```java
/*-------------------------------------------------------------------------------------------------
 * Author:      Dan Cassidy
 * Date:        2015-07-20
 * Assignment:  HW4-3
 * Source File: BlankTrimmerView.java
 * Language:    Java
 * Course:      CSCI-C 490, Android Programming, MoWe 08:00
-------------------------------------------------------------------------------------------------*/
import java.awt.*;
import java.awt.event.*;

import javax.swing.*;

/**
 * View for the Blank Trimmer program.
 *
 * @author Dan Cassidy
 */
@SuppressWarnings("serial")
public class BlankTrimmerView extends JFrame
{
    private JTextField filePath = new JTextField();
    private JButton trimButton = new JButton("Trim");

    /**
     * No-parameter constructor. Sets up the frame for display.
     */
    public BlankTrimmerView()
    {
        super("Blank Trimmer");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setResizable(false);
        setLayout(new FlowLayout());

        // Component 1.
        add(new JLabel("Please choose a file:"));

        // Component 2.
        filePath.setColumns(50);
        add(filePath);

        // Component 3.
        JButton browseButton = new JButton("Browse...");
        browseButton.addActionListener(new ActionListener()
        {
            // Use the file chooser dialog to get a file name.
            public void actionPerformed(ActionEvent event)
            {
                final JFileChooser fileChooser = new JFileChooser();
                int result = fileChooser.showOpenDialog(BlankTrimmerView.this);
                if (result == JFileChooser.APPROVE_OPTION)
                    filePath.setText(fileChooser.getSelectedFile().getAbsolutePath());
            }
        });
        add(browseButton);

        // Component 4.
        JSeparator separator = new JSeparator(SwingConstants.VERTICAL);
        separator.setPreferredSize(new Dimension(1, 20));
        add(separator);

        // Component 5.
        add(trimButton);

        // Force the frame resize itself and make it appear where the host system dictates.
        pack();
        setLocationByPlatform(true);
    }

    /**
     * Add an action listener to the trim button.
     *
     * @param listener The ActionListener to add.
     */
    public void addTrimButtonActionListener(ActionListener listener)
    {
        trimButton.addActionListener(listener);
    }

    /**
     * Interface method to get the path of the file to work with.
     *
     * @return String containing the path of the file.
     */
    public String getFilePath()
    {
        return filePath.getText();
    }

    /**
     * Show a modal dialog displaying the given message.
     *
     * @param message The message to display in the dialog.
     * @param messageType JOptionPane constant determining the look and feel of the dialog.
     */
    public void showMessage(String message, int messageType)
    {
        if (message == null)
            message = "No message.";

        String title = "Message";

        // Only expecting error messages or informational messages, so only checking for those two
        // cases.
        switch (messageType)
        {
            case JOptionPane.ERROR_MESSAGE:
                title = "Error";
                break;

            case JOptionPane.INFORMATION_MESSAGE:
                title = "Information";
                break;
        }

        JOptionPane.showMessageDialog(this, message, title, messageType);
    }
}

```