COMP 2631 (Winter 2015)
Data Structures and Algorithms II
Assignment 3 — Mandelbrot Set GUI

Due date: **Saturday, March 28, by 11:55pm** (submit to Moodle)

Overview

The challenge of this assignment is to build a GUI that will render the Mandelbrot set on a canvas, and will allow the user to perform simple manipulations.

You are required to write two (*and only two!*) classes:

- `Complex`
- `MandelbrotGUI`

An object of the `Complex` class holds the information for a single complex number, and contains methods that perform standard operations on this complex number.

The `MandelbrotGUI` class calculates and draws a portion of the Mandelbrot set lying in a square region of the complex plane. It also provides GUI components that allow the user to move and resize this square region.

The `Complex` class

The `Complex` class has two instance variables, both of type `double`—these hold the real and imaginary parts of a complex number. Include the following constructors and methods:

```
// Standard constructor.  Sets the real and imaginary components.
public Complex(double inReal, double inImag)

// Copying constructor.  Sets the real and imaginary components of
// this complex number to be equal to the corresponding components of
// the complex number passed as an argument.
public Complex(Complex c)

// Multiplies this complex number by the complex number passed as an argument.
// Stores the result back in this complex number.
public void multiply(Complex toMult)
```

```
// Adds this complex number to the complex number passed as an argument.
// Stores the result back in this complex number.
public void add(Complex toAdd)

// Subtracts the complex number passed as an argument from this complex number.
// Stores the result back in this complex number.
public void subtract(Complex toSub)

// Returns the real part of this complex number.
public double getReal()

// Returns the imaginary part of this complex number.
public double getImag()

// Returns the modulus of this complex number (distance from origin).
public double modulus()
```

The `MandelbrotGUI` class

In this section, the instructions are less specific than usual. You have freedom in how you structure your code, as long as it is well thought out and well documented (use Javadoc-style comments).

1. The canvas on which you draw the Mandelbrot set should have size 600×600.

2. When your program starts running, it should display the Mandelbrot set lying in a square region of the complex plane whose top-left corner is at $-2.2 + 1.8i$ and whose side has length 3.6.

3. For the canvas, I recommend using a custom class that extends `JPanel`, as specified in the Bitmap GUI assignment. As you know, you can easily draw a `BufferedImage` on such a canvas.

4. Your `main` method should construct a `MandelbrotGUI` object (and do nothing else).

5. Recommended method: `private int doIterations(Complex c)`
   This method takes a `Complex` object representing a point in the complex plane corresponding to a pixel, and returns the number of iterations of the "Mandelbrot Rule" before the resulting sequence of points goes outside a circle of radius 2 (or returns some maximum value stored in a class constant, e.g., 150).

6. Recommended method: `private Color pickColor(int numIter)`
   This method takes in a number of iterations (as returned by `doIterations`) and assigns a color to the corresponding pixel. You have complete freedom in how you choose colors, as long as the shape of the Mandelbrot set is clearly visible and the final result is visually pleasing/interesting.

7. At minimum, your GUI should include components that provide the following functionality to the user:

   - **Move Left, Move Right, Move Up, Move Down** — These change the position of the square region of the complex plane being rendered (but not its size). Each of these options should move this region approximately 20% in the indicated direction. (Modify this 20% value if it seems too large/small.)
   - **Zoom In, Zoom Out** — Play around with the magnitude of the zooming to find appropriate amounts. A nice idea is to use buttons with appropriate magnifying glass icons.
   - **Default** — Restore the default Mandelbrot image.
   - **Exit** — Exit the program.