



NOTE: The above design is for the enhanced image model and the new synchronous controller.

Below is the design for the older model of project 3, which is largely unchanged, and the changes documented below.

- 1) Pattern generation will implicitly call pixelate image operation on default 100 pixelation criteria distributed evenly on image width and height..
- 2) DMC is assumed to be 489 colors as provided in the project description.
- 3) The image pattern generation is assumed do not have any blanks.
- 4) Unicode characters will be used to represent the symbols for cross stitched pattern.
- 5) Only known commands will be executed and invalid commands will be returned as invalid command.
- 6) height and width of the image should not be less than 10 pixels across.
- 8) Pixelation parameter cannot be less than height and width of the image.
- 9) pixelation parameter cannot be negative.
- 10) input seed for mosaic cannot be greater than the product of image height and width.
- 11) mosaic parameter cannot be negative

- 12) Default execute and executePattern method in abstractimageprocessor will return image and imagePattern which is input and can be null to adhere command pattern .
- 13) If invalid input or invalid command is encounter then display the message and continue next command.
- 14) For load command input image can be null thats the desired behavior.
- 15) filename will be validated on execution and not in constructor.
- 16) input parameters will be validated on execution and not in constructor.
- 17) only .png or .jpg via save is supported
- 18) Only text file is supported for pattern generation.

Change log:

- 1) Updating return datatype of methods filterImage, transformImage, reduce in ImageModel from int[][][] to Image type as per recommendation received from previous assignment.
- 2) Updating return datatype of methods filterImage, transformation, reduce in ImageOperator from int[][][] to Image type as per recommendation received from previous assignment.
- 3) kernel and transformation matrix added constant matrix
- 4) kernel and transformation copy retuning from the corresponding classes.
- 5) Changed the ImageModel for mosaic, pixelate and pattern generation.
- 6) Added classes to support mosaic, pixelate and pattern generation.
- 7) Removed image instance from model class and passed as parameter in the methods.
- 8) Added Image load, save and save text in the image model interface.

Controller Test Cases using mock model:

S.No	Test Case	Test Case Description	Expected Result
1	testLoad	Test load was called on calling load()	Appendable and Readable to be checked
2	testSave	Test save was called on calling save()	Appendable and Readable to be checked
3	testFilterImageBlur	Test filterImage method call	Appendable and Readable to be verified
4	testTransformImageGreyScale	Test transformImage method call	Appendable and Readable to be verified

5	testReduce	Test reduce method call	Appendable and Readable to be verified
6	testPixelate	Test pixelate method call	Appendable and Readable to be verified
7	testCreateMosaic	Test createMosaic method call	Appendable and Readable to be verified
8	testGeneratePattern	Test correct function call by controller	Appendable and Readable to be verified
9	testSingleTransaction	Test load, operation, and save	Appendable and Readable to be verified
10	testFilterImageSharpen	Test filterImage method call	Appendable and Readable to be verified
11	testTransformImageSepiaTone	Test transformImage method call	Appendable and Readable to be verified
12	testDither	Test reduce method call	Appendable and Readable to be verified
13	testSingleTransactionforPattern	Test load, image operation pixelate, pattern generation and save pattern	Appendable and Readable to be verified
14	testSingleTransactionforFilterSharpen	Test load, sharpen operation, and save image	Appendable and Readable to be verified
15	testSingleTransactionForTransformSepiaTone	Test load, sepiatone operation, and save image	Appendable and Readable to be verified
16	testSingleTransactionForDither	Test load, dither operation and save image	Appendable and Readable to be verified
17	testSingleTransactionforFilterBlur	Test load, blur operation, and save image	Appendable and Readable to be verified
18	testSingleTransactionForTransformGreyScale	Test load, greyscale operation, and save image	Appendable and Readable to be verified

19	testSingleTransactionForReduce	Test load, color reduce operation and save image	Appendable and Readable to be verified
20	testSingleTransactionForCreateMosaic	Test load, createMosaic operation and save image	Appendable and Readable to be verified
21	testSingleTransactionForPixelate	Test load, pixelate operation and save image	Appendable and Readable to be verified
22	testMultipleOperationsSharpenPattern	Test load, image operation sharpen, pixelate operation, pattern generation and save pattern	Appendable and Readable to be verified
23	testMultipleOperationsBlurPattern	Test load, image operation blur, pixelate operation, pattern generation and save pattern	Appendable and Readable to be verified
24	testMultipleOperationsSharpenSharpenGrey	Test load, image operation sharpen, pixelate operation, pattern generation and save image	Appendable and Readable to be verified
25	testMultipleOperationsBlurBlurSepia	Test load, image operation blur, pixelate operation, pattern generation and save image	Appendable and Readable to be verified
26	testInvalidCommands	Test model with invalid strings.	Return invalid command.

Extended Model Test Cases using junit:

S.No	Test Case	Test Case Description	Expected Result
1	testClosestColor	Test nearest DCM floss color	Should return nearest DCM floss
2	testCrossStitchedPatternLegend	Test the text returned by Cross Stitched Pattern Legend	
3	testPixelatedWidth	Test Pixelated image returned Width	Should not modify width
4	testPixelateHeight	Test Pixelated image returned height	Should not modify height

5	testMosaicWidth	Test Mosaic image returned Width	Should not modify width
6	testMosaicHeight	Test mosaic image returned height	Should not modify height
7	testCrossStitchedWidth	Test the text returned by Cross Stitched Pattern Width	Should not modify width
8	testCrossStitchedHeight	Test the text returned by Cross Stitched Pattern Height	Should not modify height
9	testControllerConstructor	Test the controller constructor	Should not fail
10	testLoad	Test successful loading of image to perform operation	Image should not be null
11	testSave	Test image was saved after operation	Image should be present in destination directory
12	testSavePattern	Test generated pattern was saved after operation	Generated Pattern should be present in the destination directory as txt