Homework 4: Collage & Picture Flip

Due: 5:00pm, Friday, October 21st

Set Up

- **Read this entire write-up first before starting and START EARLY!! This HW can be long!**
- **Open your home directory (double click on your cs8fxx icon; 2\textsuperscript{nd} icon from the top) and create a new directory in your home directory called hw4 (all lowercase).**
- **Copy the folder bookClasses from your Desktop into the hw4 directory.**
  - Right click on bookClasses > Copy > Right click on hw4 > Paste Into Folder
- **When you work in DrJava be sure you are editing files in the hw4/bookClasses directory.**

Problems

1. Collage

For this part of the assignment, you will be making a collage. The first file you will create is HW4A.java. Here is a template for HW4A.java (use HW4B – later in this write-up – to help you complete this).

```java
/*
* Filename: HW4A.java
* Name: Your name and your partner’s name
* Login: Your cs8fxx account and your partner’s cs8fxx account
* Date: Month, Day, Year
* Sources of Help: ... (for example: names of people, books, websites, etc.)
*
* Describe what the code does here.
*/
public class HW4A {
  /*
   * Name: method name
   *
   * Purpose: Briefly describe the purpose of this method
   *
   * Parameters: List all parameters and their types and what they represent.
   * If no parameters, just state None.
   *
   * Return: Specify the return type and what it represents.
   * If no return value, just specify void.
   */
  public static void main ( String[] args )
  {
    // Insert code for HW4A here.
  }
}
```

This program should allow the user to choose a picture using the Filechooser.pickAFile() method and then make three copies of it. Then you should call a different filter method on each of the 3 copies of the picture (the original should not be changed, only the copies should get modified).

You should then send the 3 modified pictures to a collage() method (which you will add within the Picture class) that will put them all in one picture, side by side. After the collage is created, it would look something like this:
The three filter methods are going to be two filters of your choice plus a mirror filter. A mirror filter can be found in the book or as described below:

```java
public void mirrorVertical()
{
    int width = this.getWidth();
    int mirrorPoint = width / 2;
    Pixel leftPixel = null;
    Pixel rightPixel = null;

    for( int y = 0; y < getHeight(); y++ )
    {
        for( int x = 0; x < mirrorPoint; x++ )
        {
            leftPixel = getPixel( x, y );
            rightPixel = getPixel( width - 1 - x, y );
            rightPixel.setColor( leftPixel.getColor() );
        }
    }
}
```

You can also use a horizontal mirror if you choose. **But whatever filter you choose, document what it does within the method header.**

Now, some details about the `collage()` method. The `collage()` method will take in all three pictures after the filters are applied and copy them onto a canvas (blank picture) that is the **exact size** of all three put together side by side. **Thus the collage picture should be the same height, but three times the width of the original picture.**

Overall this part of the assignment will put three pictures that are run through different filters next to each other on one picture.
Here is what you have to do:

- Add three methods to Picture.java.
- Each method should change its calling object (this).
- Choose your favorite two filters, they can be ones that you created yourselves in the previous assignment, or ones from the book such as negate(), greyscaleWithLuminance(), clearBlue(), lighten(), darken(), etc. from Chapter 4.
- Your third filter should either be a horizontal or a vertical mirror image (found above or on page 135-136 of your text).
- These filtering methods must change the calling object. We are encouraging you to be creative.
- Now add a collage method to Picture.java: (Don’t forget to write a method header for it.)

```java
public void collage( Picture pic1, Picture pic2, Picture pic3 )
```

- This method should take in 3 parameters. These are the Pictures that will be placed onto the calling object (this). **Hint:** You need to create a new picture with height and width values first before calling collage() on it.
- You should have 3 nested for loops to copy the pixels from the 3 filtered pictures.
- The loops will look something like below. (Each of these loops are going to iterate over each and every pixel of the source picture):

```java
// Add the first picture
for (int sourceX = 0; sourceX < pic1.getWidth(); sourceX++)
{
    for (int sourceY = 0; sourceY < pic1.getHeight(); sourceY++)
    {
        // code to copy the pixels go here
        // target pixel is (sourceX, sourceY)
    }
}

// Add the second picture
for (int sourceX = 0; sourceX < pic2.getWidth(); sourceX++)
{
    for (int sourceY = 0; sourceY < pic2.getHeight(); sourceY++)
    {
        // code to copy the pixels go here
        // target pixel is (sourceX + pic1.getWidth(), sourceY)
    }
}

// Add the third picture
for (int sourceX = 0; sourceX < pic3.getWidth(); sourceX++)
{
    for (int sourceY = 0; sourceY < pic3.getHeight(); sourceY++)
    {
        // code to copy the pixels go here
        // target pixel is (sourceX + pic1.getWidth() + pic2.getWidth(), sourceY)
    }
}
```
Notice how the destination pixel changes from \((sourceX, sourceY)\) to \((sourceX + pic1.getWidth(), sourceY)\) and then finally to \((sourceX + pic1.getWidth() + pic2.getWidth(), sourceY)\). Can you think why this is so? This is because the first image gets drawn from the beginning, but each subsequent picture has to be shifted by the width of the previous image. The below diagram will make it more clear.

Using methods in an application:

- Open the file HW4A.java in your hw4/bookClasses directory.
- In this file you will choose one picture, but copy it twice, so that you have three instances of the same photo.
- You should show() the original picture (just one instance is fine).
- You should then create a blank Picture that is the same height as the one you chose, but three times as wide. Use the Picture constructor that takes in a width and a height parameter for this purpose:

  ```
  public Picture(int width, int height)
  ```

- Then call the 3 filter methods you wrote in hw4/bookClasses/Picture.java on the 3 copies of the picture to produce 3 different effects.
- On the blank Picture you created, you should call the collage() method, sending in the three pictures you altered.
- Then you should show() the final image and write it to /tmp:

  ```
  collagePic.write("/tmp/collage_name1_name2.jpg");
  ```

  where name1 and name2 are your cs8fXX login names.
See the following image for an example. This is the barbara.jpg collage, the first is grayScale(), the second is createPattern() and the third is mirrorHorizontal().

DO NOT USE barbara.jpg!!! Use a different picture or one of you own.

2. Picture Flip

See the following picture for understanding the flipping (below is an example of vertical flipping):

Method 1: flipVertical()

- Add a method called flipVertical() to Picture.java
- This method will take in 3 parameters, the x and y coordinates of the middle pixel of the box to flip, and the size (length) of the square box to flip.
- First, this method should figure out the upper left-most x and y coordinates and save those to local variables. Then figure out the lower right-most x and y coordinates and save those to other local variables.
- Then you must have a nested for loop. This will loop through the proper x and y coordinates and allows you to flip the box specified by the parameters.
- The outer for loop should loop through all y coordinates from the top left-most y coordinate to the bottom right-most y coordinate.
- The inner for loop should loop through half of the x coordinates from the top left-most x coordinate to the x coordinate passed in as the x parameter.
- Inside of the nested for loops you should get the target pixel, which is the pixel that the for loops are on, and the source pixel, which is the x coordinate opposite of the target pixel. Then you should swap their colors.
Method2: flipHorizontal()

- Add another method to Picture.java called flipHorizontal()
- This method is almost exactly like flipVertical() except that the flipping is about the horizontal axis. The outer loop will loop through all of the x pixels, from the upper left-most x to the bottom right most x. The inner loop will loop through half of the y pixels.
- The inside of the nested loops are pretty much the same, except backwards, we expect you to mess around with it and think about it to figure it out.

Using methods in an application:

Create a new file HW4B.java in your hw4/bookClasses directory. You can start with this template here:

```java
import java.io.*;

public class HW4B {
    public static void main(String[] args) throws IOException {
        /* Choosing a picture and initializing variables. */
        Picture pic = new Picture(FileChooser.pickAFile());
        pic.show();

        /* Create two copies */
        Picture pic1 = new Picture(pic);
        Picture pic2 = new Picture(pic);

        /* Prompt the user for coordinates. */
        String prompt = "Please enter a position (first x, then y) in the " +
                        "picture to flip vertically."
        System.out.println(prompt);

        /* Converting coordinates to ints. */
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int x = (new Integer(br.readLine())).intValue();
        int y = (new Integer(br.readLine())).intValue();

        /* Prompt the user for size. */
        System.out.println("Please enter the size of the box to flip.");
        int size = (new Integer(br.readLine())).intValue();

        pic1.flipVertical(x, y, size);
        pic2.flipHorizontal(x, y, size);
        pic1.show();
        pic2.show();
    }
}
```

NOTE: Please start with this template as it has a large amount of the work done for you.
• Create a new picture using FileChooser.pickAFile().
• Show the original picture.
• Create two copies: one for vertical flip and one for horizontal flip.
• Call flipVertical() on one copy and flipHorizontal() on the other copy.
• Test with different x,y coordinates and flip box sizes. **You do not need to deal with boundary cases and out-of-bounds errors in this assignment (you will in future assignments).**
• Here is the picture we want to alter: barbara.jpg

![Image](image1.png)

• The following image is the barbara.jpg with a vertical box flipped of size 50 at (100,100)

![Image](image2.png)

• The following image is the barbara.jpg with a horizontal box flipped of size 50 at (100,100)

![Image](image3.png)
How to turn in your homework electronically

Only ONE partner should run the turnin script - do not submit two copies of the same homework.

Make sure the program works correctly in your cs8fXX login on the workstations in the labs in the basement of the CSE building when the workstations are booted under Linux (CentOS).

When you are ready to turn your program in, first open a terminal window by right clicking on the empty area of your computer's desktop and select "Open a new Terminal." Then type in the following command

    turnin hw4

Three files are required to be submitted for hw4 (all 3 files must be in your hw4/bookClasses directory):

- Picture.java (with the new methods you wrote for this assignment)
- HW4A.java
- HW4B.java

You can verify your turnin with the following command

    verify hw4

It is your responsibility to make sure you properly turned in your assignment.

START EARLY!!!!