```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
/**
 * JPanel for Lab03: Hailstone Numbers. Creates a label, a textField,
 * 3 buttons labeled Set, Next, and Quit. Allows the user to find the
 * hailstone numbers. Each button has a tailor-made ActionListener.
 *
 * @author Josh Ibad
 * @author James Park
 * @version 17 November 2017
 * @teacher Coglianese
 * @period 2
 */
public class Panel03 extends JPanel
{
    private JLabel label1, label2;
    private JTextField box;
    private int number, count;
    /**
     * Constructs the JPanel with a FlowLayout, containing a JLabel,
     * a JTextField, and 3 JButtons labeled Set, Next, and Quit,
     * each button is given a listener in order to fulfill their
     * respective functions. Another label is added, displaying the
     * number of iterations of hailstone number algorithms was used
     * to obtain the number being displayed.
     */
    public Panel03()
    {
        setLayout(new FlowLayout());

        label1 = new JLabel("37");
        label1.setFont(new Font("Serif", Font.BOLD, 100));
        label1.setForeground(Color.blue);
        add(label1);

        JPanel panel = new JPanel();
        panel.setLayout(new FlowLayout());
        add(panel);

        box = new JTextField("37", 5);
        box.setHorizontalAlignment(SwingConstants.RIGHT);
        panel.add(box);

        JButton button1 = new JButton("Set");
        button1.addActionListener(new Listener1());
        panel.add(button1);

        JButton button2 = new JButton("Next");
        button2.addActionListener(new Listener2());
        panel.add(button2);

        JButton button3 = new JButton("Quit");
        button3.addActionListener(new Listener3());
        panel.add(button3);

        label2 = new JLabel("Iterations: 0");
        add(label2);
    }

    /**
     * Handles actions for the Set button. Sets the number in the
     * textField as the number to be displayed by the label. Resets
     * the iteration count.
     */
    private class Listener1 implements ActionListener
    {
        /**
         * Sets the number in the textField as the number to be
         * displayed by the label.
         *
         * param e      ActionEvent, that is, the clicking of the
         *              button by the user.
         */
        public void actionPerformed(ActionEvent e)
        {
            label1.setText(box.getText());
            label2.setText("Iterations: 0");
        }
    }
    /**
     * Handles actions for the Next button. Generates the next
     * number in the hailstone numbers and increases displayed
     * iteration count.
     */
    private class Listener2 implements ActionListener
    {
        /**
         * Generates the next number of the hailstone numbers and
         * displays it. Increases iteration count.
         *
         * param e      ActionEvent, that is, the clicking of the
         *              button by the user.
         */
        public void actionPerformed(ActionEvent e)
        {
            int x = Integer.parseInt(label1.getText());
            if(x%2 == 0)
            {
                label1.setText("" + x/2);
            }
            else
            {
                label1.setText("" + (x*3 + 1));
            }
            label2.setText("Iterations: " + (Integer.parseInt(label2.getText().substring(12)) + 1));
        }
    }
    /**
     * Handles actions for the Quit button. Quits the program.
     */
    private class Listener3 implements ActionListener
    {
        /**
         * Terminates the Java Virtual Machine, effectively closing
         * the program.
         *
         * param e      ActionEvent, that is, the clicking of the
         *              button by the user.
         */
        public void actionPerformed(ActionEvent e)
        {
            System.exit(0);
        }
    }
}
```