Proiect Java

Tudorin Dan--333—AA

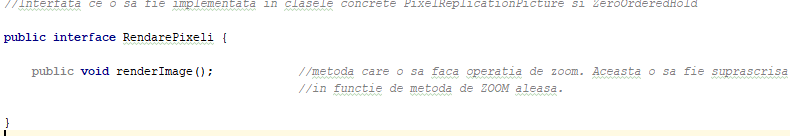
Aplicatia creata vizeaza prelucrarea de imagini. In cazul de fata procesele dezvoltate de aplicatie sunt de ZOOM. Programul contine 2 metode de realizare a ZOOM – IN-ului asupra unei fotografii: Pixel Replication si metoda Zero Order Hold. Acestea realizea in mod diferit si total independent si decuplat functia de zoom asupra unei fotografii.

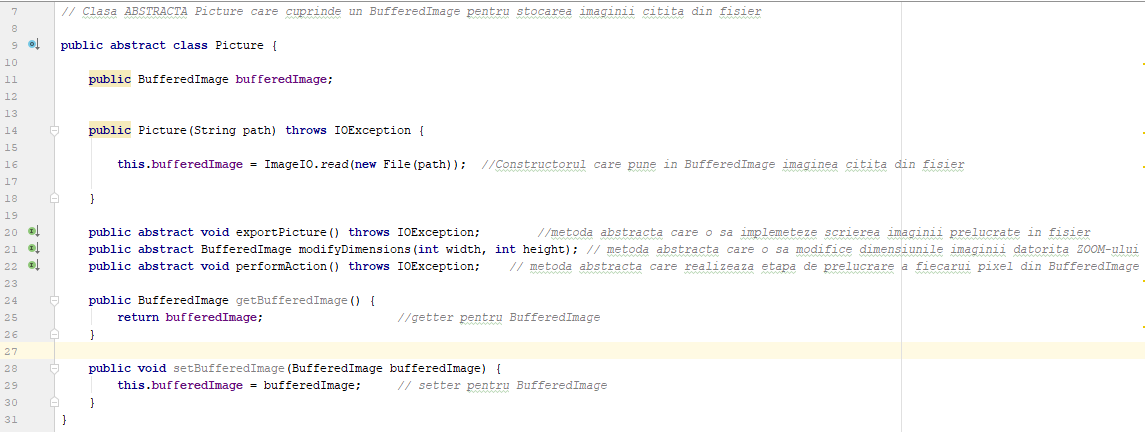
Mediul de dezvoltare utilizat este desigur Java v1.8, iar ca IDE am folosit platforma INTELLIJ de la cei de la JetBrains. IDE-ul pune la dispozitie programatorului toate toolurile necesare pentru realizare oricarui concept de java challenge. Este prietenos si fiabil, de aceea pot spune ca alegerea mea a fost una justificata fata de alte IDE-uri.

Revenind la aplicatia creata, aceasta este compusa din punct de vedere structural de urmatoarele:

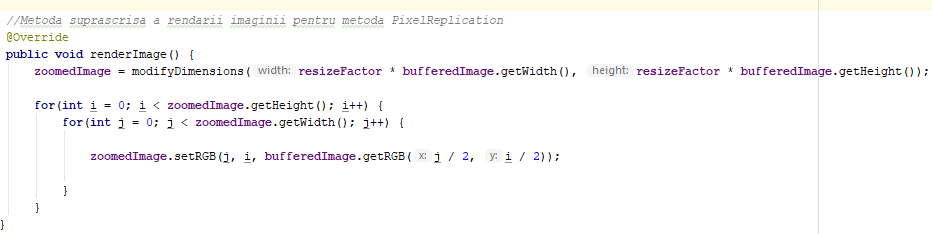
* Interfata “RendarePixeli” cu metoda abstrcta (implicita) **public void** renderImage();
* Clasa abstracta “Picture” cu metodele sale abstracte
  + **public abstract void** exportPicture() **throws** IOException;
  + **public abstract** BufferedImage modifyDimensions(**int** width, **int** height);
  + **public abstract void** performAction() **throws** IOException;
* Clasa extinsa “PixelReplicationPicture” in care se suprascriu toate metodele abstracte mostenite de la clasa abstracta si de la interfata implementata;
* Clasa extinsa “ZeroOrderedHold” in care se suprascriu toate metodele abstracte mostenite de la clasa abstracta si de la interfata implementata;
* Clasa Main in care se va executa programul principal.

In cele ce urmeaza o sa exemplificam putin mai detaliat cu ajutorul unor imagini structura si continutul acestor clase.





Pixel Replication Method:



INTRODUCTION:

It is also known as Nearest neighbor interpolation. As its name suggest , in this method , we just replicate the neighboring pixels. As we have already discussed in the tutorial of Sampling , that zooming is nothing but increase amount of sample or pixels. This algorithm works on the same principle.

WORKING:

In this method we create new pixels form the already given pixels. Each pixel is replicated in this method n times row wise and column wise and you got a zoomed image. Its as simple as that.

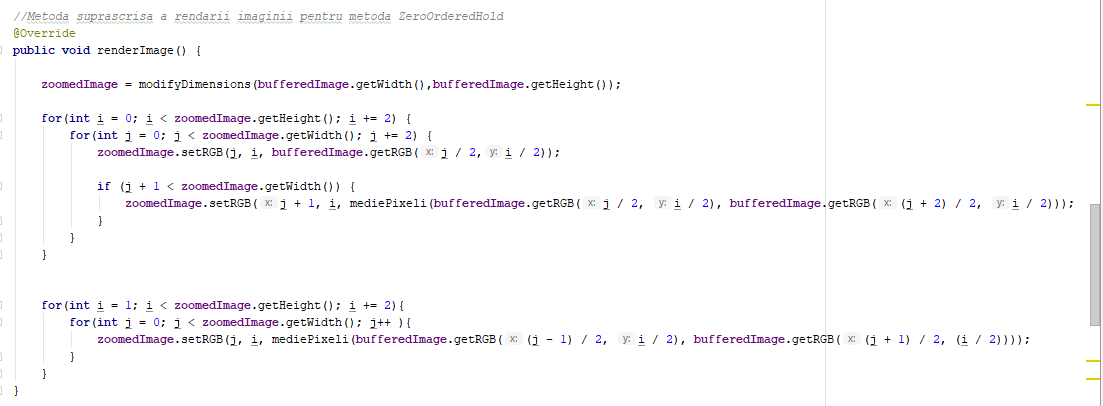
FOR EXAMPLE:

if you have an image of 2 rows and 2 columns and you want to zoom it twice or 2 times using pixel replication, here how it can be done.

For a better understanding , the image has been taken in the form of matrix with the pixel values of the image.

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |

Zero Order Hold Method:



De asemenea aceasta metoda contine o metoda auxiliara de realizare a mediei pentru doi pixeli. In cele ce urmeaza o sa exemplificam si de ce avem nevoie de aceasta:

**private int** mediePixeli(**int** pixel1, **int** pixel2) {  
 **return** (**int**) (((((pixel1) ^ (pixel2)) & 0xfffefefeL) >> 1) + ((pixel1) & (pixel2)));  
}

Aceasta prelucreaza pixelii la nivel de bit si realizeaza conversia.

INTRODUCTION

Zero order hold method is another method of zooming. It is also known as zoom twice. Because it can only zoom twice. We will see in the below example that why it does that.

WORKING

In zero order hold method , we pick two adjacent elements from the rows respectively and then we add them and divide the result by two, and place their result in between those two elements. We first do this row wise and then we do this column wise.

FOR EXAMPLE

Lets take an image of the dimensions of 2 rows and 2 columns and zoom it twice using zero order hold.

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |

First we will zoom it row wise and then column wise.

ROW WISE ZOOMING

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 2 |
| 3 | 3 | 4 |

As we take the first two numbers : (2 + 1) = 3 and then we divide it by 2, we get 1.5 which is approximated to 1. The same method is applied in the row 2.

COLUMN WISE ZOOMING

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 2 |
| 2 | 2 | 3 |
| 3 | 3 | 4 |

We take two adjacent column pixel values which are 1 and 3. We add them and got 4. 4 is then divided by 2 and we get 2 which is placed in between them. The same method is applied in all the columns.

NEW IMAGE SIZE

As you can see that the dimensions of the new image are 3 x 3 where the original image dimensions are 2 x 2. So it means that the dimensions of the new image are based on the following formula

(2(number of rows) minus 1) X (2(number of columns) minus 1)

ADVANTAGES AND DISADVANTAGE.

One of the advantage of this zooming technique , that it does not create as blurry picture as compare to the nearest neighbor interpolation method. But it also has a disadvantage that it can only run on the power of 2. It can be demonstrated here.

REASON BEHIND TWICE ZOOMING:

Consider the above image of 2 rows and 2 columns. If we have to zoom it 6 times , using zero order hold method , we can not do it. As the formula shows us this.

It could only zoom in the power of 2 2,4,8,16,32 and so on.

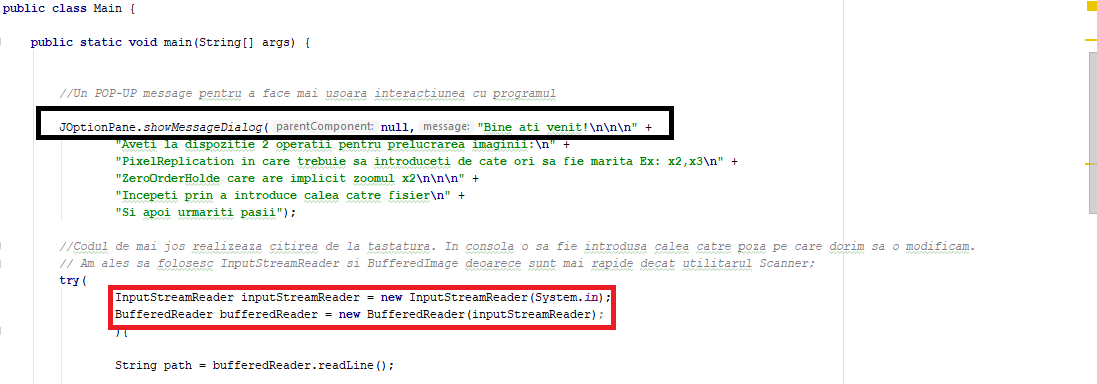
Even if you try to zoom it, you can not. Because at first when you will zoom it two times, and the result would be same as shown in the column wise zooming with dimensions equal to 3x3. Then you will zoom it again and you will get dimensions equal to 5 x 5. Now if you will do it again, you will get dimensions equal to 9 x 9.

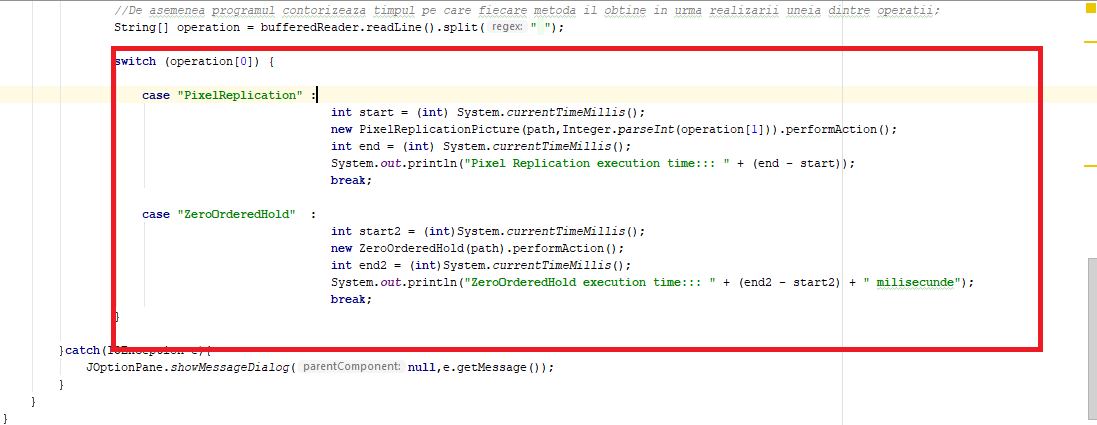
Whereas according to the formula of yours the answer should be 11x11. As (6(2) minus 1) X (6(2) minus 1) gives 11 x 11.

Clasa Main:

Aceasta pune cap la cap intregul program astfel: cu ajutorul JOPTIONPANE-urilor utilizatorul este ghidat de-a lungul programului. Pentru citirea caii catre imaginea dorita o sa folosim clasele InputStreameader si BufferedReader pentru o citire mult mai rapida. Odata introdusa calea catre imaginea dorita, utilizatorul selecteaza metoda de prelucrare dorita. Aceste comenzi se introduc in consola sistemului pe etape. Odata selectata metoda dorita conform helperului creat utilizatorul o sa aiba contact direct cu poza modificata gata transferata catre un fisier nou in folderul proiectului.

Metodele se disting intr-un switch in care sunt contorizate de asemenea si timpii de executie pentru fiecare metoda in parte.





Referinte : http://www.tutorialspoint.com/dip/Zooming\_Methods.htm