Project5 Controller Testing Plan

For the testing the controller, the following classes are need in order to mock the behavior of The controller:

MockController

MockImageModel

MockView

In the MockController has a private string builder and each time an operation is called, A corresponding message will be added to the string builder. For example if blur is called from the view, then “Blur” will be added to the string builder. By comparing the controller’s string builder object we can know if the controller is working correctly or not.

However this abstract test can only test on the sequence of all the operations, the real testing would still require running the program multiple times and seeing it ourselves.

1. Test to load a test.jpg file to the view.
   Expect:   load test.jpg.
   public void testLoadIllegalFilePath()

2. Test to load a null filepath to the view.
   Expect:  IOException.
   public void testSaveImage()

3. Test to save the image from the view as output.jpg.
   Expect:    save output.jpg
   public void testSaveIllegalFilePath()

4. Test to save a null filepath from the view.
   Expect:  IOException.

5. Test to save to filepath from the view before load image.
   Expect: IllegalStateException
   public void testSaveBeforeLoad()

6. Test load input then blur it.
   Expect: load input blur
   public void testBlur()

7. Test blur without load any image.
   Expect: IllegalStateException.
   public void testBlurWithoutLoadImage()

8. Test load input then sharpening it.
   Expect: load input sharpening
   public void testSharpening()

9. Test sharpening without load any image.
   Expect: IllegalStateException.
   public void testSharpeningWithoutLoadImage()

10. Test load input then grey scale it.
    Expect: load input greyscale
    public void testGreyScale()

11. Test grey scale without load any image.
    Expect: IllegalStateException.
    public void testGreyScaleWithoutLoadImage()

12. Test load input then sepia tone it.
    Expect: load input blur
    public void testSepiaTone()

13. Test sepia tone without load any image.
    Expect: IllegalStateException.
    public void testSepiaToneWithoutLoadImage()

14. Test load input then pixelation it with 100 chunks.
    Expect: load input pixelation 100
    public void testPixelation()

15. Test pixelation without load any image.
    Expect: IllegalStateException.
    public void testPixelationWithoutLoadImage()

16. Test reduce pixelation with negative chunk numbers.
    Expect: IllegalArgumentException.
    public void testPixelationWithoutIllegalArgument()

17. Test load input then reduce color density it with 5 color per channel.
    Expect: load input pixelation 100
    public void testReduceColorDensity()

18. Test reduce color density without load any image.
    Expect: IllegalStateException.
    public void testReduceColorDensityWithoutLoadImage()

19. Test reduce color density with negative color per channel.
    Expect: IllegalArgumentException.

public void testReduceColorDensityWithoutIllegalArgument()

20. Test load input then mosaic it with 500 seedns.
    Expect: load input mosaic 500.
    public void testMosaic()

21. Test mosaic without load any image.
    Expect: IllegalStateException.
    public void testMosaicWithoutLoadImage()

22. Test mosaic with negative color per channel.
    Expect: IllegalArgumentException.
    public void testMosiacWithoutIllegalArgument()

23. Test execute the script that is created in the interactive mode.
    Expect: run script.
    public void testExecuteScript()

24. Test execute the script that has no content.
    Expect: IOException.
    public void testExecuteScriptWithEmptyContent()

25. Test to create cross stitch pattern in the UI.Load image firs.
    Expect: load input grid view generated .
    public void testCreateCrossStitchPatternInUi()

26. Test to create cross stitch pattern in the UI without load image.
    Expect: IllegalStateException.
    public void testCreateCrossStitchPatternInUiWithoutLoad()

27. Test to create cross stitch pattern in the UI with negative chunk bumber.
    Expect: IllegalArgumentException.
    public void testCreateCrossStitchPatternInUiWithNegativeChunk()

28. Test to replace the current DmcLegend with another Dmc color.
    Expect: color replaced updating current dmc lengend.
    public void testChangeColorInCrossStitch()

29. Test to replace the current DmcLegend with another Dmc color. change color 3 times.
    Expect: color replaced updating current dmc lengend
    color replaced updating current dmc lengend
    color replaced updating current dmc lengend.

public void testChangeColorInCrossStitchThreeTimes()

30. Test to display the image after cross stitch.
    Expect: displaying image after cross stitch for UI .
    public void testDisplayImageAfterCrossStitch()

31. Test to drawy the pattern in txt after cross stitch.
    Expect:  generate txt file in UI mode .
    public void testGenerateTxtAfterCrossStitch()