Proiect IGU

Aplicatie pentru crearea unei histograme a unei imagini alb-negru cu interfata realizata in JavaFX

# Cuprinsul lucrarii

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

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

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

4.Descrierea aplicatiei...................................................................................4

5.Descrierea modulelor.................................................................................5

6.Descrierea controalelor utilizate................................................................6

7.Evaluare performante................................................................................12

8.Concluzii.....................................................................................................12

9.Bibliografie.................................................................................................12

**1.Introducere**

In cadrul acestui proiect,am avut de realizat o procesare de imagini.Temele pentru proiect au fost multiple.Am avut de realizat tema cu numarul 12-Gray Level Histogram of a Gray-Scale Image.

Am realizat acest proiect in cadrul IDE-ului Eclipse ,cu ajutorul limbajului Java.

Apoi,pentru interfata vizuala ,am folosit JavaFX.

**2.Descrierea aplicatiei cerute**

Pentru realizarea proiectului conform temei avute,a trebuit sa implementez o aplicatie care sa realizeze urmatoarele etape:

-citire de la tastatura informatii de identificare a fisierului unde se afla imaginea(tip BMP),mai exact am realizat cu ajutorul JavaFX o fereastra de alegere a fisierului BMP;

-citire fisier;

-procesare imagine astfel incat sa iau datele pentru o histograma a imaginii, aceste date fiind afisate in cadrul aplicatiei noastre in cadrul histogramei;

-citire de la tastatura a informatiilor de identificare a fisierului destinatie,mai exact am realizat in interfata cu utilizatorul o fereastra de creare/alegere a fisierului destinatie;

-realizare in acel fisier a histogramei necesare;

-inregistrare si afisare timp de executie in consola pentru etapele anterioare,excluzandu-le pe cele de citire informatii de cautare si de salvare,intrucat timpul acelora depinde de utilizator.

De asemenea,am indeplinit si cerintele cerute pentru aplicatie(tratare exceptii,includere concept POO,precum si celelalte).

**3.Partea teoretica**

Referitor la conceptele folosite,am folosit urmatoarele :

- clase si clase abstracte in Java (pentru realizarea polimorfismului si abstractizarii);

- importare a membrilor de pachete(pentru preluarea unor obiecte /functii aflate acolo);

- variabile pentru temporizare,citire din fisier,scriere,etc(pentru realizarea diferitelor operatii);

-blocuri try-catch(pentru prinderea de exceptii);

-functii(pentru realizarea de operatii in program);

-clase si obiecte specifice JavaFX,impreuna cu functiile specifice,pentru crearea interfetei grafice cu utilizatorul(GUI) ;

-main(pentru rularea propriu-zisa a programului);

**4.Descrierea aplicatiei**

**Structura**

Din punct de vedere al structurii ,aplicatia este formata din 5 clase,dintre care una abstracta.

Prima clasa,clasa abstracta numita “ Citire\_abs”,a fost realizata pentru a indeplini urmatoarele concepte de POO legate de abstractizare si polimorfism .

Cea de-a doua clasa, “Init\_citire”,a fost realizata pentru a se citi infromatiile din imagine referitoare la inaltime,latime si bitii de padding(necesari citirii corecte a imaginii).Clasa contine functia void f\_Init\_citire(String cale) in cadrul careia se realizeaza operatiile prezentate in propozitia anterioara,prinderea de exceptii si masurarea timpului necesar acestei operatii de citire.

In cadrul celei de-a treia clase,”Citire\_date\_histograma”,se citesc datele( R,G si B) din pixelii imaginii, necesare pentru datele histogramei.

In cadrul celei de-a patra clase,” Creare\_salvare\_histograma” ,exista functii pentru realizarea graficului histogramei ,impreuna cu functii pentru calcularea valorii maxime din datele histogramei si pentru salvarea graficului ca imagine.

Ultima clasa,” Main”,este folosita pentru scrierea instructiunilor ce duc la rularea programului pentru a realiza etapele cerute si afisarea interfetei grafice ,avand in ea functia main(care porneste practic aplicatia) si functia void start(folosita pentru aplicatia grafica si scrierea modului de folosire a celorlalte clase).

**Arhitectura**

Arhitectura aplicatiei consta din urmatoarele clase :

JPanel

(importata)

Citire\_abs

(abstracta)

Init\_citire

Citire\_date\_ histograma

Creare\_ Salvare\_ Histograma

Main\_

In fiecare clasa ,exista functii specifice pentru realizarea operatiilor specifice fiecareia.

**Functionarea**

Din punct de vedere al functionarii,se pot spune urmatoarele:

-sunt importate din pachete functii,clase,etc,necesare fiecarei clase pentru a functiona corespunzator(inclusiv pentru functionarea corecta a interfetei grafice);

-functia main incepe rularea programului(executa start);

-functia start porneste executia programului propriu-zis impreuna cu interfata grafica corespunzatoare ,in aceasta functie existand instante ale claselor care ajuta la realizare scopului principal al programului si la vizualizarea interfetei grafice necesare;

-din start,se citesc caile de citire a imaginii si de scriere a histogramei in fisier cu ajutorul variabilelor si functiilor din importuri si a elementelor de JavaFX;

-de asemenea ,in start ,se apeleaza functii ale instantelor claselor sau ale claselor care sa duca la realizarea scopului programului;

-in start ,se mai realizeaza operatii de masurare a timpului necesar pentru fiecare operatie importanta;

-din start,prin intermediul chemarii functiilor claselor sau ale instantelor ,se ruleaza acele functii pentru obtinerea rezultatelor dorite .

Se folosesc din pachete functii sau obiecte referitoare la urmatoarele:

-import/export de fisiere;

-afisare elemente vizuale in interiorul interfetei vizuale;

-afisare ferestre de selectie pentru alegere/salvare de fisiere;

- generarea de imagini;

-aruncarea de exceptii;

**5.Descrierea modulelor**

Citire\_abs.java

- contine clasa abstracta”Citire\_abs”:

-contine functia “f\_Init\_citire(string):void” (fara implementare propriu-zisa).

Init\_citire.java

-contine clasa “Init\_citire”:

-contine variabilele “height”,”width”,”paddingBytes”(inaltime,latime,biti de padding);

-contine functia “f\_Init\_citire(string):void” (implementeaza citirea inaltimii,latimii si bitilor de padding ,adica a celor necesare citirii corecte a codului RGB a pixelilor imaginii).

Citire\_date\_histograma.java

-contine clasa “Citire\_date\_histograma”:

-contine functia”f\_Citire\_date\_histograma(int,int,int,String):int[]” (realizeaza operatia de citire a codulul RGB a pixelilor si incadrarea pe niveluri de gri in cadrul vectorul de date pentru histograma).

Creare\_Salvare\_Histograma.java

-contine clasa “Creare\_Salvare\_Histograma”:

-contine variabilele data,serialVersionUID,PAD,SPAD;

-contine functiile “paintComponent(Graphics):void”(desenare graficului pe baza datelor histogramei),”getMax():int”(calculeaza maximul din vectorul de date al histogramei),”getScreenShot(Component):BufferedImage”(trece imaginea histogramei desenate intr-un buffer), “SaveScreenShot(Component,String):void”(primeste buffer-ul si creeza fisierul cu imaginea).

Main\_class.java

-contine clasa “Main”:

-contine functia main(String[])(se ocupa cu rularea functiei „start”) si functia start(Stage)(executa propriu-zis programul ,realizandu-se in ea si interfata grafica).

Mai multe detalii ,se gasesc in comentariile codului sau puteti afla direct de la realizatorul aplicatiei.

**6.Descrierea controalelor utilizate**

In cadrul partii de realizare a interfetei grafice,am folosit mai multe obiecte,printre care mai multe scene,containere,butoane,lista de tab-uri,etc.

Am folosit ca si containere pentru elementele grafice obiecte de tip „Grid Pane” ,”HBox” ,”VBox” sau „FlowPane”.Aceste containere le-am integrat in scene(de exemplu ,in codul meu ,numite „scene1”,”scene2” sau altfel) ,dupa care scenele le-am afisat cu ajutorul stage-urilor(numite si „ecrane” sau „windows” in engleza).

Aplicatia mea contine doua astfel de stage-uri(„ecrane”) principale numite „primaryStage” si „stage” ,ambele facute manual .De asemenea ,mai apar si alte ecrane in urma apasarii butoanelor denumite in cod ca „buton\_alegere\_imagine” si „saveBtn”,aceste ecrane avand ca scop alegerea imaginei cu extensia BMP pentru care se doreste histograma, respectiv a locatiei si numelui fisierului unde se va salva histograma respectiva.

In cadrul tuturor scenelor,am integrat elemente pentru interfata cu utilizatorul.Controalele utilizate in cadrul interfetei vizuale(toate facute manual) sunt urmatoarele:

1.Check Box



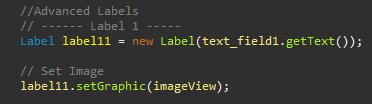


2.Simple Label





3.Advanced Label





4.Text Field





5.Tab







6.Tool Bar





7.Hyperlink





8.Choice Box





9.Toggle Button





10.Button(normal)



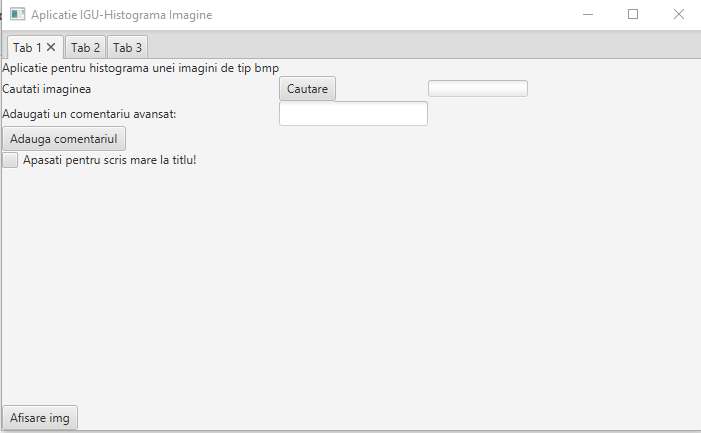


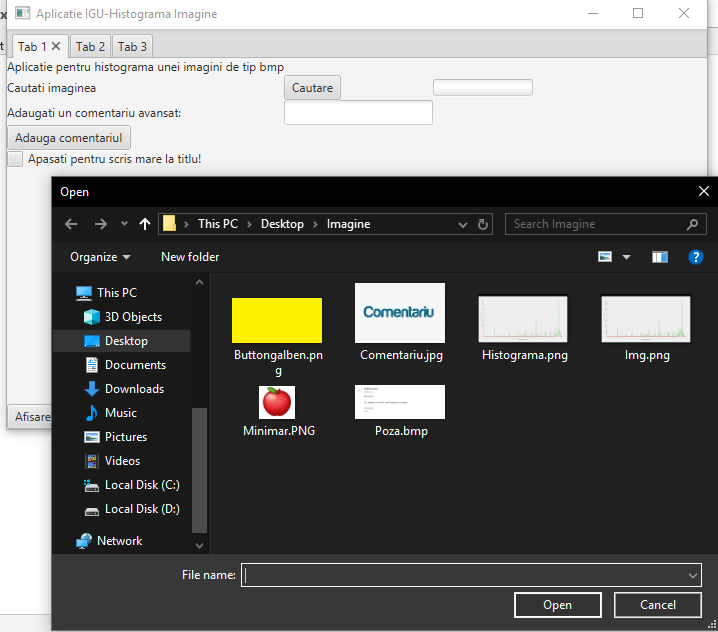
11.Progress Bar

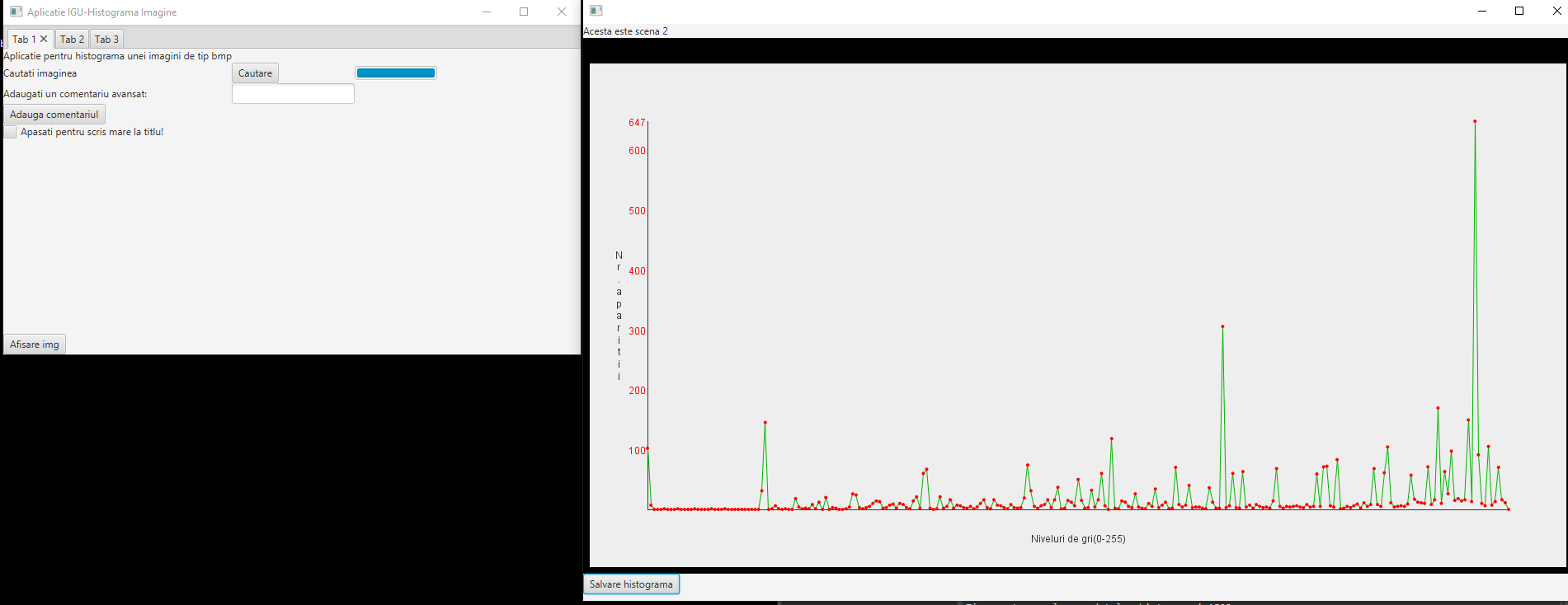


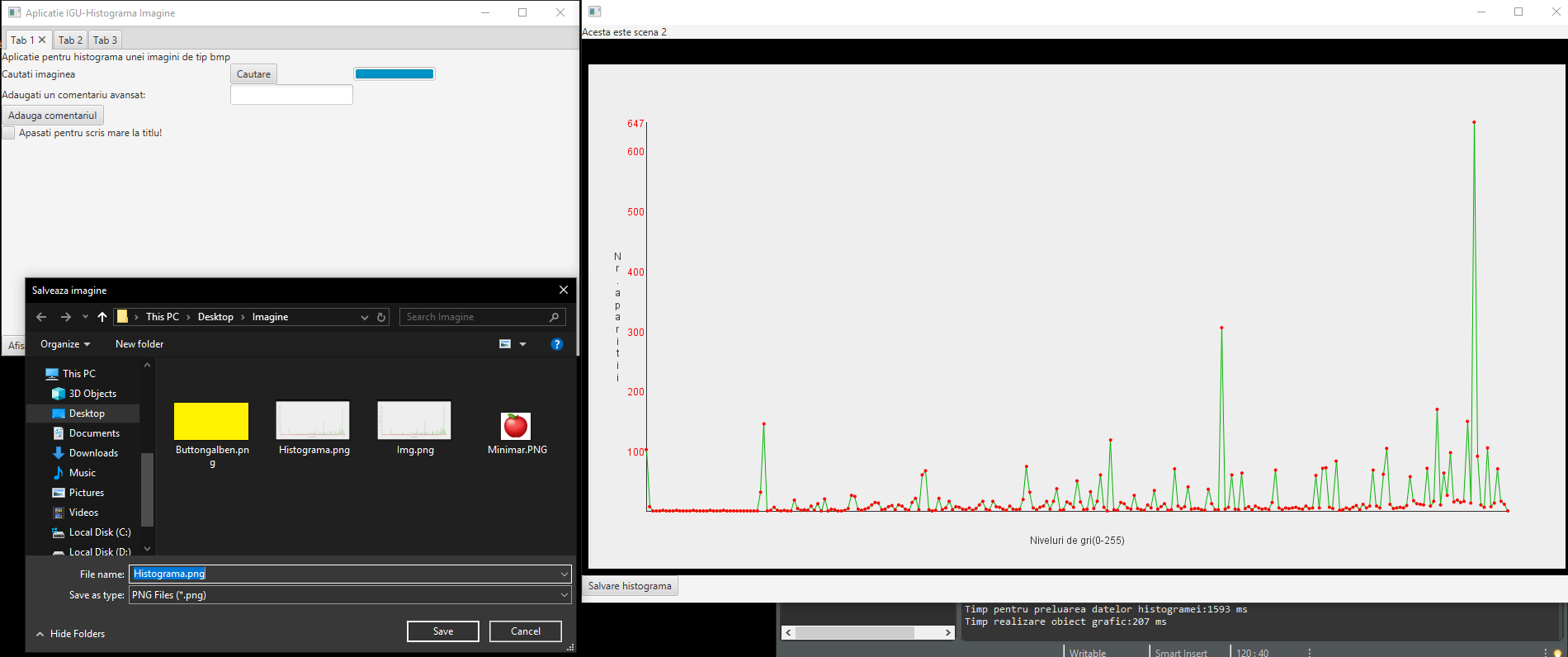


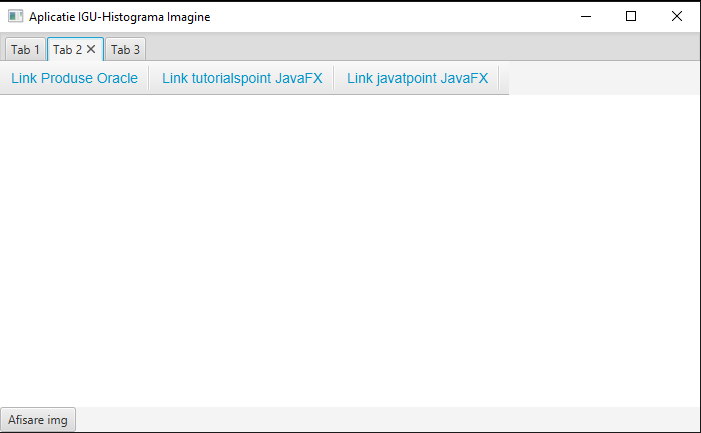
Aplicatia in pasii principali de executie arata asa:



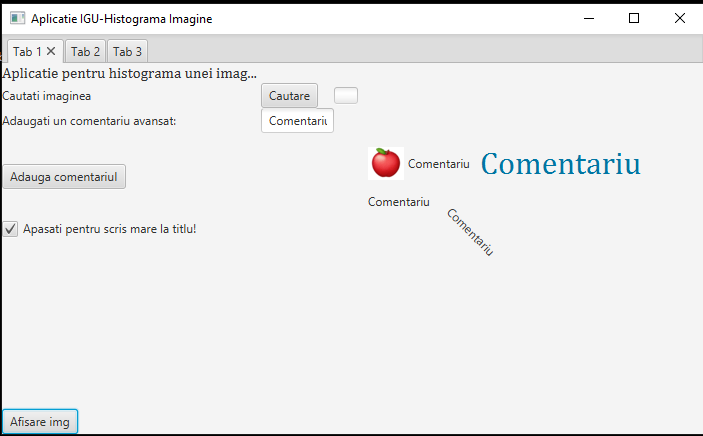












Din imaginile anterioare ale aplicatiei,se observa ca aceasta are mai multe ecrane(„windows”) in cadrul executiei,toate fiind realizate manual si toate tipurile de controale mentionate anterior(mai mult de 10).

**7.Evaluare performante**

Performantele aplicatiei pot fi evaluate dupa timpul de finalizare a diferitelor operatii.

Astfel ,timpul de citire a destinatiei difera dupa viteza de alegere a fisierului de catre utilizator.

Timpul de citire a informatiilor precum inaltimea sau latimea imaginii este de cateva milisecunde(pana in 10 de cele mai multe ori).

Timpul pentru realizarea vectorului de date a histogramei este de aproximativ 1000 ms.

Timpul pentru realizarea graficului aferent histogramei (+afisare pe ecran)este de cateva sute de secunde(aprox.200 ms).

Timpul pentru citirea caii destinatiei este diferit dupa viteza de scriere a utilizatorului.

In concluzie,consider ca performantele aplicatiei sunt foarte bune.

**8.Concluzii**

Aplicatia realizeaza scopul propus al temei.

Pornind de la o imagine alb-negru,aplicatia afiseaza un grafic care arata numarul de pixeli aferenti celor 256 de niveluri de gri si il salveaza intr-un fisier bmp.

Timpul de realizare a tuturor operatiilor necesare pentru realizarea scopului temei este bun.

Aplicatia este multimodulara si realizeaza si alte cerinte ale aplicatiei.

Se respecta etapele de executie cerute .

Se respecta implementarea in cadrul interfetei vizuale a aplicatiei a cel putin 10 controale si cel putin 2 ecrane,din care cel putin 2 controale sunt realizate manual,prin cod(in cazul meu,toate) impreuna cu cel putin un ecran realizat manual(in cazul meu ,doua ecrane sunt realizate manual).

Partea vizuala a aplicatiei este realizata in JavaFX si implementeaza ,la baza, tema mea de prelucrare a imaginii realizata pentru cursul de AWJ, la care adauga si functionalitati extra pentru realizarea cerintelor specifice exact acestei tema de la IGU.

**9.Bibliografie**

Pentru partea initiala(cea din tema de la AWJ):

[**https://www.tutorialspoint.com/java/java\_exceptions.htm**](https://www.tutorialspoint.com/java/java_exceptions.htm)

[**https://stackoverflow.com/questions/7209110/java-util-nosuchelementexception-no-line-found**](https://stackoverflow.com/questions/7209110/java-util-nosuchelementexception-no-line-found)

[**https://www.tutorialspoint.com/java\_dip/java\_buffered\_image.htm**](https://www.tutorialspoint.com/java_dip/java_buffered_image.htm)

[**https://stackoverflow.com/questions/2601365/padding-in-24-bits-rgb-bitmap**](https://stackoverflow.com/questions/2601365/padding-in-24-bits-rgb-bitmap)

[**https://stackoverflow.com/questions/3382954/measure-execution-time-for-a-java-method**](https://stackoverflow.com/questions/3382954/measure-execution-time-for-a-java-method)

[**https://www.mkyong.com/java/how-to-write-an-image-to-file-imageio/**](https://www.mkyong.com/java/how-to-write-an-image-to-file-imageio/)

[**https://www.java-forums.org/new-java/7995-how-plot-graph-java-given-samples.html**](https://www.java-forums.org/new-java/7995-how-plot-graph-java-given-samples.html)

[**https://stackoverflow.com/questions/14859593/java-basic-plotting-drawing-a-point-dot-pixel**](https://stackoverflow.com/questions/14859593/java-basic-plotting-drawing-a-point-dot-pixel)

[**https://stackoverflow.com/questions/15045228/draw-a-vector-graph-in-java-7**](https://stackoverflow.com/questions/15045228/draw-a-vector-graph-in-java-7)

[**https://stackoverflow.com/questions/8202253/saving-a-java-2d-graphics-image-as-png-file**](https://stackoverflow.com/questions/8202253/saving-a-java-2d-graphics-image-as-png-file)

[**http://www.java2s.com/Code/Java/2D-GraphicsGUI/SavingaGeneratedGraphictoaPNGorJPEGFile.htm**](http://www.java2s.com/Code/Java/2D-GraphicsGUI/SavingaGeneratedGraphictoaPNGorJPEGFile.htm)

[**https://stackoverflow.com/questions/19621105/save-image-from-jpanel-after-draw**](https://stackoverflow.com/questions/19621105/save-image-from-jpanel-after-draw)

[**https://www.rapidtables.com/web/color/gray-color.html**](https://www.rapidtables.com/web/color/gray-color.html)

[**https://stackoverflow.com/questions/13106906/how-to-create-a-histogram-in-java**](https://stackoverflow.com/questions/13106906/how-to-create-a-histogram-in-java)

[**https://www.codeproject.com/Tips/1172662/Histogram-Equalisation-in-Java**](https://www.codeproject.com/Tips/1172662/Histogram-Equalisation-in-Java)

[**https://ro.wikipedia.org/wiki/BMP\_(format\_fi%C8%99ier)**](https://ro.wikipedia.org/wiki/BMP_(format_fi%C8%99ier))

[**https://en.wikipedia.org/wiki/BMP\_file\_format**](https://en.wikipedia.org/wiki/BMP_file_format)

[**http://upm.ro/intranet/ecalin/cd\_educational/cd/test\_multim/lab\_scmm/formatul\_bmp.htm**](http://upm.ro/intranet/ecalin/cd_educational/cd/test_multim/lab_scmm/formatul_bmp.htm)

[**https://www.tutorialspoint.com/java/java\_files\_io.htm**](https://www.tutorialspoint.com/java/java_files_io.htm)

[**https://www.javatpoint.com/java-io**](https://www.javatpoint.com/java-io)

[**https://docs.oracle.com/javase/tutorial/essential/io/file.html**](https://docs.oracle.com/javase/tutorial/essential/io/file.html)

[**http://www.vogella.com/tutorials/JavaIO/article.html**](http://www.vogella.com/tutorials/JavaIO/article.html)

**Pentru interfata vizuala(cea ceruta specific la aceasta tema):**

<http://tutorials.jenkov.com/javafx/index.html>

<https://www.javatpoint.com/javafx-tutorial>

<https://www.tutorialspoint.com/javafx/index.htm>

<https://o7planning.org/en/11147/javafx-treeview-tutorial>

<http://www.learningaboutelectronics.com/Articles/How-to-create-multiple-scenes-and-switch-between-scenes-in-JavaFX.php>

<https://stackoverflow.com/questions/15041760/javafx-open-new-window>

<https://stackoverflow.com/questions/1234912/how-to-programmatically-close-a-jframe>

<https://www.geeksforgeeks.org/javafx-label/>

<https://stackoverflow.com/questions/25468882/change-color-of-background-in-javafx-canvas>