UML Class Diagram — text content:

<<interface>> ImageController
+ run(ImageModelInterface)

<<interface>> ImageCommand
+ execute(ImageModel,int[][])

Controller
-Readable in
-Appendable out
-Map<String, Function<Scanner, ImageCommand>> imageCommands
+Controller(Readable, Appendable)

Save
-String fileName
+ Save(Stirng)

BlurImage
-int intensity
+ BlurImage(int)

SharpenImage
-int intensity
+ SharpenImage(int)

GreyScaleImage
+ GreyScaleimage()

SepiaImage
+ SepiaImage()

ReduceImage
-int factor
+ ReduceImage(int)

MosaicImage
-int intensity
+ MosaicImage(int)

PixelImage
-int intensity
+ PixelImage(int)

ToPattern
+ PatternImage()

ProgramRunner
+ main(String[])

<<interface>> ImageModelInterface
+ blur(int, int[][])
+ sharpen(int, int[][])
+ greyScale(int[][])
+ sepia(int[][])
+ reduceColor(int, int[][])
+ toMosaic(int, int[][])
+ pixelate(int, int[][])
+ crossStitch(int[][])
+ loadImage(String): int[][]
+ saveImage(String,int[][])
+ saveTextToFile(String)

«enumeration» Channel
RED
GREEN
BLUE

ImageUtilities
+ readImage(String): int[][]
+ getWidth(String): int
+ getHeight(String): int
+ writeImage(int[][], int, int, String)

<<interface>> Chunking
+ applyChunking(int) : int[][]

ImageModel
+ ImageModel()

<<interface>> Pattern
+ generatePattern()

FileUtilities
+ loadCsvFile(String): String[][]
+ writeToFile(String)

AbstractImageModel
int[][] image;
#AbstractImageModel(int[][])
clamp(int[]) : int[][]
ceckIfZeroOrLess(int)

<<interface>> Fiter
+ applyFilter(int) : int[][]

<<interface>> Transform
+ applyColorTransform() : int[][]

<<interface>> Reduce
+ reduceColor(int) : int[][]

CrossStitch
+ CrossStitch(int[][])

Mosaic
+ Mosaic(int[][])
+ applyChunking(int) : int[][]

AbstractFilter
convolve(int[][], float[][])
- performConvolutionCalculation(int[][],float[][],int,int,int,int,int,int)

AbstractColorTransform
- linearTransformation(int[][], float[][])
colorTransformation(int[][], float[][])

AbstractReduceColorDensity
abstract algorithm(int[][], int)

Pixelation
+ Pixelation(int[][])
+ applyChunking(int) : int[][]

Blur
+ Blur(int[][])
+ applyFilter(int) : int[][]

Sharpen
+ Sharpen(int[][])
+ applyFilter(int) : int[][]

GreyScale
+ GreyScale(int[][])
+ applyColorTransform() : int[][]

Sepia
+ Sharpen(int[][])
+ applyColorTransform() : int[][]

FloydDithering
-performErrorDiffusion(int[][], int, int, float,float,float)
-addError(int[][], int, int, float,float,float,float)
+ FloydDithering(int[][])
+ reduceColor(int) : int[][]

Testing Suite

Testing Sharpen Command		Expected Value
Test Sharpen input Passed	log.toString()	"goat.png 1 [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]"
Test Sharpen Output Passed	out.toString()	"load command was carried out successfully sharpen command was carried out successfully"
Test Sharpen input when image not loaded first	log.toString()	" 1 []"
Test Sharpen output when image not loaded first	out.toString()	"sharpen command was not carried out successfully make sure image is loaded before calling image operation "

Testing Blur Command		Expected Value
Test Blur input Passed	log.toString()	"goat.png 1 [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]"
Test Blur Output Passed	out.toString()	"load command was carried out successfully blur command was carried out successfully"
Test Blur input when image not loaded first	log.toString()	"1 []"
Test Blur output when image not loaded first	out.toString()	"blur command was not carried out successfully make sure image is loaded before

		calling image operation

" |
| | | |
| | | |

Testing GreyScale Command		Expected Value
Test GreyScale input Passed	log.toString()	"goat.png [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]"
Test GreyScale Output Passed	out.toString()	"load command was carried out successfully
greyScale command was carried out successfully"
Test GreyScale input when image not loaded first	log.toString()	" []"
Test GreyScale output when image not loaded first	out.toString()	"greyScale command was not carried out successfully make sure image is loaded before calling image operation

" |
| | | |
| | | |

Testing Sepia Command		Expected Value
Test Sepia input Passed	log.toString()	"goat.png [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]"
Test Sepia Output Passed	out.toString()	"load command was carried out successfully
sepia command was carried out successfully"

Test Sepia input when image not loaded first	log.toString()	"[]"
Test Sepia output when image not loaded first	out.toString()	"Sepia command was not carried out successfully make sure image is loaded before calling image operation "

Testing Reduce Command		Expected Value
Test Reduce input Passed	log.toString()	"goat.png 8 [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]"
Test Reduce Output Passed	out.toString()	"load command was carried out successfully reduce command was carried out successfully"
Test Reduce input when image not loaded first	log.toString()	"8 []"
Test Reduce output when image not loaded first	out.toString()	"reduce command was not carried out successfully make sure image is loaded before calling image operation "

Testing toMosaic Command		Expected Value
Test toMosaic input Passed	log.toString()	"goat.png 1100 [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2,

		3]]]”
Test toMosaic Output Passed	out.toString()	“load command was carried out successfully toMosaic command was carried out successfully”
Test toMosaic input when image not loaded first	log.toString()	“Input: 1100 []”
Test toMosaic output when image not loaded first	out.toString()	“toMosaic command was not carried out successfully make sure image is loaded before calling image operation ”

Testing toMosaic Command		Expected Value
Test toMosaic input Passed	log.toString()	“goat.png 1100 [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]”
Test toMosaic Output Passed	out.toString()	“load command was carried out successfully toMosaic command was carried out successfully”
Test toMosaic input when image not loaded first	log.toString()	“1100 []”
Test toMosaic output when image not loaded first	out.toString()	“toMosaic command was not carried out successfully make sure image is loaded before calling image operation ”

Testing pixelate Command		Expected Value
Test pixelate input Passed	log.toString()	"goat.png 100 [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]"
Test pixelate Output Passed	out.toString()	"load command was carried out successfully pixelate command was carried out successfully"
Test pixelate input when image not loaded first	log.toString()	"100 []"
Test pixelate output when image not loaded first	out.toString()	"pixelate command was not carried out successfully make sure image is loaded before calling image operation "

Testing pattern Command		Expected Value
Test pattern input Passed	log.toString()	"goat.png [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]"
Test pattern Output Passed	out.toString()	"load command was carried out successfully pattern command was carried out successfully"
Test pattern input when image not loaded first	log.toString()	" []"
Test pattern output when image not loaded first	out.toString()	"pattern command was not carried out successfully make sure image is loaded before calling image operation

		"

Testing load Command		Expected Value
Test load input Passed	log.toString()	"goat.png"
Test load Output Passed	out.toString()	"load command was carried out successfully "
Test load input when io exception thrown	log.toString()	"fail.jpg"
Test load output when io exception thrown	out.toString()	"load command was not carried out successfully "

Testing save Command		Expected Value
Test save input Passed	log.toString()	""goat.png goat-blur.jpg [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]"
Test save Output Passed	out.toString()	"load command was carried out successfully\n" + "save command was carried out successfully\n "
Test save input when io exception thrown	log.toString()	"goat.png fail.jpg [[[1, -2, 3], [2, 3, 4]], [[-4, -5, 6, 9], [1], [2, 3]]]"

Test save output when io exception thrown	out.toString()	"load command was carried out successfully\n" + "save command was not carried out successfully\n "
Test save input when image is empty	log.toString()	goat.png []
Test save output when image is empty	out.toString	save command was not carried out successfully make sure image is loaded before calling image operation
Test save input when its text file	log.toString()	goat.png pattern.txt
Test save output when its text file	out.toString()	load command was carried out successfully\n" + "save command was carried out successfully\n
Test save input when its text file and io exception thrown	log.toString()	goat.png fail.txt
Test save output when its text file and io exception thrown	out.toString()	load command was carried out successfully\n" + "save command was not carried out successfully\n

Testing Unsupported Command		Expected Value
Test the input for command thats not supported	log.toString()	""
Test the output for command thats not supported	out.toString()	""Sorry we do not support the kill command currently\n" + "Sorry we do not support the goat.png command currently\n "

Testing Wrong extra input to supported command		Expected Value
Test the input for wrong extra input thats not supported	log.toString()	""
Test the output for wrong extra input thats not supported	out.toString()	""blur command was not carried out successfully check ReadMe for how to use commands " "

Testing Null Model		Expected Exception
Test passing null model	log.toString()	IllegalArgumentException