplicatie Java care sa realizeze o procesare de imagini.

Am inceput cu citirea imaginile din fisier.

Pentru a citi din fisier m-am folosit de o variabila string file care este citita de la tastatura de utilizator.

Am folosit clasa ImageIO.read(stream) pentru a stoca imaginea intr o variabila.

Am afisat si timpul de citire al imaginii in secunde , dupa care am facut o functie ce contine argumenti variabili ,

pentru a citi numele imaginii , pentru ca unele imagini pot avea in titlu mai multe cuvinte , fiecare fiind un string.

Dupa ce am citit imaginea si m-am asigurat ca a fost citita cu succes , am trecut la transformarea imaginii  
clasa ImageTransform mosteneste clasa ReadImageFromFile.

Am realizat o histograma dupa care am extras valorile intregi ale pixelilor ce corespund culorilor Rosu , Verde , Albastru folosind getRed(), getGreen(), getBlue(), getAlpha(),

care au valoare intre 0 si 255. Folosind histograma am setat pt fiecare culoare noile valori ale pixelilor , am convertit valorile in pixeli

dupa care am scris pixelii in imagine.

In functia equalizeHistogram() am creat un Lookup Table: ArrayList<int[]> imageLUT = new ArrayList<int[]>();

L-am umplut cu vectorii in care am stocat valorile fiecarei culori, dupa care am modificat acesti vectori cu factorul de scalare

si dupa care am returnat histograma.

Am calculat factorul de scalare: float scale\_factor = (float) (255.0 / (input.getWidth() \* input.getHeight()));

functia saveHistogram() returneaza un ArrayList cu valorile histogramei salvate pentru culorile RGB.

functia convertColorToRGB returneaza valorile R,G,B convertite la 8 biti prin shiftare.

am facut si o functie timpProcesare() care calculeaza timpul de procesare.

Pentru a scrie imaginea intr-un fisier am folosit clasa ImageIO din packageul jav.imageio.

Am folosit ImageIO.write(super.process(), "BMP", stream) pentru a scrie imaginea in format BMP , iar numele fisierului fiind preluat de

la tastatura , daca nu exista o cale , o sa returnam mesajul "Eroare in timpul scrierii in fisier".

Tot in aceasta clasa am calculat si timpul de scriere.

Am folosit cele 4 principii de POO: **Abstractizarea** prin clasa abstracta TimpProcesare si functiile abstracte din aceasta clasa: timpCitire(), timpProcesare().

Am folosit **Încapsularea** de exemplu in clasa CitesclImaginea , ascunzand variabile pentru a nu fi vizibile celorlalte clase , de exemplu : private long time.

Am folosit **Polimorfismul** , de exemplu functie timpProcesare() o gasim si in clasa CitesclImaginea, cat si in NormalizezCuloare.

Am folosit **Moștenirea** , de exemplu NormalizezCuloare extends CitesclImaginea arata faptul ca clasa NormalizezCuloare mosteneste CitesclImaginea.

Pentru a realiza tema , m-am documentat din urmatoarele surse:

[https://en.wikipedia.org/wiki/Color\\_normalization](https://en.wikipedia.org/wiki/Color_normalization)

<https://www.sciencedirect.com/science/article/pii/S1077314206002141>

<https://docs.oracle.com/javase/6/docs/api/java/awt/Color.html>

<https://docs.oracle.com/javase/7/docs/api/java/awt/image/BufferedImage.html>