**Part 1: Reading from/to a file using byte streams**

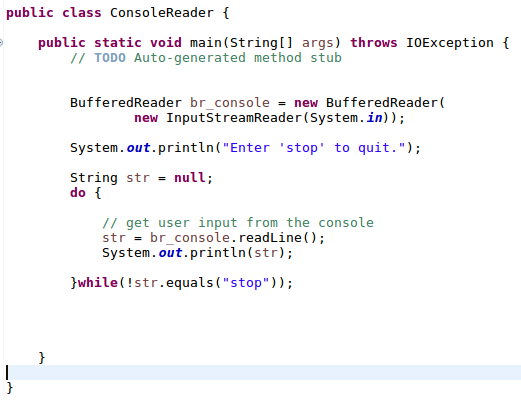
Create a class named “ByteStreamHandler” and include a main() method. Download the “my\_text\_file.text” provided in the “supplementary\_materials” folder under “Week 3 – working with files and images” in Moodle, and copy it to a location in your local hard drive.

Create a program to read the file, one byte at a time and write back the content to a file named “my\_new\_file.text” which will be saved at the same as the “my\_text\_file.text” file. You may use the following class implementation. Comment each section of the code explaining the functionality of each code segment.



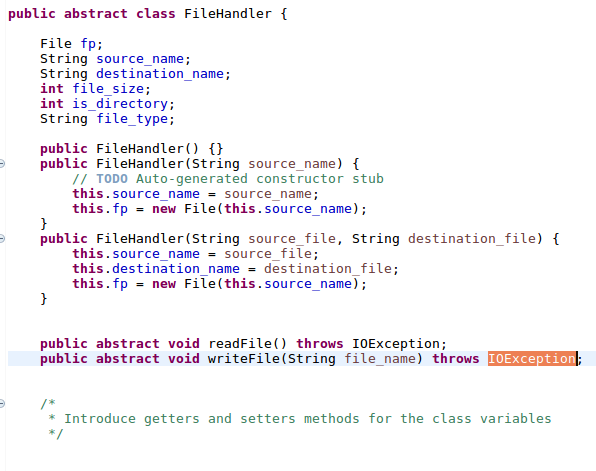
**Part 2: Reading data from the console**

Create a class named “ConsoleReader” and include a main() method. Include the functionality to read user input from the console until the user enters “stop”. You may use the following implementation of the “ConsoleReader” class.

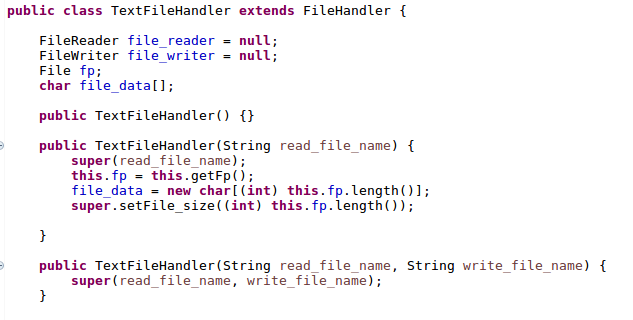


**Part 3: Reading/writing from/to files**

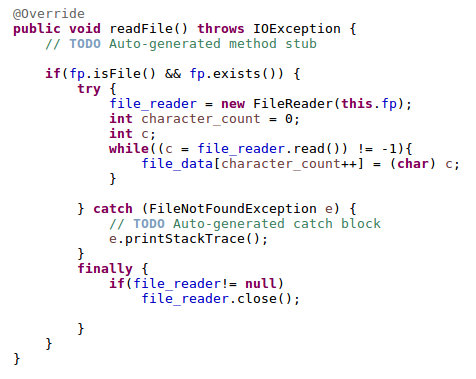
Create an abstract class named “FileHandler” and include the following attributes with appropriate methods to set and get class attribute values.



Create a “TextFileHandler” class that extends from the “FileHandler” class.



Override the readFile() abstract method in the "FileHandler" super class, to read data from the text file available in “supplementary\_materials” folder to store the character data in an array structure.



Override the writeFile() abstract method in the “FileHandler” superclass to write the contents of the file\_data array to a file specified by the file\_name



Create a new class named “FileApplicationTester” and include a main() method.

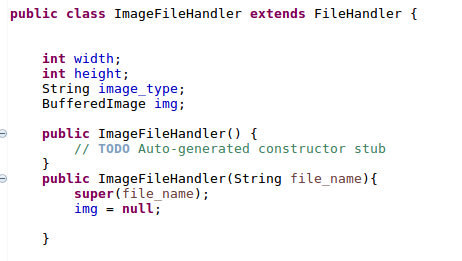
1. Implement the functionality to read the text file provided in the “supplementary\_materials” folder and store the content in the file\_data array in “TextFileHandler” class.
2. Implement the functionality to write the data in the file\_data array to a new file named “new\_file.text” in the “supplementary\_materials” folder.

Modify the readFile() and writeFile() method to use BufferedReader and BufferedWriter instances and relevant methods to read/write data from/to the files, respectively. You may refer to the lecture notes on working with files and images on Moodle.

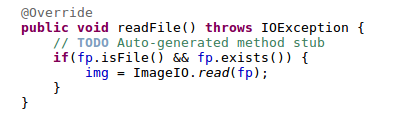
Include comments to explain the operation of each code segment.

**Part 4: Reading image files**

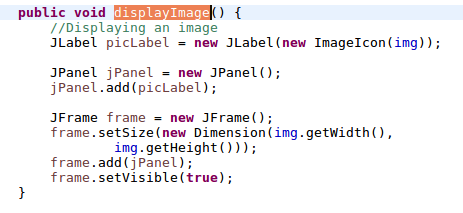
Create a new class “ImageFileHandler” which extends from the “FileHandler” class. Implement the appropriate constructors to accept image file name as an argument.



Override the readFile() method to read the image to a BufferedImage instance.



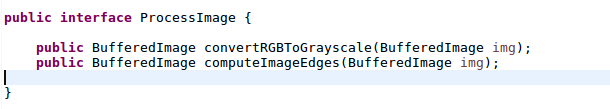
Implement a new displayImage() function to view the image read from the file



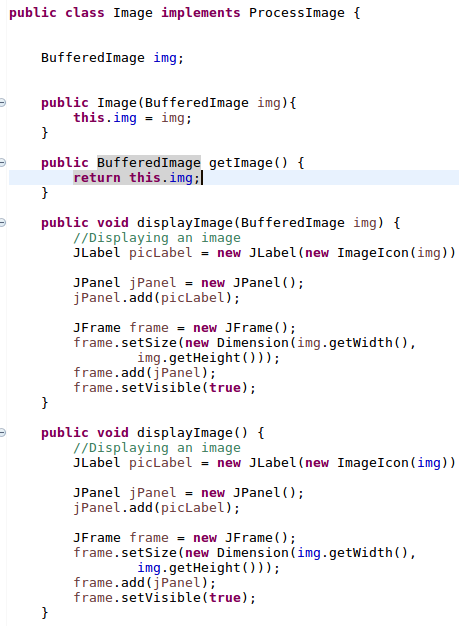
Modify the main() in the “FileApplicationTester” class create an instance of “ImageFilHandler” class to read the image file provided in the “supplementary\_materials” folder.

Insert a function call to the displayImage() method to view the image file in the main() of “FileApplicationTester” class.

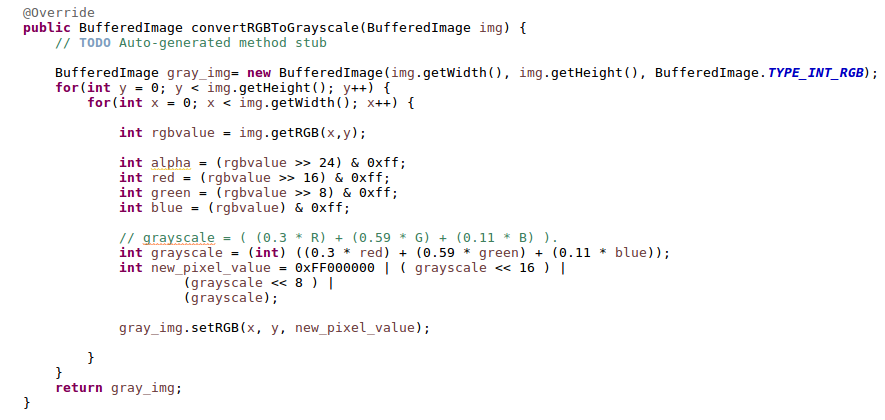
Create an interfaced named “ProcessImage” and include the following implementation.



Create a new class named “Image” and implement the interface “ProcessImage”



Override the “convertRGBToGrayscale()” method to convert an RGB image into grayscale



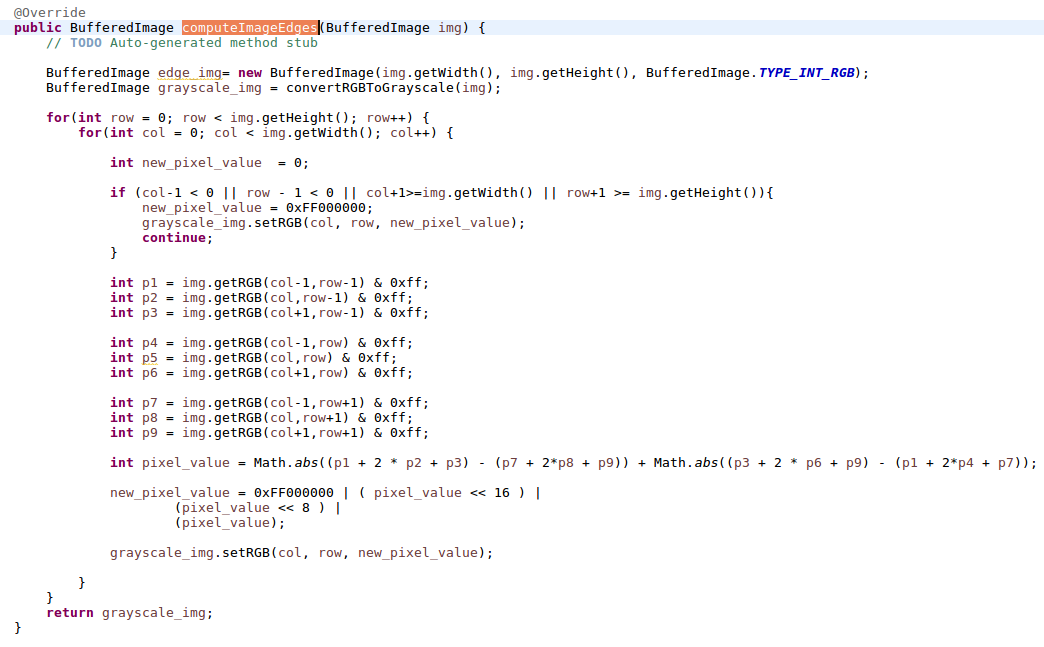
Modify the main() in “FileApplicationTester” class to read the image file provided in the “supplementary\_materials” folder and view both the RGB and grayscale versions of the image



Override the “computeImageEdges()” method to detect the edges of the image. Edge detection can be achieved by convolving the 3x3 Sobel kernels with the original image to compute the approximations of the derivatives in both horizontal and vertical directions of the image. A simplified implementation can be achieved using the following process.

The pixel value of the (x,y) location of the edge image P for the original image A is given by:

P(x,y) = |(A(x-1,y-1) + 2 \* A(x, y-1) + A(x+1, y-1) ) - ((A(x-1,y+1) + 2 \* A(x, y+1) + A(x+1, y+1))| + |((A(x+1,y-1) + 2 \* A(x+1, y) + A(x+1, y-1)) - ((A(x-1,y-1) + 2 \* A(x-1, y) + A(x-1, y+1))|



Modify the main() in “FileApplicationTester” class to compute and view the edges of the image provided in the “supplementary\_materials” folder.