

CS4222 Software Development Assessment

Part 1 Specification

Introduction

The goal of this assignment is to develop a Java class that allows us to store and manipulate camera images. The images will be stored as a two-dimensional array of integers. Each location in the array will store the value of one pixel in the image. Real images have reasonably high “resolutions” and relatively large dimensions (e.g. 640 x 320). If your camera takes pictures that have ‘megapixels’ they can be quite large (4000 x 3000) and contain millions of pixels. However, for testing purposes we will keep our two-dimensional arrays relatively small, mostly measured with dimensions that are in the tens (e.g. 15x10 or 7x5 and so on).

Typically, picture images are rectangular but they may be square. Our class should be able to deal with both.

The images can be manipulated by “Flipping” them (i.e. switching the top to the bottom or the left to the right) and “Rotating” them (i.e. turning the image through 90° clockwise or anti-clockwise). A short video showing the type of operations required has been provided on Sulis.

Currently, the Image class looks like this

```
public class Image {  
    private int[][] pixels;  
    private int width;  
    private int height;  
  
    public Image(int[][] pixels) {  
        this.pixels = pixels;  
        this.height = pixels.length;  
        this.width = pixels[0].length;  
    }  
}
```

The constructor is passed the image pixels as a two-dimensional array of integers. In addition to the constructor the class should provide the following methods

- A toString method with the following header

```
public String toString()
```

The method returns a String that contains the two-dimensional array data neatly arranged so that it can be displayed on a screen or printed on a page.

- A flip method that has the following header

```
public void flip( boolean horizontal )
```

The method ‘flips’ the image by swapping the rows or the columns depending on the value of the parameter. NOTE: The ‘flipped’ image replaces the original image.

If the boolean parameter ‘horizontal’ is true then the method should swap the first row of the image with the last row; the second row with the second last row; the third row with the third last row; and so on until the entire image has been ‘flipped’ top to bottom.

If the boolean parameter ‘horizontal’ is false then the method should swap the first column of the image with the last column; the second column with the second last column; the third column with the third last column; and so on until the entire image has been ‘flipped’ left to right.

NOTE: All of the rows in the two-dimensional image array are the same size. All of the columns in the two-dimensional image array are the same size. So swapping rows or columns does not alter the dimensions of the two-dimensional image array – it just rearranges the pixel data in it.

Here is what an image might look like before and after flipping

0	1	2	3	4	5	6
---	---	---	---	---	---	---

5	5	5	5	5	5	5
0	1	2	3	4	5	6

1 0	11	1 2	1 3	1 4	1 5	1 6
2 0	2 1	2 2	2 3	2 4	2 5	2 6
3 0	3 1	3 2	3 3	3 4	3 5	3 6
4 0	4 1	4 2	4 3	4 4	4 5	4 6
5 0	5 1	5 2	5 3	5 4	5 5	5 6

Flip Horizontal

4 0	4 1	4 2	4 3	4 4	4 5	4 6
3 0	3 1	3 2	3 3	3 4	3 5	3 6
2 0	2 1	2 2	2 3	2 4	2 5	2 6
1 0	11	1 2	1 3	1 4	1 5	1 6
0	1	2	3	4	5	6

0	1 0	2 0	3 0	4 0	5 0
1	11	2 1	3 1	4 1	5 1
2	1 2	2 2	3 2	4 2	5 2
3	1 3	2 3	3 3	4 3	5 3
4	1 4	2 4	3 4	4 4	5 4
5	1 5	2 5	3 5	4 5	5 5

Flip Vertical

5 0	4 0	3 0	2 0	1 0	0
5 1	4 1	3 1	2 1	11	1
5 2	4 2	3 2	2 2	1 2	2
5 3	4 3	3 3	2 3	1 3	3
5 4	4 4	3 4	2 4	1 4	4
5 5	4 5	3 5	2 5	1 5	5

- The header for the rotate method is as follows

```
public void rotate( boolean clockwise )
```

The method 'rotates' the image by transposing each row of the image into a column.

If the boolean parameter 'clockwise' is true then the method should transpose the first (top) row of the image so that it becomes the last column on the right; the second row transposes to become the second last column on the right; the third row transposes to become the third last column on the right; and so on until the entire image has been 'rotated' clockwise.

If the boolean parameter 'clockwise' is false then the method should transpose the first (top) row of the image so that it becomes the first column on the left; the second row transposes to become the second column on the left; the third row transposes to become the third column on the left; and so on until the entire image has been 'rotated' anti-clockwise.

NOTE: To rotate an image you will have to create a second two-dimensional image array and copy the data from the original to the second array. Once all the data has been copied correctly then the new 'rotated' image should replace the previous one.

Keep in mind that the width and height of an image may be different so when you rotate an image the resulting two-dimensional array will have the same height as the original width and the same width as the original height.

0	1	2	3	4	5	6	7	8	9
1		1	1	1	1	1	1	1	1
0	11	2	3	4	5	6	7	8	9
2	2	2	2	2	2	2	2	2	2
0	1	2	3	4	5	6	7	8	9
3	3	3	3	3	3	3	3	3	3
0	1	2	3	4	5	6	7	8	9

Rotate Clockwise
90°

5	4	3	2	1	
0	0	0	0	0	0
5	4	3	2		
1	1	1	1	11	1
5	4	3	2	1	
2	2	2	2	2	2
5	4	3	2	1	
3	3	3	3	3	3

4 0	4 1	4 2	4 3	4 4	4 5	4 6	4 7	4 8	4 9
5 0	5 1	5 2	5 3	5 4	5 5	5 6	5 7	5 8	5 9

5 4	4 4	3 4	2 4	1 4	4
5 5	4 5	3 5	2 5	1 5	5
5 6	4 6	3 6	2 6	1 6	6
5 7	4 7	3 7	2 7	1 7	7
5 8	4 8	3 8	2 8	1 8	8
5 9	4 9	3 9	2 9	1 9	9

0	1	2	3	4	5	6	7	8	9
1 0	11	1 2	1 3	1 4	1 5	1 6	1 7	1 8	1 9
2 0	2 1	2 2	2 3	2 4	2 5	2 6	2 7	2 8	2 9
3 0	3 1	3 2	3 3	3 4	3 5	3 6	3 7	3 8	3 9
4 0	4 1	4 2	4 3	4 4	4 5	4 6	4 7	4 8	4 9
5 0	5 1	5 2	5 3	5 4	5 5	5 6	5 7	5 8	5 9

Rotate Anti-Clockwise
90°

9	1 9	2 9	3 9	4 9	5 9
8	1 8	2 8	3 8	4 8	5 8
7	1 7	2 7	3 7	4 7	5 7
6	1 6	2 6	3 6	4 6	5 6
5	1 5	2 5	3 5	4 5	5 5
4	1 4	2 4	3 4	4 4	5 4

3	1 3	2 3	3 3	4 3	5 3
2	1 2	2 2	3 2	4 2	5 2
1	11	2 1	3 1	4 1	5 1
0	1 0	2 0	3 0	4 0	5 0

Finally, when a series of flip and rotate operations is applied to an image, the arrangement of the pixel data in the two-dimensional array should represent the cumulative effect of the various manipulations applied to the original image.

Submission Requirements

Your solution to the assignment should be submitted on Sulis on or before 16h00 Friday 4th March 2022.

You should submit TWO files (1) Image.java that implements the operations specified for the class, and (2) ImageDriver.java that tests the various operations to verify that they work correctly. In each file you should include prominent comments that contain your ID number and your Name. Sulis will accept any filenames you use but it is extremely important that you adhere to the file name conventions.

You should NOT submit zip files.

This part of the assignment is worth 25% of the total of 100%.

The marks will be allocated as follows

Component	Allocation
Flip Method	9
Rotate Method	13

Testing	3
---------	---