CS 1110 Assignment 4

1. Pixel Class

Name: `Pixel.java`

A color can be represented numerically by its color components: red, blue, and green (RGB). A common way to represent RGB values is to allow each component to be a value between 0 and 255, with a value of 0 being the minimum for that color component and a value of 255 being the maximum for that color component. For example, (0, 255, 0) would be completely green without any red or blue. (0, 128, 128) would be a mix of blue and green with no red, so it would be yellow. With this system, we can represent 256^3 = 16,777,216 possible colors. A pixel (picture element) is basically a small square that is a particular color.

Create a Pixel class with the following items:

Attributes (private):
- Red (int)
- Green (int)
- Blue (int)

Constructors:
- No argument constructor (set the attributes to default values of 0)
- Constructor that allows the attributes to be set

Operations (public):
- toString() to return a string representation of the Pixel in the format (*red*, *green*, *blue*) (e.g., (0, 128, 0))
- Getters for each attribute
- Setters for each attribute
  - Validate that the value is between 0 and 255, inclusive

Static Methods (public):
- public static Pixel createRandomPixel() – return a Pixel object with random color components

Add a main() method to the Pixel class that creates two Pixel objects, one using the no arg constructor and one using the other constructor. Use the setters to set values for one of the Pixel objects. Use createRandomPixel() to create a third pixel. Display the result of all three Pixel object's toString() methods.

Example Output:

```
pixel1: (0, 0, 0)

pixel2: (0, 255, 128)

Random pixel: (218, 77, 41)
```

2. Image Class

Name: `Image.java`

An image is essentially a grid of pixels. Thus, we can use a 2D array of Pixel objects to represent an image.

Create an Image class with the following items:

Attributes (private):
- Width (int)
- Height (int)
- Data (Pixel[][])

Constructors:
- No argument constructor (set the attributes to default values)
- Constructor that allows the attributes to be set

Operations (public):
- toString() to return a string representation containing the dimensions and data:

  Dimensions: 5 by 5
  Data:
  (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0)
  (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0)
  (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0)
  (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0)
  (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0)

- Getters for each attribute
- Setters for each attribute
  - Validate width and height are greater than zero
- public void setPixel(int row, int column, Pixel pixel) that allows a specific pixel in the data to be set

Add a main() method to the Image class that creates an Image object. Add five random pixels at random locations in the image. Display the result of the Image object's toString() method.

Example Output:

```
image1:
Dimensions: 5 by 5
Data:
 (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0)
 (0, 0, 0) (179, 183, 240) (0, 0, 0) (0, 0, 0) (14, 186, 164)
 (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0)
 (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0)
 (252, 70, 127) (0, 0, 0) (0, 0, 0) (123, 94, 55) (115, 206, 114)
```