2 DIRECTION ARRAY AND APPLICATION

11/2/2021

Thạc sỹ: Trần Lê Như Quỳnh

DEFINITION

COLUMN

ROW

	0	1	2	3	…	n
0	A[0][0]	A[0][1]	A[0][2]	A[0][3]		A[0][n]
1	A[1][0]					
2	A[2][0]					
3	A[3][0]					
4	A[4][0]					
…						
n	A[n][0]					

DECLARE 2D ARRAY IN JAVA CODE

- Create 2D array:
  - dataType *NameArray[row][column];*
  - dataType NameArray [ ][ ];
  - dataType NameArray [ ][column];
  - dataType NameArray [row][];
- Example:

```
int Scores[100][10]; /* 100 x 10 set of scores */
char Maze[5][5];     /* 5 x 5 matrix of chars for Maze */
float FloatM[3][4];  /* 3 x 4 matrix of floats */
```

WORKING WITH ELEMENTS IN 2D ARRAY IN JAVA

- Get the number of rows :

  array.length => 3
- Get data in 2D array

Ex: get data at position row = 0, col=1

  array[0][1] => get value 122
- Set data in 2D array

Ex: set data at position row = 0, col=1 equal 10

  array[0][1] =10
- Get a row in 2D array

Ex: get row 1

  array[1] => [100,123,145]

	0	*1*	*2*	*3*
0	**120**	**122**		
1	**100**	**123**	**145**	
2	**124**	**134**	**155**	**156**
3	**138**	**125**	**145**	**155**

	0	*1*	*2*	*3*
0	**120**	**10**		
1	**100**	**123**	**145**	
2	**124**	**134**	**155**	**156**
3	**138**	**125**	**145**	**155**

WORKING WITH ELEMENTS IN 2D ARRAY IN JAVA

- Using 2 for loop to working with each elements: one for row, one for column

```
 for(int i=0; i< array.length; i++){// row
     for(int j=0; j<array[i].length; j++){ // column
of each row
        }
           }
```

ThS.Trần Lê Như Quỳnh

Application Of 2D Array

EXERCISES

11/2/2021

Thạc sỹ: Trần Lê Như Quỳnh

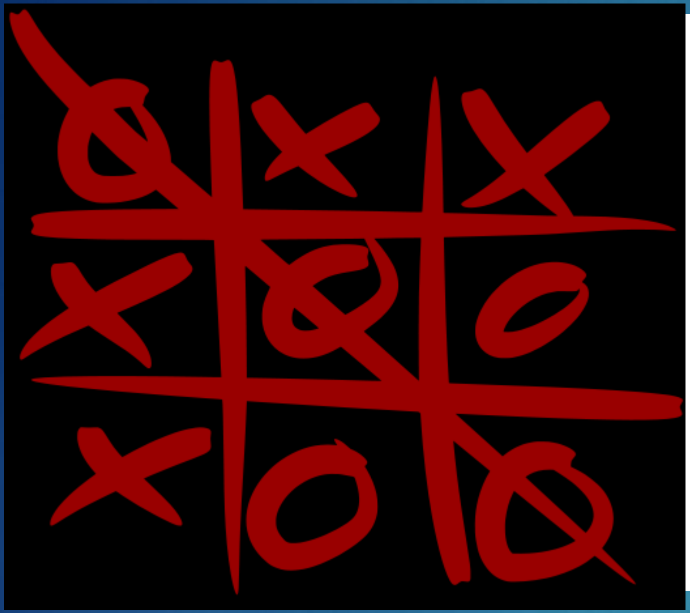
REQUIREMENTS

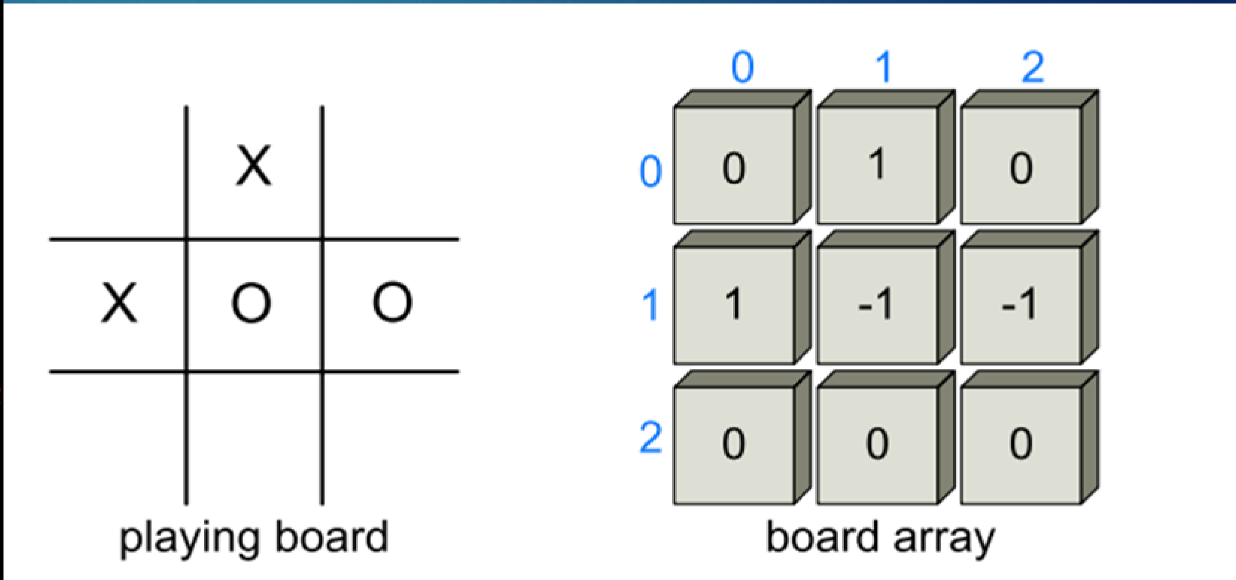
- WORKING AT CLASS
    1. CREATE UTILS CLASS FOR 2D ARRAY
    2. MATRIX
- WORKING AT LAB , AT HOME
    1. TIC-TAC-TOE
    2. MINESWEEPER
    3. BLUR FILTER WITH IMAGEJ
    4. CARO GAME (FULL FUNCTIONS)**
    5. RESEARCH IMAGEJ**

CREATE UTILS CLASS FOR 2D ARRAY

```
public class  Array2DUtils {
  // Get number of elements in 2D array
  public static int getSize(int[ ][ ]  array){
        //TO DO
  }
// Get Column
 public static int[ ] getColumn (int[ ][ ] array){
        //TO DO
}
  // Search
  public static  boolean search(int[ ][ ]  array, int target) {
  // TODO
  }
  // Print data of 2D array
  public static String toString(int[ ][ ]){
  //TODO }
  }
```

TIC-TAC-TOE





	X	
X	O	O

playing board

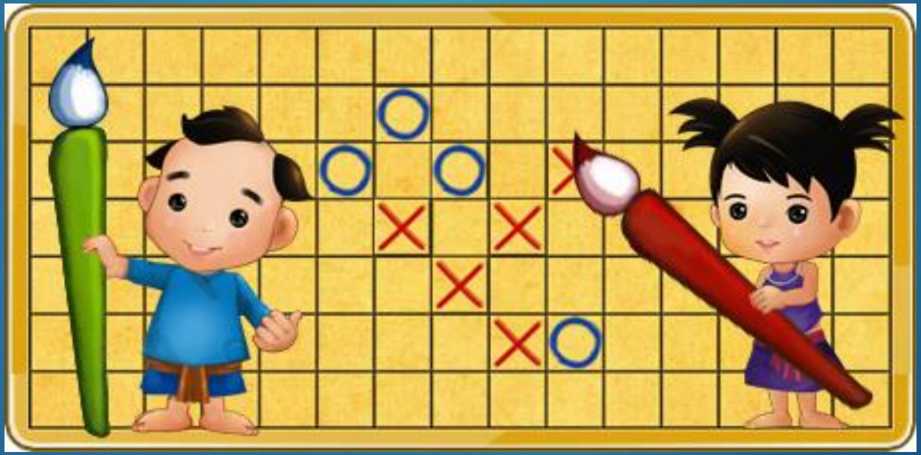
	0	1	2
0	0	1	0
1	1	-1	-1
2	0	0	0

board array

Example code on pages: 119, 120

Advance

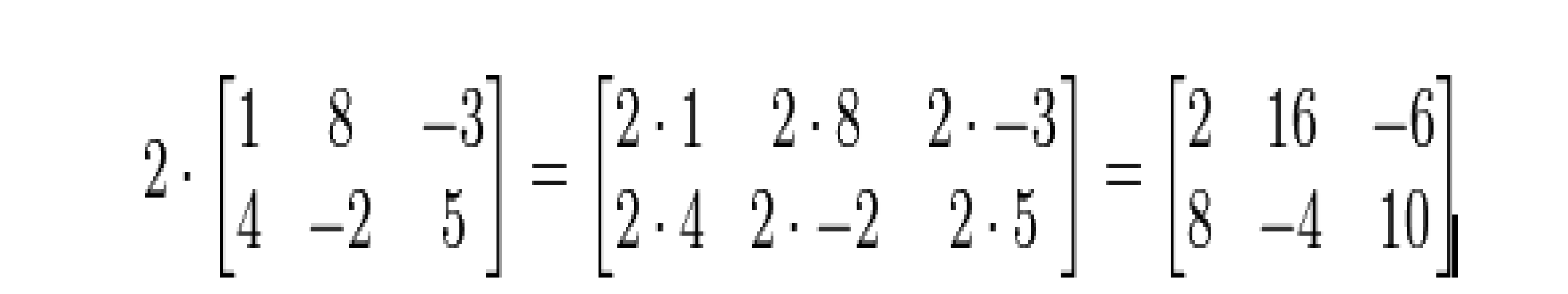
- Build CARO game with full functions:
- 2 player play together
- 1 player play with computer with 3 level: easy, normal, hard
- save game and play saved game



SUM - MATRIX WITH MATRIX

$$\begin{bmatrix} 1 & 3 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 2+5 \\ 1+7 & 0+5 & 0+0 \\ 1+2 & 2+1 & 2+1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 7 \\ 8 & 5 & 0 \\ 3 & 3 & 3 \end{bmatrix}$$

MULTIPLICATION- MATRIX WITH CONSTANT



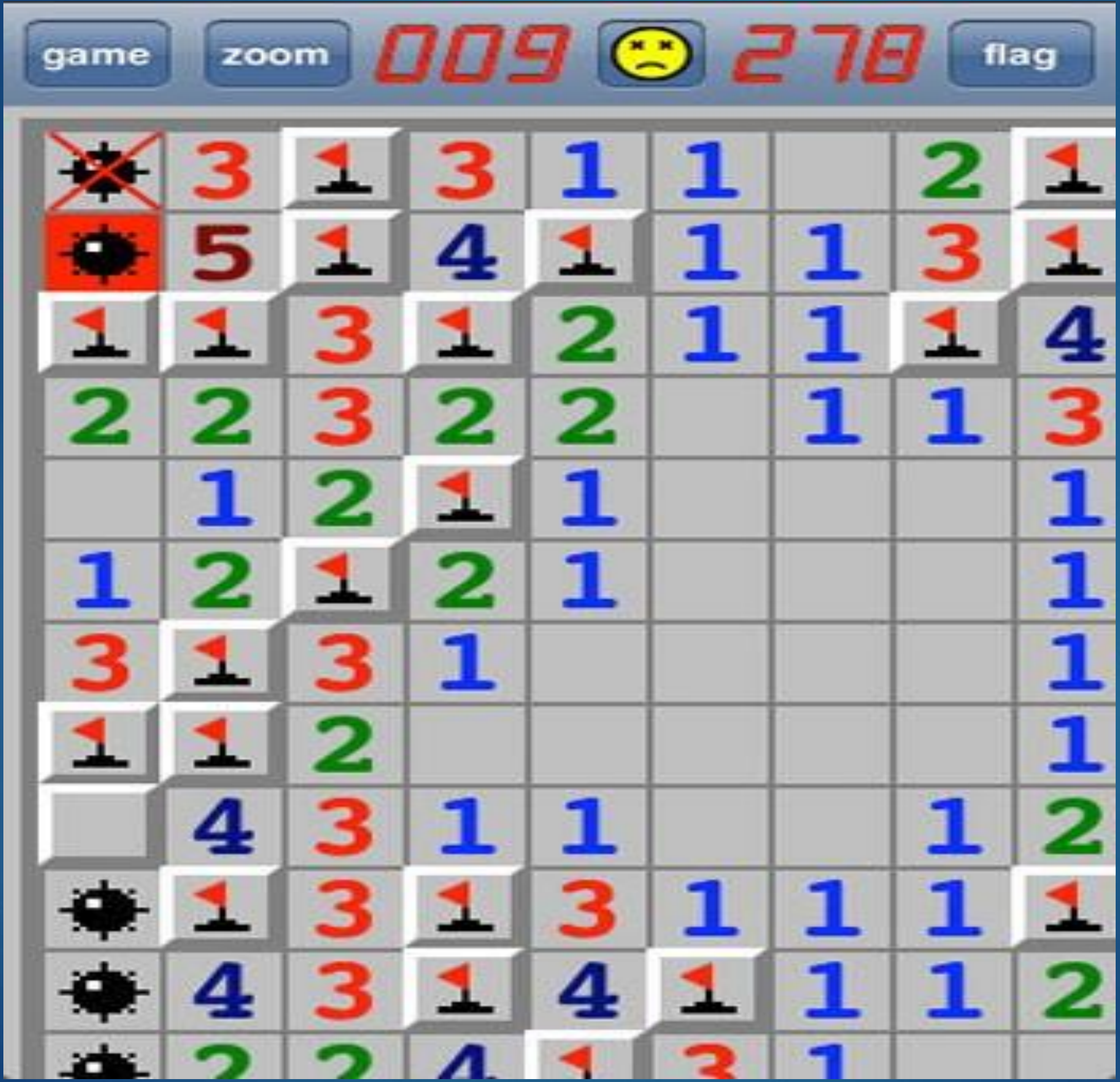
$$2\cdot\begin{bmatrix}1 & 8 & -3\\ 4 & -2 & 5\end{bmatrix}=\begin{bmatrix}2\cdot 1 & 2\cdot 8 & 2\cdot -3\\ 2\cdot 4 & 2\cdot -2 & 2\cdot 5\end{bmatrix}=\begin{bmatrix}2 & 16 & -6\\ 8 & -4 & 10\end{bmatrix}$$

MULTIPLICATION- MATRIX WITH MATRIX

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix} \times \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} (1 \times 3 + 0 \times 2 + 2 \times 1) & (1 \times 1 + 0 \times 1 + 2 \times 0) \\ (-1 \times 3 + 3 \times 2 + 1 \times 1) & (-1 \times 1 + 3 \times 1 + 1 \times 0) \end{bmatrix}$$

$$= \begin{bmatrix} 5 & 1 \\ 4 & 2 \end{bmatrix}$$

MINESWEEPER GAME



Hints

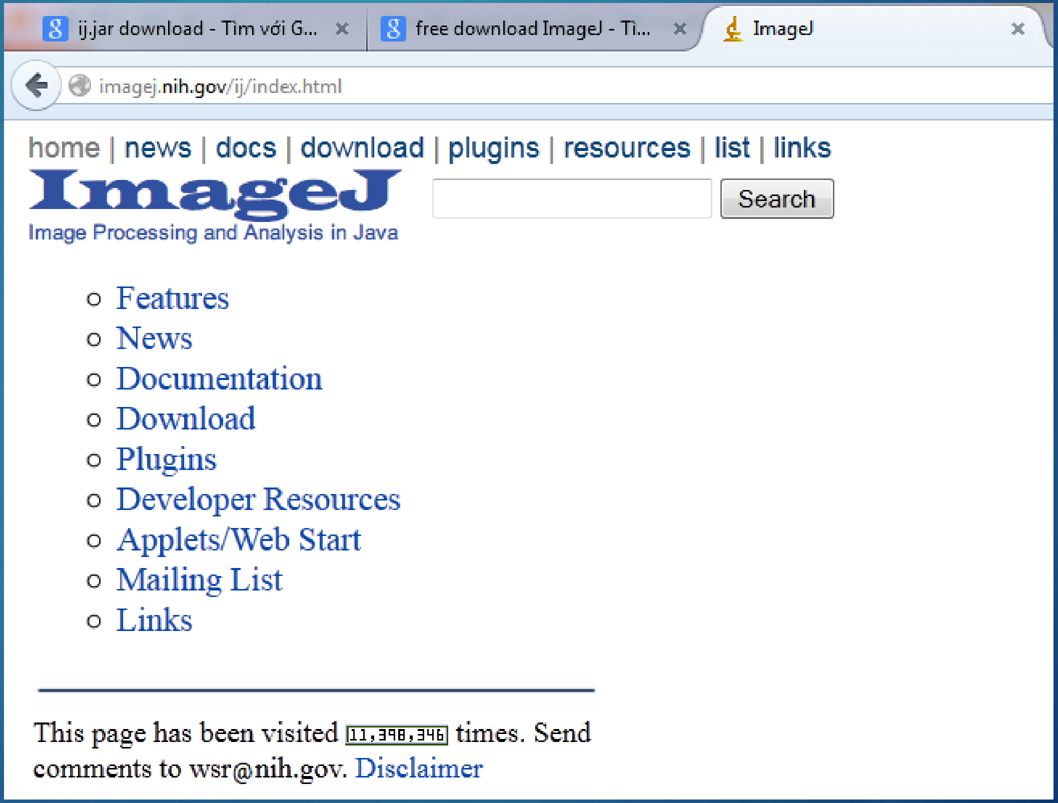
	-1							
					-1			
		-1	-1				-1	
	-1							
-1					-1			
				-1				
							-1	
			-1	-1	-1		-1	

Hints

1	-1	1	0	0	1	0	0	0
1	2	2	2	2	-1	2	1	1
0	1	-1	-1	2	1	2	-1	1
1	2	3	2	1	0	1	1	1
2	-1	1	0	1	1	1	0	0
-1	2	1	1	2	-1	1	0	0
1	1	0	1	-1	2	2	1	1
0	0	1	3	4	3	3	-1	2
0	0	1	-1	-1	-1	3	-1	2

ImageJ

- Vào trang download gói thư viện hình ảnh ij.jar về http://imagej.nih.gov/ij/docs/index.html

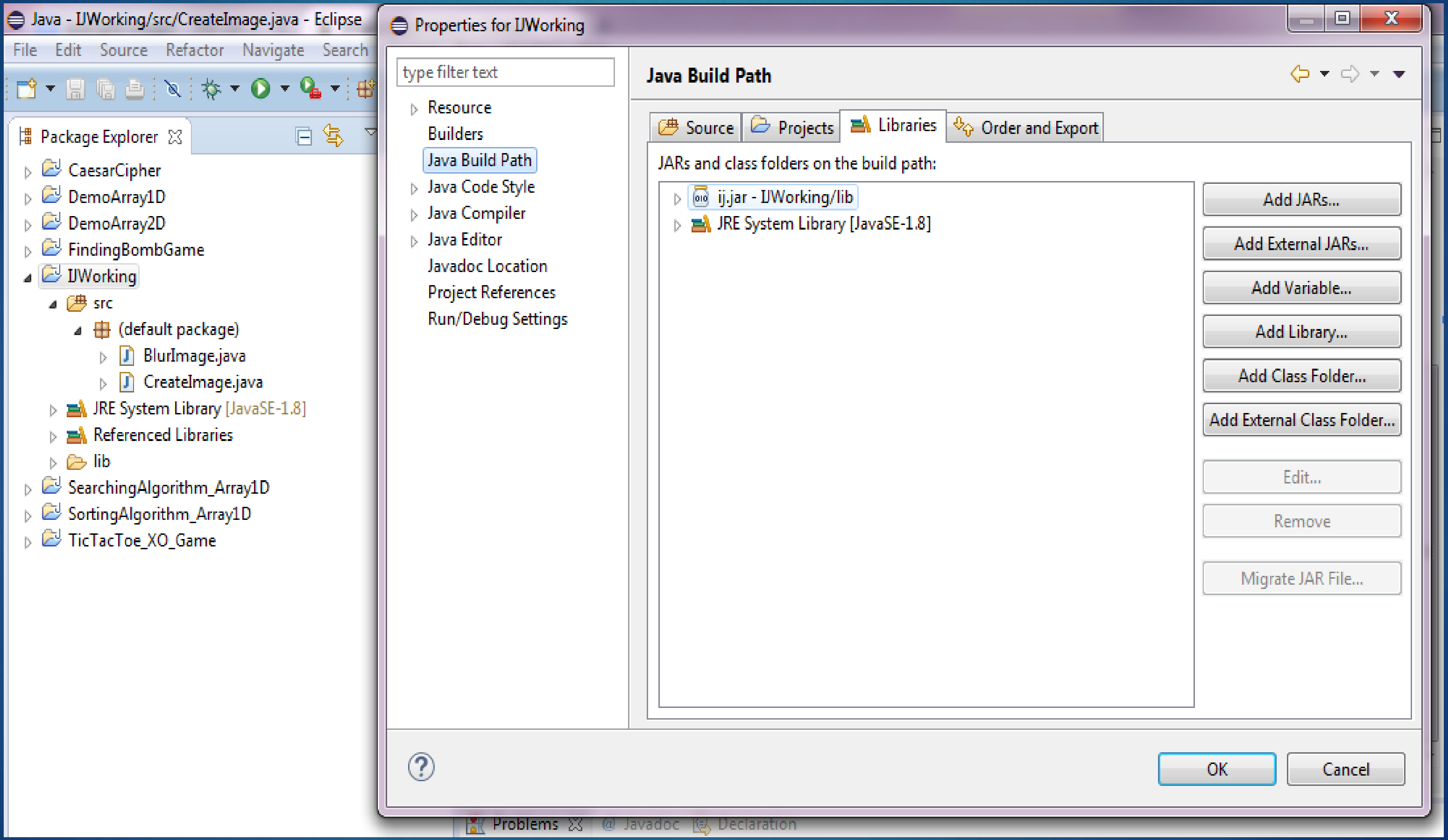

ij.jar download - Tìm với G...
free download ImageJ - Ti...
ImageJ
imagej.nih.gov/ij/index.html
home | news | docs | download | plugins | resources | list | links
ImageJ
Image Processing and Analysis in Java
Search

- Features
- News
- Documentation
- Download
- Plugins
- Developer Resources
- Applets/Web Start
- Mailing List
- Links

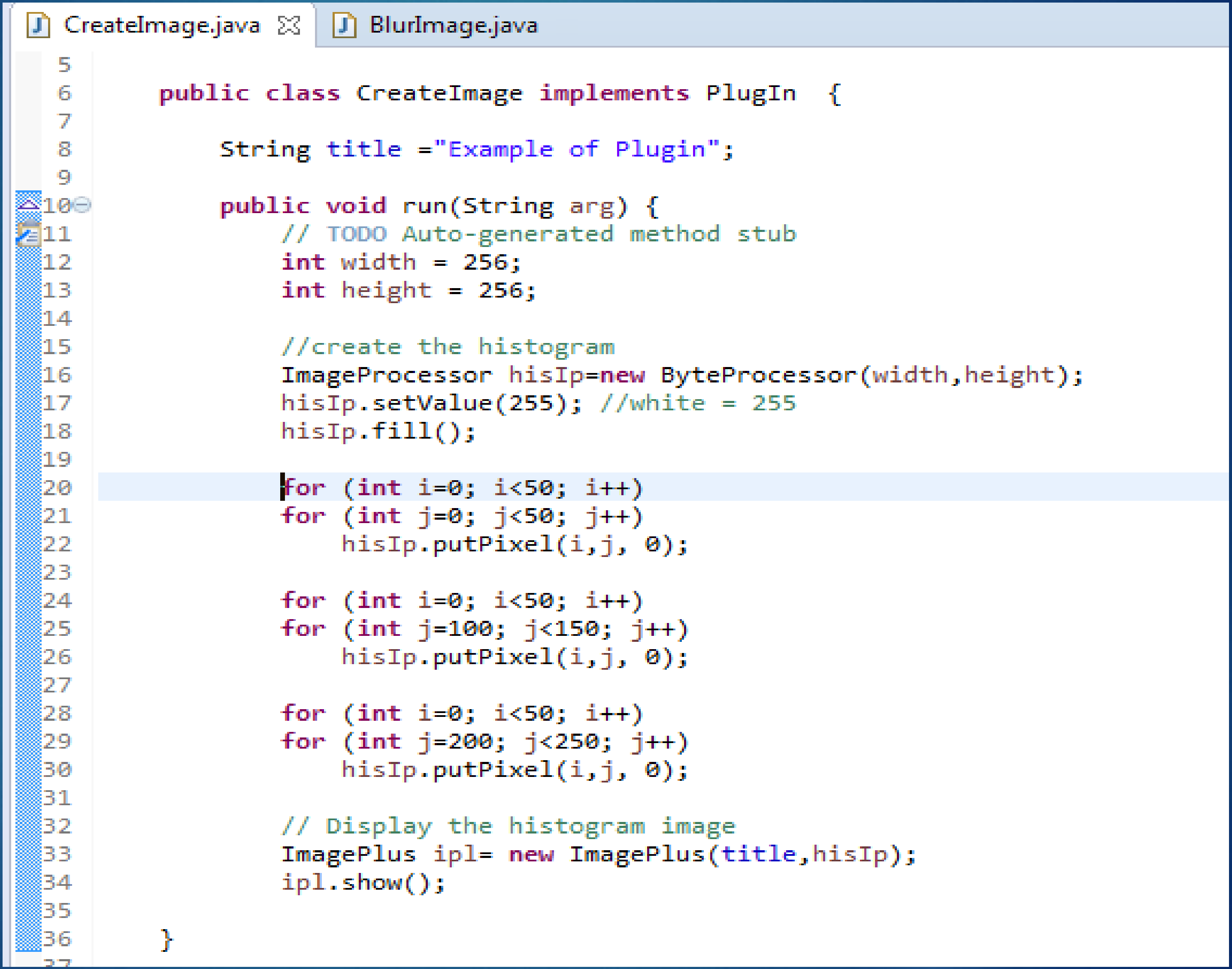
This page has been visited 11,398,346 times. Send comments to wsr@nih.gov. Disclaimer

Example 1: create image

- Tạo một java project, sau đó vào Java build path add gói ij.jar vào.



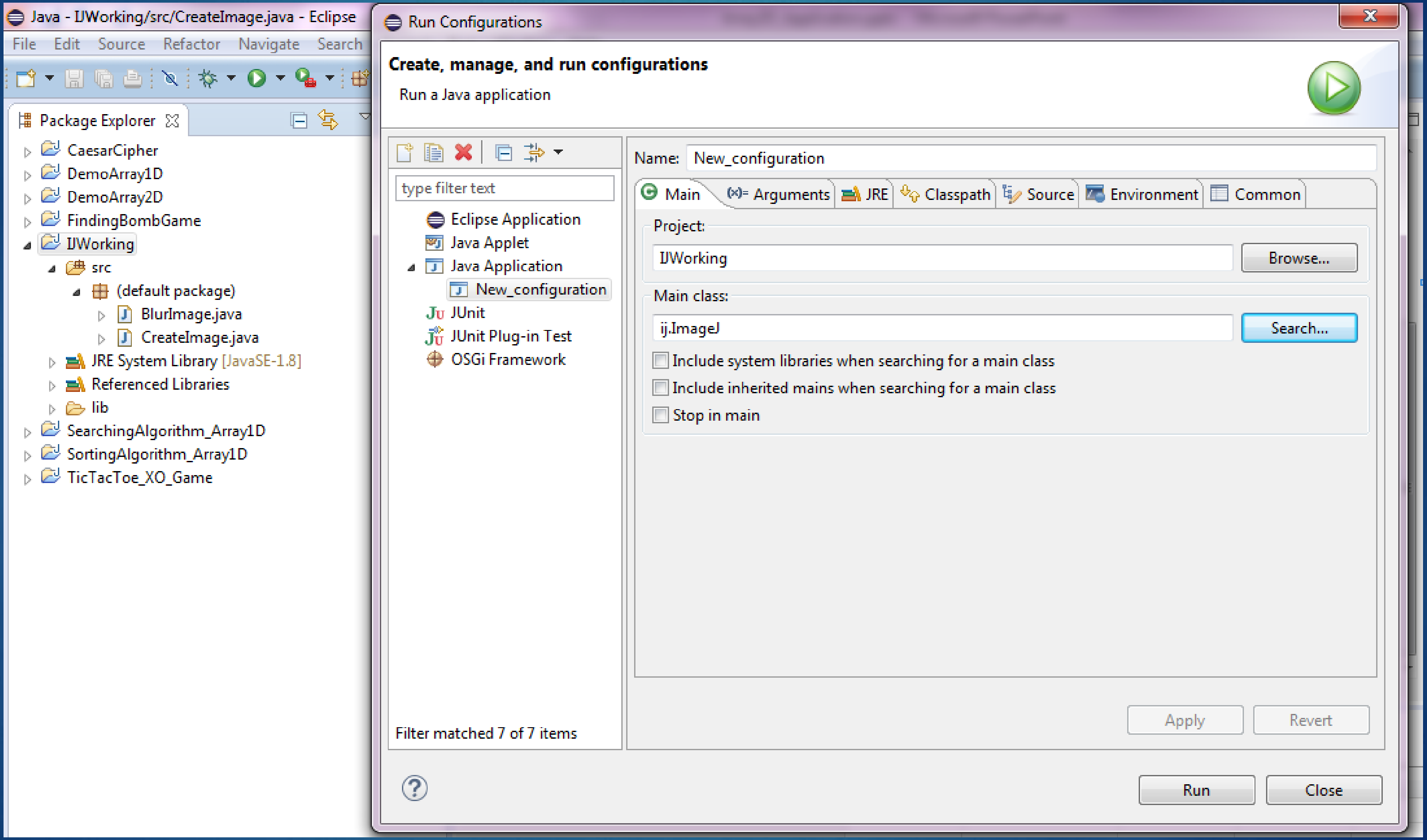
Example 1: create image



```

    public class CreateImage implements PlugIn  {

        String title ="Example of Plugin";

        public void run(String arg) {
            // TODO Auto-generated method stub
            int width = 256;
            int height = 256;

            //create the histogram
            ImageProcessor hisIp=new ByteProcessor(width,height);
            hisIp.setValue(255); //white = 255
            hisIp.fill();

            for (int i=0; i<50; i++)
            for (int j=0; j<50; j++)
                hisIp.putPixel(i,j, 0);

            for (int i=0; i<50; i++)
            for (int j=100; j<150; j++)
                hisIp.putPixel(i,j, 0);

            for (int i=0; i<50; i++)
            for (int j=200; j<250; j++)
                hisIp.putPixel(i,j, 0);

            // Display the histogram image
            ImagePlus ipl= new ImagePlus(title,hisIp);
            ipl.show();

    }
```

Example 1: create image

- Cấu hình để chạy



Run Configurations

Create, manage, and run configurations

Run a Java application

Name: New_configuration

Project: IJWorking

Main class: ij.ImageJ

□ Include system libraries when searching for a main class

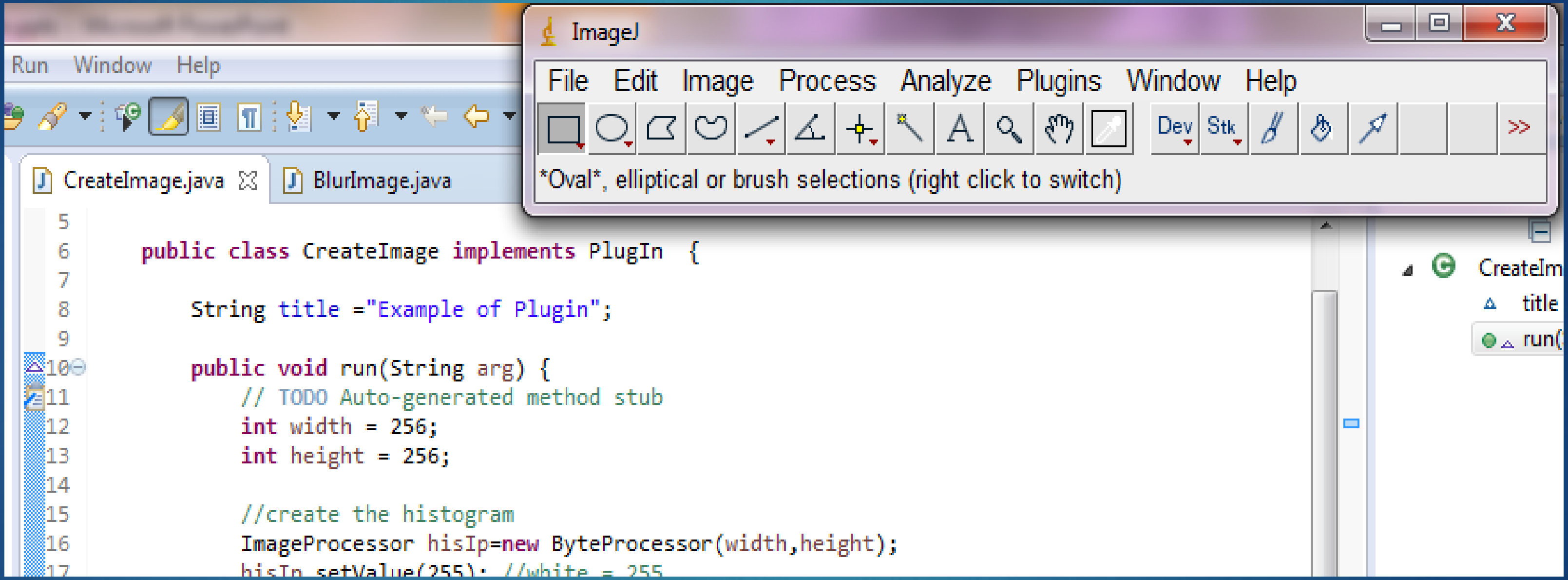
□ Include inherited mains when searching for a main class

□ Stop in main

Filter matched 7 of 7 items

Example 1: create image

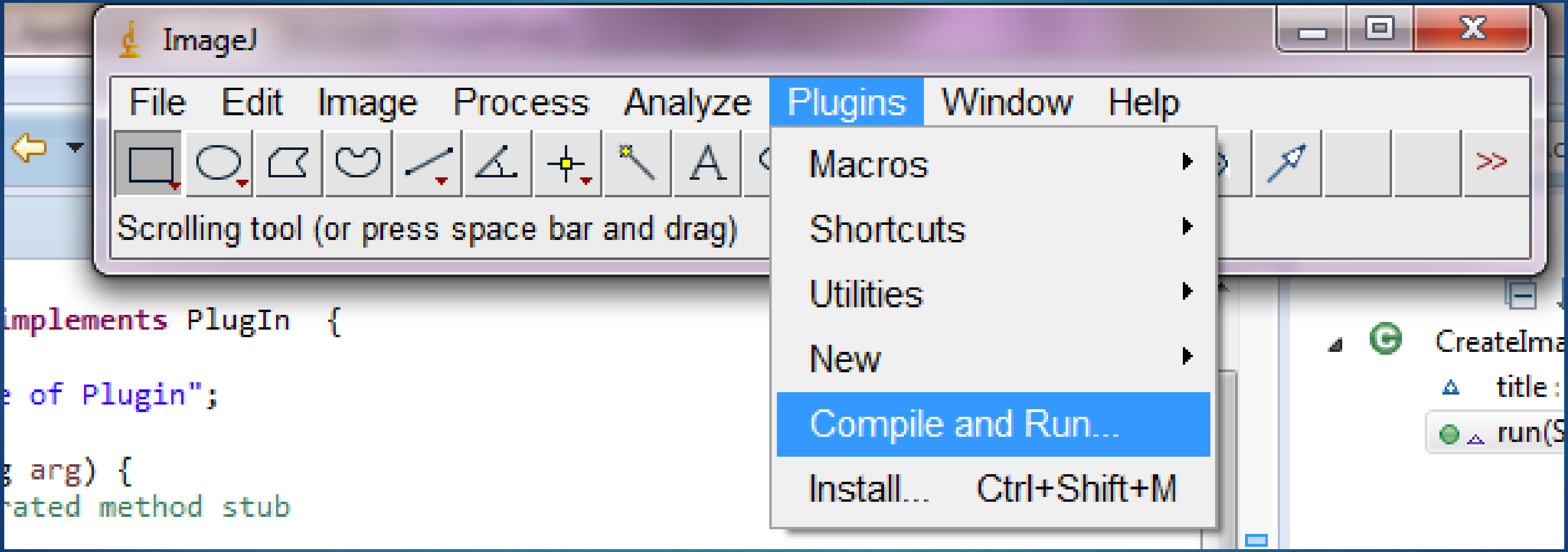
- Run



Oval, elliptical or brush selections (right click to switch)

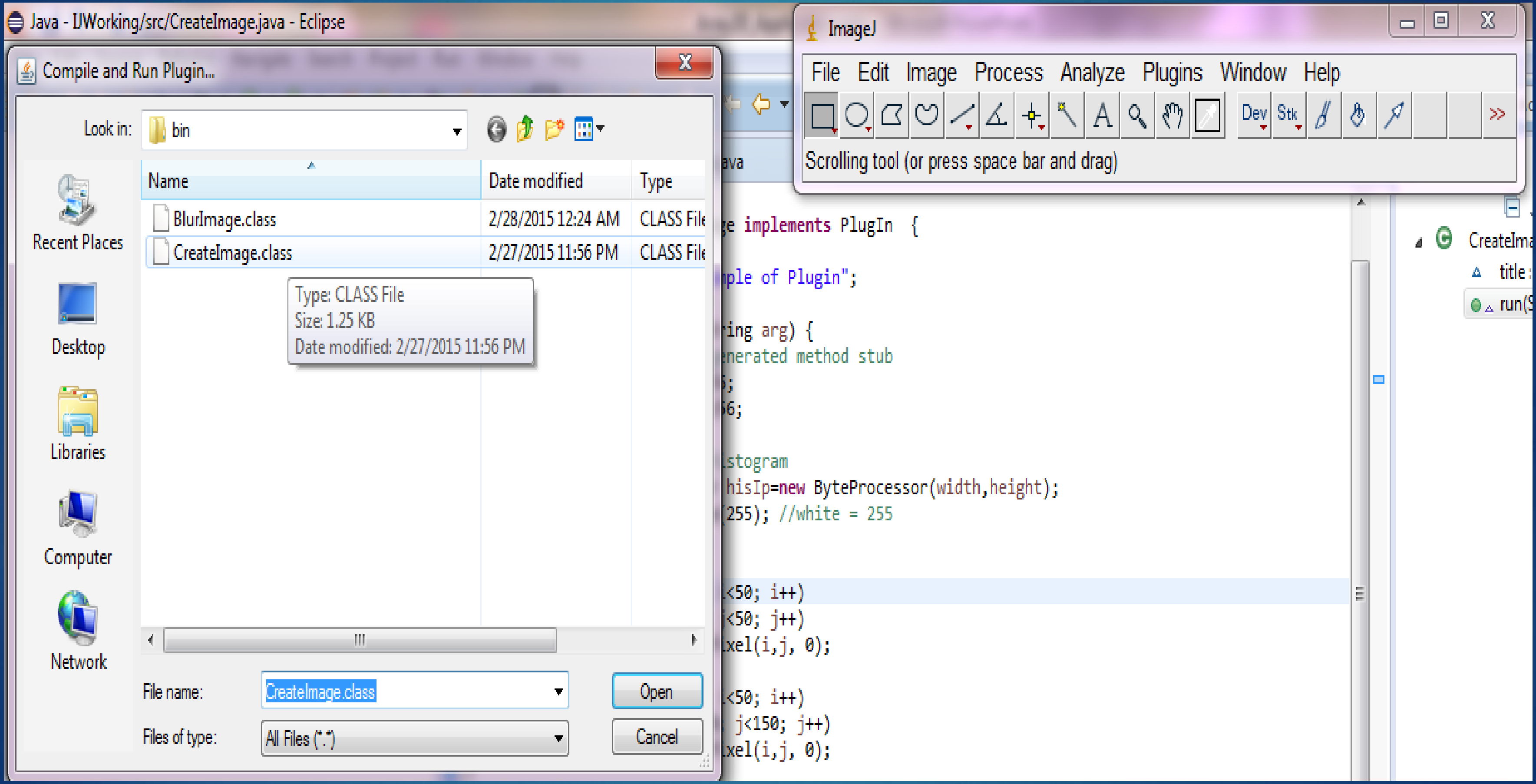
```
public class CreateImage implements PlugIn  {

    String title ="Example of Plugin";

    public void run(String arg) {
        // TODO Auto-generated method stub
        int width = 256;
        int height = 256;

        //create the histogram
        ImageProcessor hisIp=new ByteProcessor(width,height);
        hisIp.setValue(255); //white = 255
```

Ví dụ 1: tạo một ảnh bằng IJ



```
implements PlugIn  {

 of Plugin";

 arg) {
rated method stub
```

Example 1: create image



Java - IJWorking/src/CreateImage.java - Eclipse

Compile and Run Plugin...

Look in: bin

Name	Date modified	Type
BlurImage.class	2/28/2015 12:24 AM	CLASS File
CreateImage.class	2/27/2015 11:56 PM	CLASS File

Type: CLASS File
Size: 1.25 KB
Date modified: 2/27/2015 11:56 PM

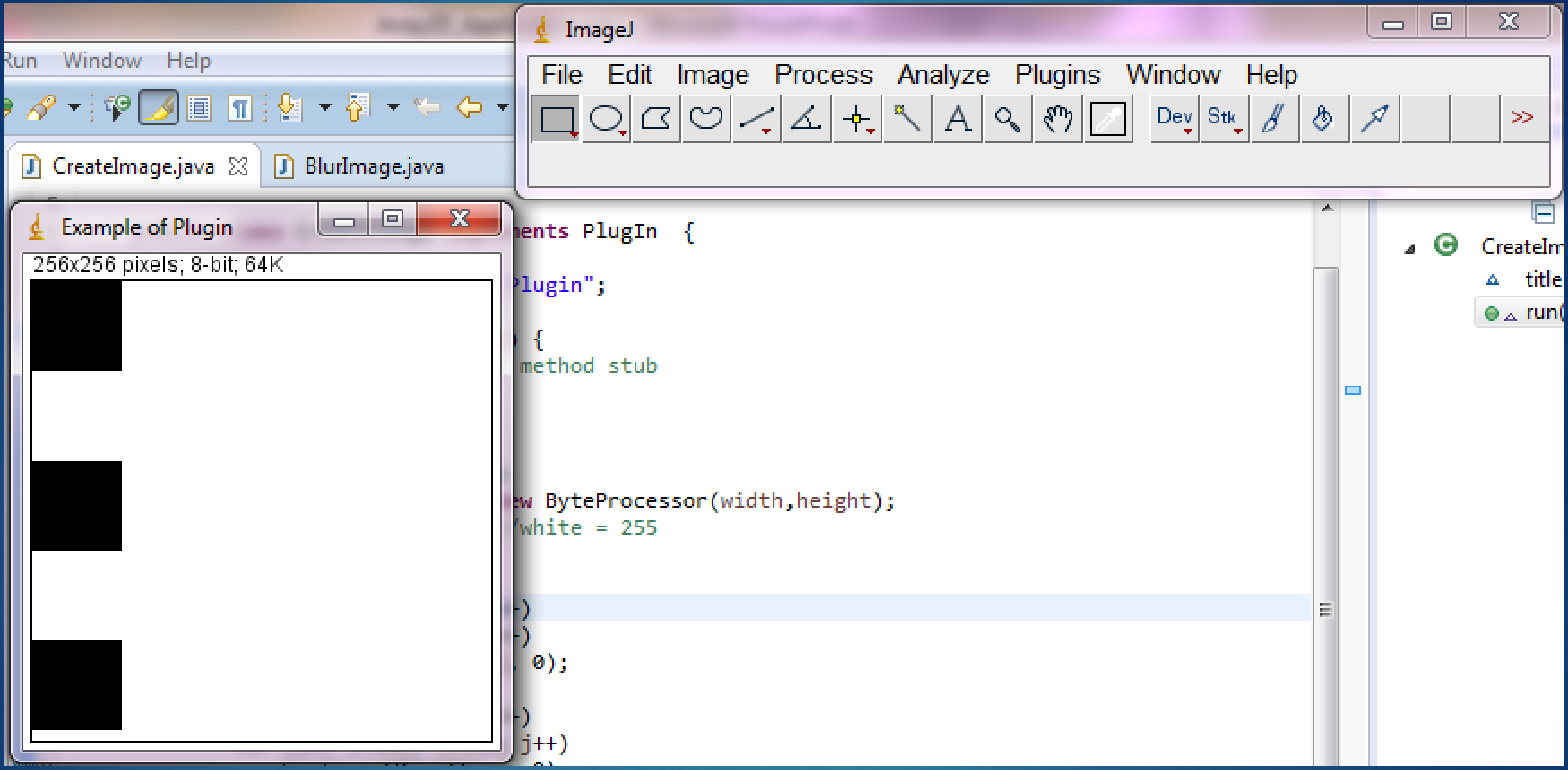
File name: CreateImage.class

Files of type: All Files (*.*)

Scrolling tool (or press space bar and drag)

```
e implements PlugIn  {

ple of Plugin";

ing arg) {
enerated method stub
;
6;

stogram
hisIp=new ByteProcessor(width,height);
255); //white = 255


<50; i++)
<50; j++)
xel(i,j, 0);

<50; i++)
 j<150; j++)
xel(i,j, 0);
```

Example 1: create image



256x256 pixels; 8-bit; 64K

```
ents PlugIn  {

lugin";

 {
method stub




w ByteProcessor(width,height);
/white = 255



)
)
 0);

)
 j++)
```

Mean Filter 3x3

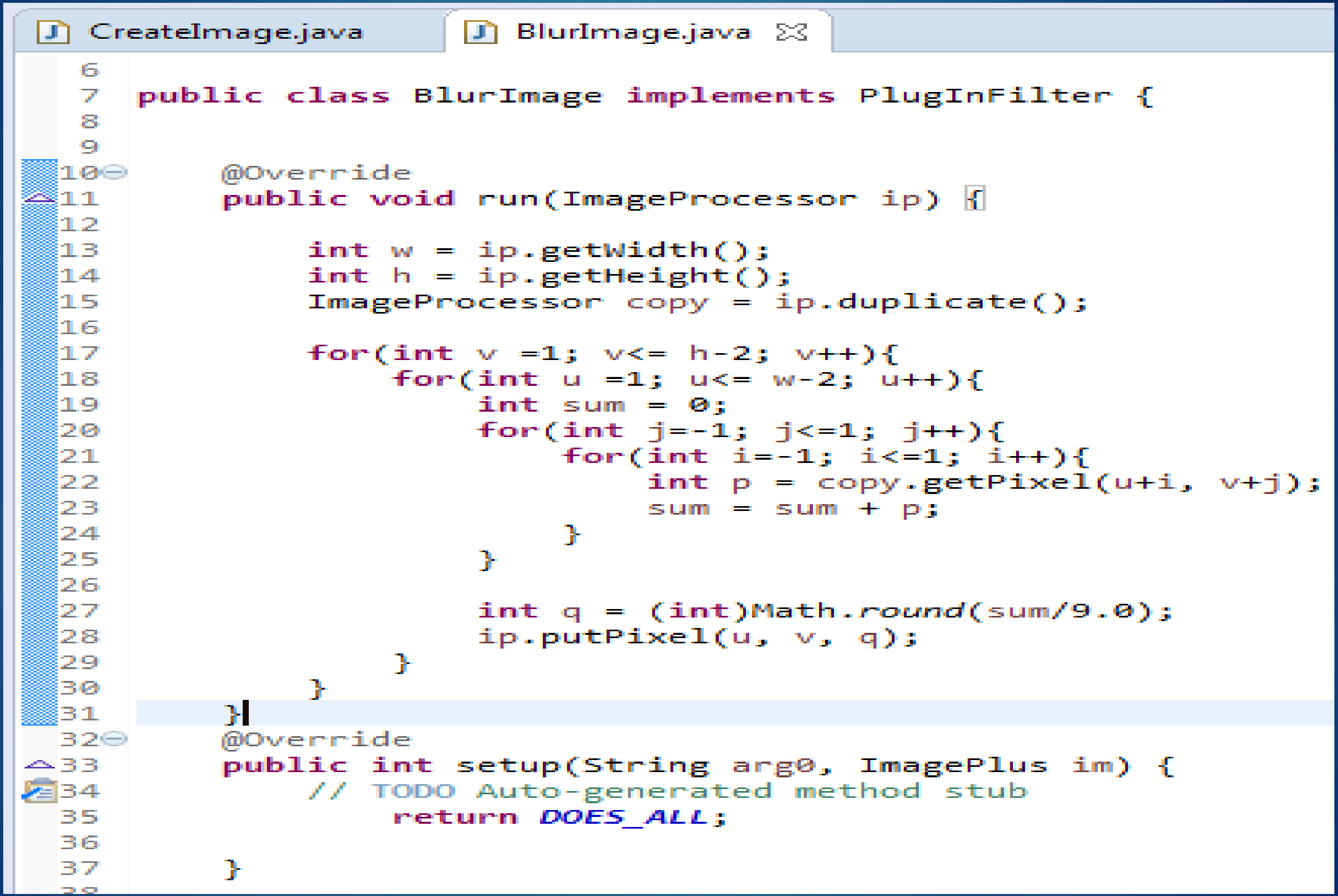
124	126	127	125	125	127
120	150	125	156	200	167
115	119	123	145	148	180
134	154	135	154	176	178
125	154	145	167	178	180
145	145	154	158	200	190

.

Trong cửa sổ đầu tiên, ta lấy giá trị trung bình của 9 ô: (115 + 119 + 120 + 123 + 124 +125 + 126+127+150)/9. Ta lấy giá trị trung bình 125. Gán giá trị trung bình lại cho tất cả các ô trong ma trận của cửa số.

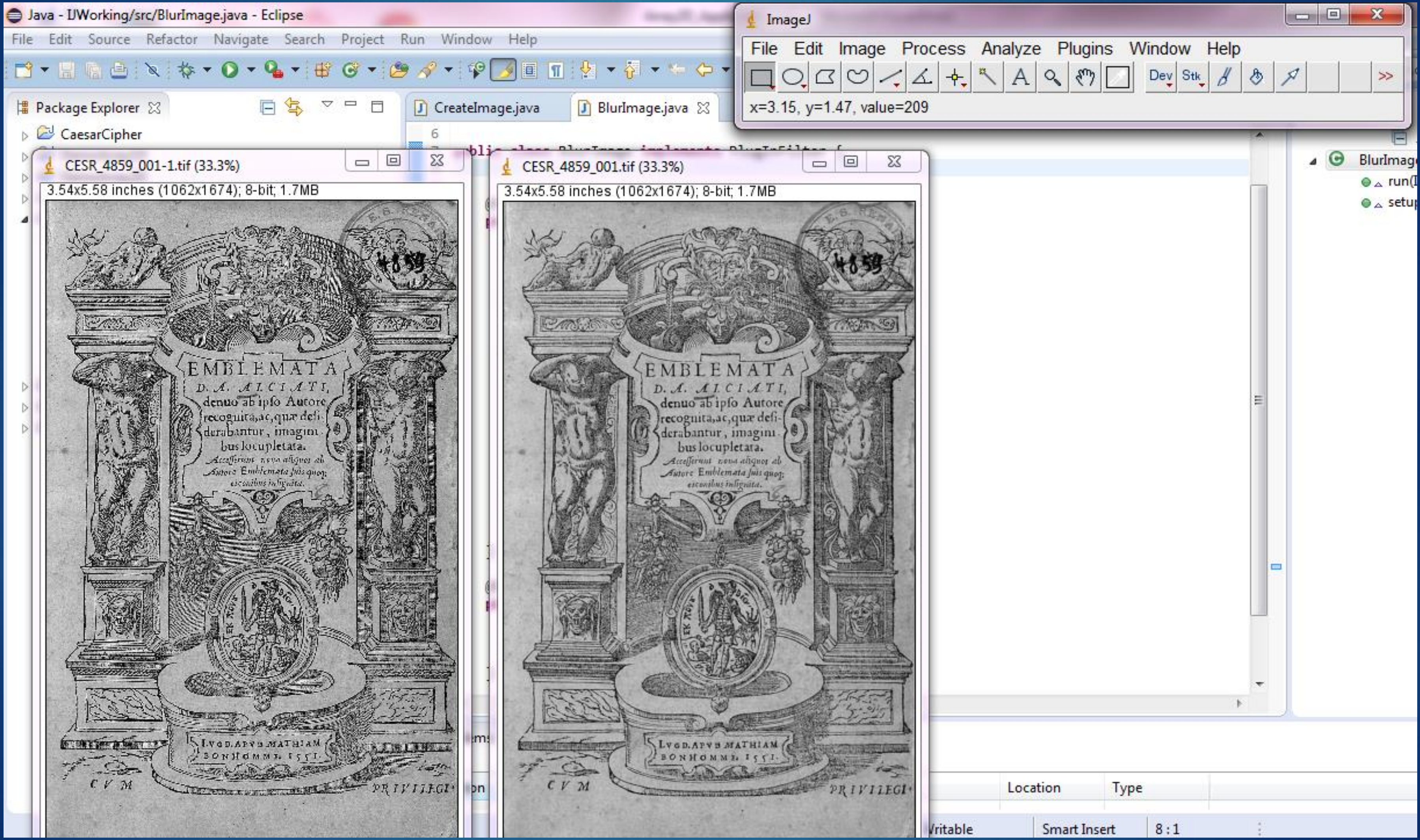
125	125	125	125	125	127
125	125	125	156	200	167
125	125	125	145	148	180
134	154	135	154	176	178
125	154	145	167	178	180
145	145	154	158	200	190

Mean filter 3x3



```java

public class BlurImage implements PlugInFilter {


    @Override
    public void run(ImageProcessor ip) {

        int w = ip.getWidth();
        int h = ip.getHeight();
        ImageProcessor copy = ip.duplicate();

        for(int v =1; v<= h-2; v++){
            for(int u =1; u<= w-2; u++){
                int sum = 0;
                for(int j=-1; j<=1; j++){
                    for(int i=-1; i<=1; i++){
                        int p = copy.getPixel(u+i, v+j);
                        sum = sum + p;
                    }
                }

                int q = (int)Math.round(sum/9.0);
                ip.putPixel(u, v, q);
            }
        }
    }
    @Override
    public int setup(String arg0, ImagePlus im) {
        // TODO Auto-generated method stub
            return DOES_ALL;

    }
```

Mean filter 3x3



Java - IJWorking/src/BlurImage.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
CaesarCipher
CreateImage.java
BlurImage.java
ImageJ
File Edit Image Process Analyze Plugins Window Help
Dev Stk
x=3.15, y=1.47, value=209
CESR_4859_001-1.tif (33.3%)
3.54x5.58 inches (1062x1674); 8-bit; 1.7MB
CESR_4859_001.tif (33.3%)
3.54x5.58 inches (1062x1674); 8-bit; 1.7MB
EMBLEMATA
D. A. ALCIATI,
denuo ab ipſo Autore recognita, ac, quæ deſiderabantur, imaginibus locupletata.
BlurImage
run(
setup
Location
Type
Writable
Smart Insert
8 : 1

Median filter 3x3

124	126	127	125	125	127
120	150	125	156	200	167
115	119	123	145	148	180
134	154	135	154	176	178
125	154	145	167	178	180
145	145	154	158	200	190

Trong cửa sổ đầu tiên ta lấy 9 giá trị sắp xếp theo thứ tự tăng dần: 115, 119, 120, 123, 124, 125, 126, 127, 150. Ta lấy giá trị trung vị được 124. Gán giá trị trung vị lại cho tất cả các ô trong ma trận của cửa số

124	124	124	125	125	127
124	124	124	156	200	167
124	124	124	145	148	180
134	154	135	154	176	178
125	154	145	167	178	180
145	145	154	158	200	190