```java
/**
settingsWindow.java
This file is used to create/open the window in which the user can edit the settings.
**/

import org.eclipse.swt.widgets.Display;
import org.eclipse.swt.widgets.Shell;
import org.eclipse.swt.widgets.Button;
import org.eclipse.swt.widgets.ColorDialog;

import java.awt.FileDialog;
import java.io.*;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

import org.eclipse.core.runtime.Path;
import org.eclipse.swt.SWT;
import org.eclipse.swt.layout.GridLayout;
import org.eclipse.swt.widgets.Text;
import org.eclipse.swt.widgets.Label;
import org.eclipse.swt.layout.GridData;
import org.eclipse.swt.widgets.Scale;
import org.eclipse.swt.events.SelectionAdapter;
import org.eclipse.swt.events.SelectionEvent;
import org.eclipse.swt.graphics.Color;
import org.eclipse.swt.events.MouseAdapter;
import org.eclipse.swt.events.MouseEvent;

public class settingsWindow {

    public Settings currentSettings = new Settings();
    protected Shell shlSettings;
    private Text textSaveName;

    /**
     * Launch the application.
     * @param args
     */
    public static void main(String[] args) {
        try {
            settingsWindow window = new settingsWindow();
            Settings dummySettings = new Settings();
            window.open(dummySettings);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * Open the window.
     */
    public void open(Settings mainWindowSettings) {
        Display display = Display.getDefault();
        Shell shlSettings = new Shell();
        shlSettings.setSize(758, 272);
        shlSettings.setText("Settings");
        shlSettings.setLayout(new GridLayout(3, false));

        currentSettings = mainWindowSettings;

        Button btnShowAxes = new Button(shlSettings, SWT.CHECK);
        btnShowAxes.setSelection(currentSettings.showAxes);
        btnShowAxes.addSelectionListener(new SelectionAdapter() {
            @Override
            public void widgetSelected(SelectionEvent e) {
                currentSettings.showAxes = btnShowAxes.getSelection();
            }
        });
        btnShowAxes.setText("Show axes");

        Label lblInequalityTransparency = new Label(shlSettings, SWT.NONE);
        lblInequalityTransparency.setLayoutData(new GridData(SWT.RIGHT, SWT.CENTER, false, false, 1, 1));
        lblInequalityTransparency.setText("Inequality Transparency");

        Scale scaleTransparency = new Scale(shlSettings, SWT.NONE);
        scaleTransparency.setEnabled(false);
        scaleTransparency.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseUp(MouseEvent e) {
                currentSettings.inequalityTransparency = scaleTransparency.getIncrement()/100.0;
            }
        });
        scaleTransparency.setLayoutData(new GridData(SWT.LEFT, SWT.CENTER, true, false, 1, 1));

        Button btnShowGridlines = new Button(shlSettings, SWT.CHECK);
        btnShowGridlines.setEnabled(false);
        btnShowGridlines.setSelection(currentSettings.showGridlines);
        btnShowGridlines.addSelectionListener(new SelectionAdapter() {
            @Override
            public void widgetSelected(SelectionEvent e) {
                currentSettings.showGridlines = btnShowGridlines.getSelection();
            }
        });
        btnShowGridlines.setText("Show gridlines");

        Button btnPickBackgroundColour = new Button(shlSettings, SWT.NONE);
        btnPickBackgroundColour.setLayoutData(new GridData(SWT.FILL, SWT.CENTER, false, false, 1, 1));
        btnPickBackgroundColour.addSelectionListener(new SelectionAdapter() {
            @Override
            public void widgetSelected(SelectionEvent e) {
                ColorDialog newColourPicker = new ColorDialog(shlSettings);
                newColourPicker.open();
                currentSettings.backgroundColor = new Color(Display.getCurrent(), newColourPicker.getRGB());
            }
        });
        btnPickBackgroundColour.setText("Pick Background Colour");

        Button btnPickPenColour = new Button(shlSettings, SWT.NONE);
        btnPickPenColour.addSelectionListener(new SelectionAdapter() {
            @Override
            public void widgetSelected(SelectionEvent e) {
                ColorDialog newColourPicker = new ColorDialog(shlSettings);
                newColourPicker.open();
                currentSettings.penColor = new Color(Display.getCurrent(), newColourPicker.getRGB());
            }
        });
        btnPickPenColour.setText("Pick Pen Colour");

        Button btnLabelAxes = new Button(shlSettings, SWT.CHECK);
        btnLabelAxes.setEnabled(false);
        btnLabelAxes.setGrayed(true);
        btnLabelAxes.addSelectionListener(new SelectionAdapter() {
            @Override
            public void widgetSelected(SelectionEvent e) {
                currentSettings.labelAxes = btnLabelAxes.getSelection();
            }
        });
        btnLabelAxes.setText("Label axes");

        Button btnOpenFile = new Button(shlSettings, SWT.NONE);
        btnOpenFile.addSelectionListener(new SelectionAdapter() {
            @Override
            public void widgetSelected(SelectionEvent e) {
                try {
                    org.eclipse.swt.widgets.FileDialog newFilePicker = new org.eclipse.swt.widgets.FileDialog(shlSettings, FileDialog.LOAD);
                    newFilePicker.open();
                    newFilePicker.setFilterExtensions(new String[] { "*.gwf", "*.*" });

                    String fileName = newFilePicker.getFilterPath() + "\\" + newFilePicker.getFileName();
                    //Reads the contents of the file at the given location into a string.
                    //Note that this does not need to be paired with a 'close' statement.
                    String fileContents = Files.readString(Paths.get(fileName));

                    currentSettings.currentEquations += "\n" + fileContents;
                } catch (IOException ioException) {
                    ioException.printStackTrace();
                }
            }
        });
        btnOpenFile.setLayoutData(new GridData(SWT.FILL, SWT.CENTER, false, false, 1, 1));
        btnOpenFile.setText("Open file");

        Button btnSaveToFile = new Button(shlSettings, SWT.NONE);
        btnSaveToFile.addSelectionListener(new SelectionAdapter() {
            @Override
            public void widgetSelected(SelectionEvent e) {
                org.eclipse.swt.widgets.MessageBox saveWarningDialog = new org.eclipse.swt.widgets.MessageBox(shlSettings, SWT.OK | SWT.CANCEL);
                saveWarningDialog.setText("Save/Overwrite");
                saveWarningDialog.setMessage("Warning: This will overwrite any file named " + textSaveName.getText() + " in the specified folder.");
                saveWarningDialog.open();
                try {
                    org.eclipse.swt.widgets.DirectoryDialog newFolderPicker = new org.eclipse.swt.widgets.DirectoryDialog(shlSettings, SWT.SAVE);
                    File location = Path.fromOSString(newFolderPicker.open()).makeAbsolute().toFile();
                    String fileName = location.toString() + "\\" + textSaveName.getText();
                    if (currentSettings.debugMode) {
                        System.out.println("Writing to file "+fileName);
                        System.out.println("Equations to be written: "+currentSettings.currentEquations);
                    }
                    PrintWriter printWriter = new PrintWriter(fileName);
                    printWriter.println(currentSettings.currentEquations);
                    printWriter.close();
                } catch (Exception exceptionInstance) {
                    exceptionInstance.printStackTrace();
                }
            }
        });
        btnSaveToFile.setText("Pick folder and save/overwrite");

        Button btnLabelIntercepts = new Button(shlSettings, SWT.CHECK);
        btnLabelIntercepts.setEnabled(false);
        btnLabelIntercepts.setGrayed(true);
        btnLabelIntercepts.addSelectionListener(new SelectionAdapter() {
            @Override
            public void widgetSelected(SelectionEvent e) {
                currentSettings.labelIntersections = btnLabelIntercepts.getSelection();
            }
        });
        btnLabelIntercepts.setText("Label intercepts with axes");

        Label lblFileName = new Label(shlSettings, SWT.NONE);
        lblFileName.setLayoutData(new GridData(SWT.RIGHT, SWT.CENTER, false, false, 1, 1));
        lblFileName.setText("File name:");

        textSaveName = new Text(shlSettings, SWT.BORDER);
        textSaveName.setText("Graph.gwf");
        GridData gd_textSaveName = new GridData(SWT.LEFT, SWT.CENTER, true, false, 1, 1);
        gd_textSaveName.widthHint = 143;
        textSaveName.setLayoutData(gd_textSaveName);

        Button btnLabelIntersections = new Button(shlSettings, SWT.CHECK);
        btnLabelIntersections.setEnabled(false);
        btnLabelIntersections.setGrayed(true);
        btnLabelIntersections.addSelectionListener(new SelectionAdapter() {
            @Override
            public void widgetSelected(SelectionEvent e) {
                currentSettings.labelIntersections = btnLabelIntersections.getSelection();
            }
        });
        btnLabelIntersections.setText("Label intersections on curves");
        new Label(shlSettings, SWT.NONE);
        new Label(shlSettings, SWT.NONE);

        shlSettings.open();
        shlSettings.layout();
        while (!shlSettings.isDisposed()) {
            if (!display.readAndDispatch()) {
                display.sleep();
            }
        }
    }
}
```