```java
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

/**
 * Test class for NaturalNumber calculator GUI (view in MVC).
 *
 * @author Vaishnavi Kasabwala
 */
public final class NNCalcViewLab extends JFrame implements ActionListener {

    /**
     * Text areas.
     */
    private final JTextArea tTop, tBottom;

    /**
     * Operator and related buttons.
     */
    private final JButton bClear, bSwap, bEnter, bAdd, bSubtract, bMultiply,
            bDivide, bPower, bRoot;

    /**
     * Digit entry buttons.
     */
    private final JButton[] bDigits;

    /**
     * Useful constants.
     */
    private static final int TEXT_AREA_HEIGHT = 5, TEXT_AREA_WIDTH = 20,
            DIGIT_BUTTONS = 10, MAIN_BUTTON_PANEL_GRID_ROWS = 4,
            MAIN_BUTTON_PANEL_GRID_COLUMNS = 4, SIDE_BUTTON_PANEL_GRID_ROWS = 3,
            SIDE_BUTTON_PANEL_GRID_COLUMNS = 1, CALC_GRID_ROWS = 3,
            CALC_GRID_COLUMNS = 1;

    /**
     * No-argument constructor.
     */
    public NNCalcViewLab() {

        // Create the JFrame being extended

        /*
         * Call the JFrame (superclass) constructor with a String parameter to
         * name the window in its title bar
         */
        super("Natural Number Calculator");

        // Set up the GUI widgets --------------------------------------------

        // TODO: fill in rest of body, following outline in comments

        this.tTop = new JTextArea("", TEXT_AREA_HEIGHT, TEXT_AREA_WIDTH);
        this.tBottom = new JTextArea("", TEXT_AREA_HEIGHT, TEXT_AREA_WIDTH);

        this.bClear = new JButton("Clear");
        this.bSwap = new JButton("Swap");
        this.bEnter = new JButton("Enter");

        this.bAdd = new JButton("+");
        this.bSubtract = new JButton("-");
        this.bMultiply = new JButton("*");
        this.bDivide = new JButton("/");

        this.bPower = new JButton("Power");
        this.bRoot = new JButton("Root");

        this.bDigits = new JButton[11];

        for (int count = 0; count <= DIGIT_BUTTONS; count++) {
            JButton numbers = new JButton(Integer.toString(count));
            this.bDigits[count] = numbers;
        }

        // Set up the GUI widgets --------------------------------------------

        /*
         * Text areas should wrap lines, and should be read-only; they cannot be
         * edited because allowing keyboard entry would require checking whether
         * entries are digits, which we don't want to have to do
         */

        this.tTop.setEditable(false);
        this.tTop.setLineWrap(true);
        this.tTop.setWrapStyleWord(true);
        this.tBottom.setEditable(false);
        this.tBottom.setLineWrap(true);
        this.tBottom.setWrapStyleWord(true);

        /*
         * Initially, the following buttons should be disabled: divide (divisor
         * must not be 0) and root (root must be at least 2) -- hint: see the
         * JButton method setEnabled
         */

        /*
         * Create scroll panes for the text areas in case number is long enough
         * to require scrolling
         */

        JScrollPane inputScrollPane = new JScrollPane(this.tTop);
        JScrollPane outputScrollPane = new JScrollPane(this.tBottom);

        /*
         * Create main button panel
         */

        JPanel mainPanel = new JPanel(new GridLayout(
                MAIN_BUTTON_PANEL_GRID_ROWS, MAIN_BUTTON_PANEL_GRID_COLUMNS));

        /*
         * Add the buttons to the main button panel, from left to right and top
         * to bottom
         */

        mainPanel.add(this.bDigits[7]);
        mainPanel.add(this.bDigits[8]);
        mainPanel.add(this.bDigits[9]);
        mainPanel.add(this.bAdd);

        mainPanel.add(this.bDigits[4]);
        mainPanel.add(this.bDigits[5]);
        mainPanel.add(this.bDigits[6]);
        mainPanel.add(this.bSubtract);

        mainPanel.add(this.bDigits[1]);
        mainPanel.add(this.bDigits[2]);
        mainPanel.add(this.bDigits[3]);
        mainPanel.add(this.bMultiply);

        mainPanel.add(this.bDigits[0]);
        mainPanel.add(this.bPower);
        mainPanel.add(this.bRoot);
        mainPanel.add(this.bDivide);

        /*
         * Create side button panel
         */

        JPanel sidePanel = new JPanel(new GridLayout(
                SIDE_BUTTON_PANEL_GRID_ROWS, SIDE_BUTTON_PANEL_GRID_COLUMNS));

        /*
         * Add the buttons to the side button panel, from left to right and top
         * to bottom
         */

        sidePanel.add(this.bClear);
        sidePanel.add(this.bSwap);
        sidePanel.add(this.bEnter);

        /*
         * Create combined button panel organized using flow layout, which is
         * simple and does the right thing: sizes of nested panels are natural,
         * not necessarily equal as with grid layout
         */

        JPanel combinedPanel = new JPanel(new FlowLayout());

        /*
         * Add the other two button panels to the combined button panel
         */

        combinedPanel.add(mainPanel);
        combinedPanel.add(sidePanel);

        /*
         * Organize main window
         */

        this.setLayout(new GridLayout(CALC_GRID_ROWS, CALC_GRID_COLUMNS));

        /*
         * Add scroll panes and button panel to main window, from left to right
         * and top to bottom
         */

        this.add(inputScrollPane);
        this.add(outputScrollPane);
        this.add(combinedPanel);

        // Set up the observers ----------------------------------------------

        /*
         * Register this object as the observer for all GUI events
         */

        this.bDigits[9].addActionListener(this);
        this.bDigits[8].addActionListener(this);
        this.bDigits[7].addActionListener(this);
        this.bDigits[6].addActionListener(this);
        this.bDigits[5].addActionListener(this);
        this.bDigits[4].addActionListener(this);
        this.bDigits[3].addActionListener(this);
        this.bDigits[2].addActionListener(this);
        this.bDigits[1].addActionListener(this);
        this.bDigits[0].addActionListener(this);

        this.bClear.addActionListener(this);
        this.bSwap.addActionListener(this);
        this.bEnter.addActionListener(this);

        this.bAdd.addActionListener(this);
        this.bSubtract.addActionListener(this);
        this.bMultiply.addActionListener(this);
        this.bDivide.addActionListener(this);

        this.bPower.addActionListener(this);
        this.bRoot.addActionListener(this);

        // Set up the main application window --------------------------------

        /*
         * Make sure the main window is appropriately sized, exits this program
         * on close, and becomes visible to the user
         */

        this.pack();
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);

    }

    @Override
    public void actionPerformed(ActionEvent event) {
        JOptionPane.showMessageDialog(this,
                "You pressed: " + event.getActionCommand());
    }

    /**
     * Main method.
     *
     * @param args
     *             the command line arguments; unused here
     */
    public static void main(String[] args) {
        NNCalcViewLab view = new NNCalcViewLab();
    }

}
```