

```
// structure/presentation-model/CalcViewController.java
// Fred Swartz -- December 2004
// Paolo Bucci -- May 2011 (modified)

// GUI Organization - GUI independent of model.
//
// GUI subclasses JFrame and builds it in the constructor.
//
// The GUI creates a calculator model object, but knows nothing
// about the internal implementation of the calculator.

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class CalcViewController extends JFrame {
    //===== Constants
    private static final String INITIAL_VALUE = "1";

    //===== instance vars
    //... The Model.
    private CalcModel m_logic;

    //... Components
    private JTextField m_userInputTf = new JTextField(5);
    private JTextField m_totalTf     = new JTextField(20);
    private JButton    m_multiplyBtn = new JButton("Multiply");
    private JButton    m_clearBtn    = new JButton("Clear");

    //===== constructor
    /** Constructor */
    public CalcViewController(CalcModel model) {
        //... Set up the logic
        m_logic = model;
        m_logic.setValue(INITIAL_VALUE);

        //... Initialize components
        m_totalTf.setText(m_logic.getValue());
        m_totalTf.setEditable(false);

        //... Layout the components.
        JPanel content = new JPanel();
        content.setLayout(new FlowLayout());
        content.add(new JLabel("Input"));
        content.add(m_userInputTf);
        content.add(m_multiplyBtn);
        content.add(new JLabel("Total"));
        content.add(m_totalTf);
        content.add(m_clearBtn);

        //... Add button listeners.
        m_multiplyBtn.addActionListener(new MultiplyListener());
        m_clearBtn.addActionListener(new ClearListener());

        //... finalize layout and set window parameters.
        this.setContentPane(content);
        this.pack();
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setTitle("Simple Calc - Presentation-Model");
    } //end constructor

    ////////////////////////////////////////////////// inner class MultiplyListener
    /** When a multiplication is requested.
     * 1. Get the user input number.
     */
}
```

```
* 2. Call the model to multiply by this number.
* 3. Get the result from the Model.
* 4. Set the Total textfield to this result.
* If there was an error, display it in a JOptionPane.
*/
private class MultiplyListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        String userInput = "";
        try {
            userInput = m_userInputTf.getText();
            m_logic.multiplyBy(userInput);
            m_totalTf.setText(m_logic.getValue());
        } catch (NumberFormatException nfex) {
            JOptionPane.showMessageDialog(CalcViewController.this,
                "Bad input: '" + userInput + "'");
        }
    }
} //end inner class MultiplyListener

////////////////////////////////////// inner class ClearListener
/** 1. Reset model.
 * 2. Put model's value into Total textfield.
 */
private class ClearListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        m_logic.reset();
        m_totalTf.setText(m_logic.getValue());
    }
} //end inner class ClearListener
}
```