

Proiect Java: Negativizarea unei Imagini

De Stoica Vlad Ionut 333AA

Negativizarea unei imagini consta in atribuirea fiecarui pixel din imagine valoarea ei “negativa”, ceea ce inseamna ca valorile celor trei canale de culoare (Red, Green si Blue, exista si al patrulea canal, Alpha, dar acesta nu este de culoare) sunt scazute din valoarea maxima pe care o pot lua, 255, acea valoare noua este apoi inlocuita in canalul respective, astfel rosu devine cian (albastru deschis) si verde devine magenta.

La apasarea butonului de run (asta sperand ca am pus totul cum trebuie in jar si proiectul s-a deschis cum trebuie), aplicatia va cere in command line 2 cai: una in care se afla poza de input (trebuie sa se introduca calea intreaga incluzand numele fisierului, mai putin extensia, aceasta e adaugata automat) si calea in care sa se scrie fisierul cu imaginea output (trebuie sa se introduca calea intreaga excluzand ultimul backslash si numele imaginii, acestea sunt adaugate automat), apoi sunt create trei obiecte, unul de tip Image, si obiectele Producator si Consumator care primesc obiectul de tip Image si calea lor respective, intr-un final, ambele primesc comanda run().

Clasa Image nu extinde nicio clasa, dar are campurile: o matrice de obiecte de tip Pixel, lungimea si latimea matricei, un Boolean si un int folosit la sincronizarea Threadurilor si un int static folosit in blocul static de initializare ce numara cate imagini au fost create

Aceasta clasa contine trei constructoare, unul cu niciun parametru, unul cu lungime si latime si unul cu un alt obiect de tip Image, mai multe gettere si settere, doua dintre care sunt sincronizate, de a fi folosite exclusive doar in Threaduri.

Clasa Pixel extinde clasa abstracta RGB si nu are campuri ale ei.

Are doua constructoare, unul cu patru parametrii int (canalele ARGB) si unul cu un singur parametru int, care ar reprezenta un pixel in intregime (desi formatul BMP nu are cel de-al patrulea canal, Alpha, functia de citire a fisierelor imaginie tot ofera acest canal, asadar, de dragul sigurantei, am decis sa-l includ in clase); si functia Negate, ce negativizeaza valorile celor trei canale de culoare.

Clasa RGB implementeaza interfetele Red, Green si Blue, campurile sale sunt patru variabile int pentru cele patru canale si un public static final ce reprezinta valoarea maxima pe care canalul Alpha il poate avea.

Nu are constructoare, doar gettere si settere pentru canale; si o functie abstracta, Negate.

Interfetele Red, Green si Blue are fiecare un public static final ce reprezinta valoarea maxima pe care canalul lor respectiv il poate avea

Metodele ce sunt de implementat de alte clase sunt un getter si un setter pentru canalul lor respectiv.

Intr-un final, Clasele Consumer si Producer: prima clasa are un camp de tip Image pentru imaginea pe care o va primi si prelucra si un string care va reprezenta calea fisierului; Producer are si el un Image si un string dar si campuri de tip BufferedImage si File pentru citirea fisierului imagine.

Ambele au un singur constructor in care completeaza stringul filePath, in acesta Producer citeste fisierul imagine din noul filePath si redimensioneaza obiectul de tip Image pentru a fi de aceeasi marime ca imaginea citita.

Pe langa un bloc de initializare, singur alta metoda este run care creeaza Threadurile.

In Threadul Producer, obiectul de tip Image primeste, licate o linie pe rand, valorile pixelilor din fisierul imagine.

In Threadul Consumer, citeste, cate o linie pe rand, Pixelii din obiectul de tip Image, apoi le negativizeaza valoarea si stocheaza valoarea noua intr-un obiect de tip BufferedImage. Cand toata imaginea a fost citita si prelucrata, obiectul de tip BufferedImage este salvat intr-un fisier la calea introdusa la inceput.

Pentru a asigura folosirea sincronizata a obiectului de tip Image intre Producer si Consumer, getterul si setterul implementate special pentru cele doua clase numara de cate ori sunt folosite, astfel incat sa fie folosite doar de cate ori cat este imaginea de lunga (cate o linie pe rand), apoi campul boolean isi schimba valoarea si notifica toate Threadurile pentru ca cealalta clasa sa inceapa sa foloseasca obiectul de tip Image.

Output:

Before:



After:



Librariile importate pentru citirea/scrierea fisierelor imagine sunt:

- `java.awt.image.BufferedImage`
- `java.io.File`
- `java.io.IOException`
- `javax.imageio.ImageIO`

Acestea au fost folosite in clasele `Producer` si `Consumer`

S-a mai importat si libraria `java.util.*` in clasa `Main_Class` pentru a putea folosi clasa `Scanner` folosita la citirea intrarilor utilizatorilor.