Lab 4: Learning about Picture and Pixel Objects

Objectives:

1. To gain more understanding of objects and methods in Java using the Picture class.
2. To gain an understanding of Picture objects and Pixel objects.

Preparation:

1. Go over the Lecture Notes: Topic 3, Topic 4, Topic 5.
2. Textbook reading: Chapter 3 section 3.6; Chapter 4 sections 4.1 and 4.2

Exercise 0: Setting up the mediaSources folder

The exercises in this lab as well as in future labs use images (pictures) provided by the textbook authors. These images may be downloaded from a link provided on our course website in the **Resources** folder within the folder **DrJava Information** – it is a file called `mediasources-no-movies-7-30-06.zip.`

Download the mediaSources zip file (`mediasources-no-movies-7-30-06.zip`) from the above link and save it in your home directory.

- The file you downloaded is a compressed file. To uncompress it, double click on the filename. This will start up the WinZip utility.
- In WinZip, click on the Extract button. In the *Extract to* window enter `Z:\mediaSources`, then click on *Extract*. This will save the media files to a folder called mediaSources in your home directory.
- Add the mediaSources folder to the path that DrJava will use to look for Java files: In the *Edit* menu, click on *Preferences*. Click on *Add*, then *Select* your mediaSources folder. Click on *OK*.

Exercise 1: Finding media files in DrJava

There are many files containing images in the mediaSources folder. This exercise shows you how to pick a file from the mediaSources folder and display the image stored in it on the screen. You will do this in the Interactions pane.

a) Select a file with a picture in it by typing the following statement in the Interactions pane. When the file window appears, find the folder mediaSources and in that folder select the file named *butterfly2.jpg*

```
String fileName = FileChooser.pickAFile();
```

b) Create a Picture object that will contain the image from the file:
```
Picture pictureObj = new Picture(fileName);
```
What is the name of the reference variable that refers to that Picture object?

c) To actually see the picture on the screen, type the following:
```
pictureObj.show();
```

d) Now type Java statements in the Interactions pane that will allow you to select the file *beach.jpg* from the mediaSources folder and display it on the screen. (You can use the same variable names as above, but be careful that you do not redeclare them.)

e) Leave the Interactions pane as it is (do not Reset it), since you will be using this `pictureObj` in the next exercise.

Exercise 2: Picture Explorer

We have learned that a picture (an object of the Picture class) is composed of many pixels (objects of the Pixel class). Each pixel has an (x,y) position in a picture, and has its color defined by red, green and blue components. The Picture Explorer is a tool provided by the textbook authors that lets us explore a picture by moving the cursor around in the picture; for the pixel at the cursor, we can see the x and y position values, the colour of the pixel, and its red, green, blue components. To use this tool on the beach picture from Exercise 1, type:

```
pictureObj.explore();
```

What are the red,green,blue values for the pixel at (0,0)? for a pixel somewhere in the sand? for a pixel somewhere in the water? What are the coordinates of the pixel at the very bottom right-hand corner?

Exercise 3: Properties of Pictures and Pixels

In this exercise, you will explore properties of Picture objects and of Pixel objects. You are still using the Picture object from Exercises 1 and 2.

a) Type the following statement:
```
System.out.println(pictureObj);
```
What properties of the Picture object are printed?

b) You can get the width and height of a Picture object by using the methods `getWidth` and `getHeight` of the Picture class. Type the following:
```
int width = pictureObj.getWidth();
int height = pictureObj.getHeight();
```

Now type a Java statement that will print the width and height in the format "`Picture size is` *(width will go here)* `by` *(height will go here)*".
Then print the number of pixels in the picture (Hint: use width and height).

c) The Picture class has a method getPixel that can be used to get a pixel of a picture at a specific location in the picture. Type the following statement to get the pixel at the top left-hand corner, i.e. at coordinates (0,0) of your picture, and store a reference to it in the variable `pixelObj1`:

```
Pixel pixelObj1 = pictureObj.getPixel(0,0);
```

Then type the following statement:

```
System.out.println(pixelObj1);
```

What properties of the Pixel object are printed?

d) Type the statements to print the properties of the pixels at the top right-hand corner of the picture, at the bottom left-hand corner, and of some pixel in the middle of the picture. Use different variable names to refer to each of these Pixel objects (for example `pixelObj2`, `pixelObj3`,and `pixelObj4`).

e) You can get the red, green and blue values of a Pixel object by using the methods `getRed`, `getGreen` and `getBlue` of the Pixel class. Type the following statement to get and store the red value of the pixel at (0,0):

```
int redValue = pixelObj1.getRed();
```

Now type similar statements to get and store the green and blue values of this pixel, and then print these values in the format

"`Pixel red, green, blue values are:` *`(red, green blue values will go here, separated by commas)".`*

Exercise 4: Working with Pictures in a Program

You will now create an application program in a class called ShowProperties that will call (invoke) the methods of the Picture class and the Pixel class used in the previous exercises. The main method of ShowProperties will get an image from a file, display the picture on the screen, print its width and height, get the pixel at specific coordinates (x,y) in that picture, and print the pixel's color values. Much of the code is provided for you, but you should add the Java statements as specified by the comments within the code. Enter the code for the program in the Definitions pane and save it as ShowProperties.java.

```
import java.awt.*;
public class ShowProperties
{
  public static void main (String[] args)
  {

    /* add code here to choose a filename using FileChooser,
       referenced by the variable fileName */

    Picture picObj = new Picture(fileName);

    /* add code here to display the picture on the screen */

    /* add code here to get the width and height of the picture
       and store them in the variables picWidth and picHeight */
```

```
    System.out.println("The picture is " + picWidth +
                       " pixels wide and " + picHeight + " pixels high");

    int xcoord = 320;
    int ycoord = 240;
    Pixel pixelObj = picObj.getPixel(xcoord,ycoord);

    /* add code here to get the red, green and blue values for the pixel
   and store them in the variables redValue, greenValue and blueValue */

    System.out.println(" Colors of pixel at (" + xcoord + "," + ycoord +
    ") are: red = "+redValue+" green = "+greenValue+ " blue = "+blueValue );

  }
}
```

a) Save the program in ShowProperties.java, compile it, and run it, choosing the image file *blueMotorcycle.jpg* from the mediaSources folder.

b) What happens if the pixel coordinates are greater than the size of the picture? Change the values of `xcoord` and `ycoord` so that they are outside the picture and compile and run the program. What error message do you see in the Interactions pane?

Exercise 5: Return Values from Methods

The methods getWidth and getHeight of the Picture class return a value (an integer), as do the methods getRed, getGreen, and getBlue of the Pixel class. In the preceding exercises we invoked these methods and stored the returned values in variables (`picWidth, picHeight, redValue`, etc.). We then printed the contents of the variables. We could also have invoked the methods directly in the print statements, without storing the returned values in variables, for example:

```
System.out.println("The picture is " + picObj.getWidth()+ " pixels wide and "
                   + picObj.getHeight() + " pixels high");
```

Create a new program in a class ShowProperties2 that is exactly the same as your ShowProperties of Exercise 4, but that invokes the methods getWidth, getHeight, getRed, getGreen and getBlue from the print statements rather than using variables to store the returned values.

Save your program in ShowProperies2.java, compile and run it; it should produce exactly the same results as your program of Exercise 4.