**Filip Horak**

**Nurgyun Hasanova**

**Victor Cervantes**

**Question #A.1: What is the first file your group looks at and why? What are the first (and most important) things that your group takes notice of in the code in this file?**

The first file we looked at is “image.java”. Upon opening the file, we noticed the different methods that were created in the image class. We can observe that there are multiple packages that are imported as well.

**Question #A.2: What happens in the constructor?**

The constructor of image class is overloaded and takes input filename of type String. It even throws an exception if the input is not in the desired format.

**Question #A3: Find the read method (note that the read method is not complete). How many parameters does the read method take and what are their types? What is its return type? How many objects have been created in this method?**

The read method takes in one parameter which is filename of type String. The method’s return type is void which means that the method doesn’t return anything. Two objects are created in the method, a File object and BufferedImage.

**Question #A.4: Notice that the read method has an additional part appended to the end of the method signature: ​throws Exception. ​Use Google to find out the role of this additional part ​ what does it do? Where did you find your answer?**

When an error occurs within a method, the method creates an object and hands it off to the runtime system. The object, called an exception object, contains information about the error, including its type and the state of the program when the error occurred. Creating an exception object and handing it to the runtime system is called throwing an exception.

After a method throws an exception, the runtime system attempts to find something to handle it. The set of possible "somethings" to handle the exception is the ordered list of methods that had been called to get to the method where the error occurred. The list of methods is known as the call stack.

Source: <https://docs.oracle.com>

**Question #A.5: The first line of the read method creates a ​File ​object. What is a ​File object? In what Java package does the ​File ​class live? Where did you find your answers?**

A file object is the wrapper class for what you want java to manipulate. It is found in the IO package. We found the answer in the java API documentation on Oracle.com

**Question #A6 : The second line of the read method creates a BufferedImage ​object by reading in the ​File​object using the ​ImageIO ​class. What is a ​BufferedImage ​object? More specifically ​ what is a ​BufferedImage ​object composed of? In what Java package does the ​BufferedImage ​class live? Where did you find your answers?**

A BufferedImage object is composed of a Raster (a rectangular array of pixels) and a ColorModel (translates the integer values to RGB colors) and lives in the awt.image package.

**Question #A7: In Java, how are images represented? What are they composed of? This is a very important question! In order to be able to manipulate the images (flip them around, etc), you have to be able manipulate the pieces of the image in Java**

Images are represented by its width, height, pixels and coordinates(x,y). We have methos in Image class for loading, drawing and to manipulate image files. Images are arrays of bits.

**Question #A.8: After the ​BufferedImage ​object is created, the width and height instance variables are set using the ​getWidth ​and ​getHeight ​methods from the BufferedImage ​class. What do these methods do? Hint: If you wrote down that they give you the width and height of the image, you are wrong! How do they relate to the "pieces" of an image?**

The getWidth and getHeight methods get the x and y coordinates of the grid. Each set of coordinates represent one pixel. This cartesian system allows the picture to be copied or manipulated.

**Question # B1) Is there anything in the Image.java code that makes you think of a grid? What is it?**

The 2d array called pixels integer array makes us think of a grid.

**Question # B2) The pixels instance variable has been declared, but not created. How big is this array instance variable? Create it and compile your code**

Since the Pixels instance variable has been only declared but not created. Its size is null.

**Question # B.3) Now that you have a 2D integer array of all 0s that is the same size as your image, you need to assign each pixel from your BufferedImage to its corresponding value in the array. Find the Java documentation for the BufferedImage class. Is there a method that returns an integer value for each piece of the image? Hint: Remember that each position in an image can be thought of as a grid where the point (0, 0) is in the upper left​hand corner. Use the method and assign it to the correct element in your 2D array. Compile your code.**

We use getRGB method to return integer value for each pixel.

**Question #C.1 Find the createBufferedImage method. What is the return type of this method? Using the information about the BufferedImage class from your previous research, what does this method do? Why would this method be useful for manipulating an image?**

The return type of createBufferedImage is of type BufferedImage. It creates a new object ‘bufImage’ that takes in integer parameters named width, height, BufferedImage.TYPE\_INT\_RGB i.e which is an image with 8-bit RGB color components packed in it. This method is useful because we create an image using this method based on the values of rows, col and pixel array values.

**Question #C.2 Find the write method. What is the return type of this method? How many parameters does this method take and what are their types? What does this method do (be specific!)? Why would you want to use this method?**

The return type of the write method is void, so it does not return anything. This method takes one parameter which is of type String. It creates an object ‘fileImage’ and throws in exception if cannot create it. You would want to use this method to call the code that implements the plug-in (jpg) writing.

**Question #C.3 Find the draw method. What is the return type of this method? How many parameters does this method take and what are their types? What are the second and third parameters used for? What is a Graphics object and what does its drawImage method do? How will this method (the draw method of the Image class) be useful?**

The draw method has no return type and it uses three parameters, one object and two integers. The second and third parameters are the integers and they are used for placing the pieces of the image on the cartesian plane - coordinates. A graphic object stores the information which is needed for executing operations which java supports. This instance contains information where to draw the image and what type of color or font to use for it. The drawImage method enables you to draw as much of the specified area of the image as possible and then scales the image to size. This method is useful for making the new image.

**Question #D.1 What do the first 4 lines in the main method do?**

The first line sets up an object called frame and titles it "My Images." The second line sets the size of the frame in pixels. The third line sets the frame as visible. The fourth line sets up a new object, gc, of class Graphics that gets graphics onto the frame.

**Question #D.2 Notice that there is some code that looks like a block of code ​ with the words try and catch, but with nothing inside the braces that follow the word try. This is called a try​catch block. You will learn about this in detail later in the class. Using Google, find out what a try​catch block should do and explain it.**

The try-catch block can be used to handle code where an exception may occur. The try block contains the

code and the catch block is used to handle the potential exeptions. The catch block throws an exception

if it occurs. For this program the try-catch block is used to handle the exception of a file not being found. The try code contains the code and the catch contains the exception and what to print as an error.

**Question #D.3 + coding You want to make a new Image object using the Image class that you finished. What type of parameter does your Image class constructor require? Inside of the try​block, create a new Image object using the animals image provided. Make sure that the animals.jpg file, the Image.java class, and the TestImage.java class are all in the same folder! Then, call the draw method on the Image object that you just created and pass in the Graphics object, 10 for the parameter x, and 40 for the parameter y. Compile and run the TestImage class**

The image class constructor contains the name of the file in the form of a string. For this example we use "animals.jpg" but this will work for any file name.

**Question #E.1 + coding The first method that your boss wants you to create in the Image class is named flipY. This method should take the image and flip it around the y​axis. Think of this as what you would see if you looked at the image in a mirror. Everything would be reversed horizontally, but not vertically (i.e. not upside​down). Should your method take any parameters? Why or why not? Should your method return anything? Why or why not? Create the code for this method and then compile. Then, call your method on the Image object in the TestImage class and make sure to draw it (place it next to the original image!). Compile and run your TestImage class**

The flipY method takes A string argument and doesn't return anything. The string will be the file name of the image file wanted to be flipped.

**Question #E.2 + coding The second method that your boss wants you to create in the Image class is named flipX. This method should take the image and flip it around the x​axis, meaning that the modified image should be upside​down. Should your method take any parameters? Why or why not? Should your method return anything? Why or why not? Create the code for this method and then compile. Then, call your method on the Image object in the TestImage class and make sure to draw it (place it next to the original image!). Don’t forget to comment out the code for flipping the image horizontally, otherwise your image will end up reversed and upside down! Compile and run your TestImage class**

The flipX method takes A string argument as well. It doesn't return anything either. The string will be the file name of the image file wanted to be flipped.

**Question #F.1 What was the most challenging part of this research lab for your group?**

The most challenging part for the group was getting together to work on the project. We all have very different schedules apart from this class. Once we got together we worked very efficiently. On the problems themselves the hardest part for us was figuring out the read method of the Image class.

**Question #F.2 What did your group learn/find the most useful by doing this research lab?**

The most useful thing we learned from this lab is the importance of research in the field of computer science. This lab gave us the opportunity to solve a problem on our own by researching methods online. The information we found through Oracle's java API website was extremely useful in solving our problem.

**Question #F.3 What was the most fun aspect of doing this research lab?**

We enjoyed running the final code and had fun rotating with different images. It was also fun to learn how java manipulates images and treats them as a two-dimensional array.