

A dark blue vertical bar on the left side of the slide. A blue arrow points to the right from the bar, containing the date.

1/28/2022

Image Brightness modification

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and curve upwards and to the right.

Andrei-Daniel SAMOILĂ 333AC

FACULTATEA DE AUTOMATICA SI CALCULATOARE

Contents

Introducere:	2
Partea teoretica	2
Descrierea implementarii.....	2
Descrierea Structurala.....	2
Evaluare performante:	8
Concluzii:	8
Bibliografie.....	9

Introducere:

Proiectul are ca tema modificarea luminozitatii unei imagini(.jpg, .jpeg, .bmp).

Datele imaginii intr-un fisier BMP sunt de obicei compresate folosind o tehnica numita Lossless Compression (compresie fara pierderi de date).

Partea teoretica:

Un tip de fisier cu extensia .bmp este alcatuit dintr-un Header si un Array de Pixeli.

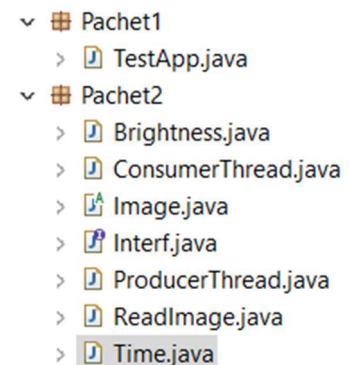
Asadar, algoritmul aduna fiecare byte corespunzator Red, Green si Blue unui pixel pentru a-i creste luminozitatea fara a depasii valoarea de 255 (maximul unui byte) sau 0 (minimul unui byte) pentru a nu se produce overflow/underflow. Fiecare byte este fara semn.

Descrierea implementarii:

Citire informatii de identificare fisier sursa (fisier sursa) si citire informatii de identificare fisier destinatie - citire fisier sursa - procesare imagine - scriere fisier destinatie - inregistrare timp de executie fiecare etapa - afisare rezultate timp de procesare fiecare etapa

Descrierea Structurala:

Numele pozei ce se doreste a modifica este introdus de la tastatura, la fel ca si cel in care vom salva noua poza. Tema este realizata prin intermediul interfetei Interf si a sapte clase: Image, ReadImage, Brightness, Time, ConsumerThread, ProducerThread si TestApp, impartite in doua pachete, unul de testare si celalalt de prelucrare.



Interfata Interf contine metodele din clasa Time si ajuta la organizarea codului.

```

1 package Pachet2;
2
3
4 public interface Interf {
5     double getTime();
6     public void setStartTime(long startT);
7     public long getStartTime();
8     public void printTimeConversie();
9     public void printTimeTest();
10    public void printTimeDeschidereFisier();
11    public void printTimeCitireFisier();
12 }
13

```

Clasa Image este o clasa abstracta ce continue attributele width, height, imaginea ce va fi modificata, precum si metodele de deschidere fisier, respective citire din fisier.

```

1 package Pachet2;
2
3 import java.awt.image.BufferedImage;
4
5 public abstract class Image {
6
7     double width;
8     double height;
9     BufferedImage image;
10
11     abstract void ReadFromFile(String s);
12     abstract String OpenFile();
13
14     Image() {
15     }
16 }
17

```

Clasa ReadImage este derivata clasei abstracte Image si sunt implementate metodele respective.

```

1 package Pachet2;
2 import javax.imageio.ImageIO;
3
4
5 public class ReadImage extends Image {
6     Time time=new Time();
7     public ReadImage() throws InterruptedException {
8     }
9
10    public void ReadFromFile(String s ) {
11        synchronized(this) {
12            time.setStartTime(System.nanoTime());
13            try {
14                //Se creeaza alt fisier cu pathul s
15                File input = new File(s);
16                this.image = ImageIO.read(input);
17                //Se citeste imaginea
18                this.width = (double) this.image.getWidth();
19                //Se preiau datele din imagine
20                this.height = (double) this.image.getHeight();
21            } catch (IOException var8) {
22                System.out.println("Fisierul nu exista.");
23                System.exit(0);
24            }
25            time.printTimeCitireFisier();
26            //Afisare timp citire
27        }
28    }
29 }

```

```

29 public String OpenFile() {
30     synchronized(this) {
31         String filePath=null;
32         try {
33             InputStreamReader isr = new InputStreamReader(System.in);
34             BufferedReader br = new BufferedReader(isr);
35             System.out.println("Numele imaginii (nume.bmp):"); //imaginea ce urmeaza a fi prelucrata
36
37             for (filePath = br.readLine().toString(); !filePath.endsWith(".bmp"); filePath = br.readLine())
38                 System.out.println("Fisierul nu are extensia potrivita. Introdueceti un fisier cu extensia .bmp");
39         }
40         //Fisierul citeste informatia pana cand numele este
41         //se termina cu extensia potrivita
42     } catch (IOException var8) {
43         System.out.println("Nu s-a realizat citirea in mod corect");
44     }
45     return filePath;
46 }
47 }
48 }
49

```

In clasa Time, care implementeaza interfata Interf se calculeaza timpul corespunzator fiecarui task.

```

public class Time implements Interf{
    private long startTime;
    @Override
    public double getTime() {
        //Calcularea duratei
        long endTime = System.nanoTime();
        double totalTime = (double)(endTime - startTime);
        //System.out.println("Timp de executie conversie : " + totalTime / 1.0E9D + " secunde");
        return totalTime / 1.0E9D;
    }
    public void setStartTime(long startT) {
        this.startTime=startT;
    }
    public long getStartTime() {
        return this.startTime;
    }
    public void printTimeConversie() {
        System.out.println("Timp de executie conversie : " + this.getTime() + " secunde");
    }
    public void printTimeTest() {
        System.out.println("Timp de executie Test : " + this.getTime() + " secunde");
    }
    public void printTimeDeschidereFisier() {
        System.out.println("Timp de executie pentru deschiderea si\n citirea pathului"
            + "unui fisier : " + this.getTime() + " secunde");
    }
}

```

In clasa Brightness se vor aplica modificarile de luminozitate asupra imaginii si va fi returnata imaginea finala, cu modificarile luminozitatii.

```
}  
public BufferedImage ConvertToBrightened() {  
  
    synchronized(this) {  
        timeConversie.setStartTime(System.nanoTime());  
  
        //Se modifica parametrii pozei  
        for(int y = 0; (double)y < this.bHeight; ++y) {  
            for(int x = 0; (double)x < this.bWidth; ++x) {  
                int pixel = this.brightenedImage.getRGB(x, y);  
                int alpha = pixel >> 24 & 255;  
                int red = pixel >> 16 & 255;  
                int green = pixel >> 8 & 255;  
                int blue = pixel & 255;  
                red = Math.min(255, red + 63);  
                green = Math.min(255, green + 63);  
                blue = Math.min(255, blue + 63);  
                int newPixel = alpha << 24 | red << 16 | green << 8 | blue;  
                this.brightenedImage.setRGB(x, y, newPixel);  
            }  
        }  
  
        //Se afiseaza timpul de conversie  
        this.timeConversie.printTimeConversie();  
        return this.brightenedImage;    //poza prelucrata  
    }  
}  
  
    //metode de get si set:  
public void setHeight(double height) {
```

In clasa TestApp se apeleaza doua threaduri:

- ProducerThread, in care se deschide fisierul de citire:

```
private Brightness obiect;
private ReadImage imgob;

public ProducerThread(Brightness ob, ReadImage img) {
    this.obiect=ob;
    this.imgob=img;
}
public void start() {
    super.start();
}

public void run() {
    try {
        synchronized(this) {
            Time timeDeschidere=new Time();

            timeDeschidere.setStartTime(System.nanoTime());
            obiect.setFilePath(imgob.OpenFile());
            timeDeschidere.printTimeDeschidereFisier();

            Thread.sleep(1000);
        }
    } catch (InterruptedException e)
    {
        e.printStackTrace();
    }
}
```


- ConsumerThread, in care se citesc datele din fisierul deschis:

```
package Pachet2;

public class ConsumerThread implements Runnable {
    private Brightness obiect;
    private ReadImage imgob;

    public ConsumerThread(Brightness ob, ReadImage img) {
        this.obiect=ob;
        this.imgob=img;
    }

    public void run() {

        //citire din fisier
        imgob.ReadFile(obiect.getFilePath());
        //setarile pentru height,width
        //buffer
        obiect.setHeight(imgob.height);
        obiect.setWidth(imgob.width);
        obiect.setBufferedImage(imgob.image);

    }
}
```

```
22 consumerThread procesare = new ConsumerThread(citire, imgob);
23 Thread threadProcesare = new Thread(procesare);
24 //Pornirea threadului citire
25 citire.start();
26 //citire.notify(); //Permite executia altui thread doar cand se termina acesta
27 citire.join(); //Se porneste threadul de procesare
28 threadProcesare.start();
29
30
31 Time time=new Time();
32 String outputFilePath = null;
33
34 ReadImage fisier=new ReadImage();
35 System.out.print("Noul fisier: ");
36 //Se preia path-ul noului fisier
37 outputFilePath=fisier.OpenFile();
38 //Se creeaza noul fisier
39 File output = new File(outputFilePath);
40
41 time.setStartTime(System.nanoTime());
42 //Se adauga in fisierul creat
43 //poza modificata corespunzator
44 try {
45     ImageIO.write(obiect.ConvertToBrightened(), "bmp", output);
46 } catch (IOException e) {
47     // TODO Auto-generated catch block
48     e.printStackTrace();
49 }
50 System.out.println("Conversie reusita.");
```


Evaluare performante:

```
Problems @ Javadoc Declaration Console ×
<terminated> TestApp [Java Application] C:\Users\Samoi\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v2021111
Numele imaginii (nume.bmp):
model.bmp
Timp de executie pentru deschiderea si
citirea pathului unui fisier : 3.9030843 secunde
Noul fisier: Numele imaginii (nume.bmp):
Timp de executie citire din fisier: 0.1133058 secunde
final.bmp
Timp de executie conversie : 0.0281811 secunde
Conversie reusita.
```

Sunt masurati timpii de executie atat pentru citirea unei imagini, prelucrarea ei cat si pentru scrierea noii imagini.

Concluzii:

Model:



Final:



Metoda aleas este una eficienta cu un timp de executie mic.

Bibliografie

<https://stackoverflow.com/questions/12980780/how-to-change-the-brightness-of-an-image>

<https://www.geeksforgeeks.org/java-program-to-increase-or-decrease-brightness-of-an-image/>

<https://stackoverflow.com/questions/5657469/how-do-i-go-about-adding-an-image-into-a-java-project-with-eclipse>