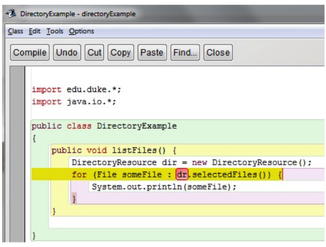
Batch Grayscale Images

Quiz, 6 questions

1 point

1. Consider writing code to list out files in one of your folders using **DirectoryResource**. The following attempt at writing such code has an error.



Which one of the following best explains what the error is?

- The **dr** that is highlighted should be **File**.
- The **dr** that is highlighted should be **DirectoryResource**.
- The **dr** that is highlighted should be **dir**.
- The **dr** that is highlighted should be **listFiles**.

1 point

2. Shown below is the code that was developed in one of the videos to convert many images to grayscale and to display those grayscale images.

```
import edu.duke.*;
import java.io.*;
public class GrayScaleConverterBatch
{
  public ImageResource makeGray(ImageResource inImage) {
    ImageResource outImage = new ImageResource(inImage.getWidth(), inImage.getHeight());

    for (Pixel pixel: outImage.pixels()) {
      Pixel inPixel = inImage.getPixel(pixel.getX(), pixel.getY());

      int average = (inPixel.getRed() + inPixel.getGreen() + inPixel.getBlue())/3;

      pixel.setRed(average);
      pixel.setGreen(average);
      pixel.setBlue(average);

    }

    return outImage;
  }

  public void selectAndConvert() {
    DirectoryResource dr = new DirectoryResource();
    for (File f: dr.selectedFiles()){
      ImageResource inImage = new ImageResource(f);
      ImageResource gray = makeGray(inImage);
      gray.draw();
    }
  }
}
```

Consider adding additional code to this program to save each of the new grayscale images created as a file.

Which one of the following is the best method to modify to make this change?

- Modify the **selectAndConvert** method.
- Modify the **makeGray** method.
- Create a new method to read each image from a file and save the image.

1 point

3. Suppose one wants to convert a given image to grayscale and then display and save the resulting grayscale image as a file.

The code below has been started for you. The variable **f** is a file of an image and the method **makeGray** returns an image that is the grayscale image of the original image.

```
1 ImageResource original = new ImageResource(f);
2 ImageResource grayscale = makeGray(original);
3 [MISSING CODE]
```

Which one of the following is the missing code that will convert the original image into a new file that is a grayscale version of the original image?

```
1 String fname = original.getFileName();
2 String newName = "grayscale-" + fname;
3 grayscale.setFileName(newName);
4 grayscale.draw();
5 grayscale.save();
```

```
1 String fname = original.getFileName();
2 String newName = "grayscale-" + fname;
3 original.setFileName(newName);
4 grayscale.draw();
5 grayscale.save();
```

```
1 String fname = grayscale.getFileName();
2 String newName = "grayscale-" + fname;
3 grayscale.setFileName(newName);
4 grayscale.draw();
5 grayscale.save();
```

```
1 String fname = original.getFileName();
2 grayscale.setFileName(newName);
3 String newName = "grayscale-" + fname;
4 grayscale.draw();
5 grayscale.save();
```

1 point

4. Consider writing a program to create new images that are photographic negatives (or inverted images) of selected images.

In inverting an image, a pixel's red, blue and green components are modified to be the exact opposite within the 0 to 255 range. That is, if a pixel's red, blue, and green values are (34, 198, 240) then that same pixel in the inverted image would have the red, blue, and green values of (221, 57, 15). Note that 255 - 34 is 221, 255 - 198 is 57, and 255 - 240 is 15.

For example, these images show the original and inverse images of Robert.

Suppose a pixel has RGB values of (100, 30, 250).

Which one of the following shows the correct RGB values for the inverted pixel?

- The RGB values would be (100, 30, 250).
- The RGB values would be (155, 225, 5).
- The RGB values would be (126, 126, 126).
- The RGB values would be (225, 155, 5).
- The RGB values would be (200, 230, 50).

1 point

5. Consider writing a program to create new images that are photographic negatives (or inverted images) of selected images.

Suppose we have a Pixel named **pxInvert** and a Pixel named **pxOriginal**.

What is the line of code to change **pxInvert**'s red color to the inverted red color of **pxOriginal**?

Hints: Start with **pxInvert.setRed**

Remember the semi colon at the end.

pxInvert.setRed(255-pxOriginal.getRed( ));

1 point

6. Consider writing a program to create new images that are photographic negatives (or inverted images) of selected images.

Suppose we have an ImageResource variable named **picture** whose current value is for an image file named **dragon.png**. See the following code segment below.

```
1 ImageResource invertImage = makeInverted(picture);
2 String fname = picture.getFileName();
3 invertImage.setFileName("inv-" + fname);
4 invertImage.draw();
5 invertImage.save();
```

What is the name of the resulting file?

inv-dragon.png

I, **Ning Zheng**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Submit Quiz