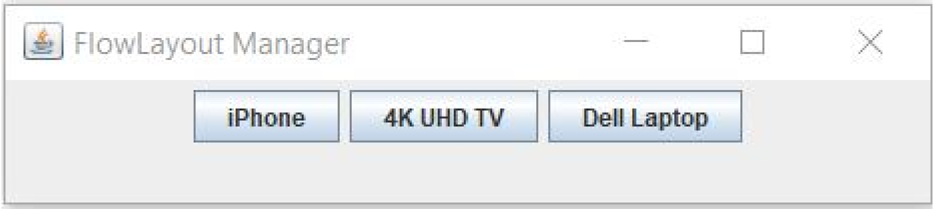
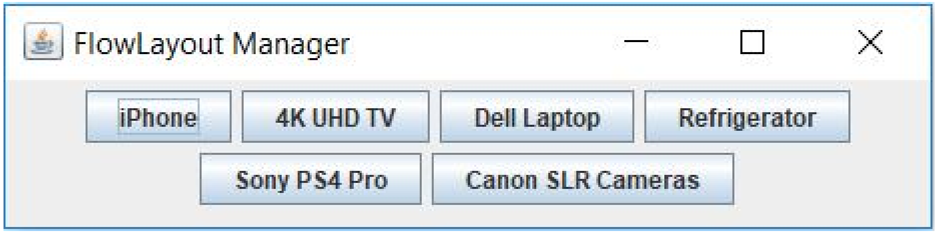
FlowLayout

Overview

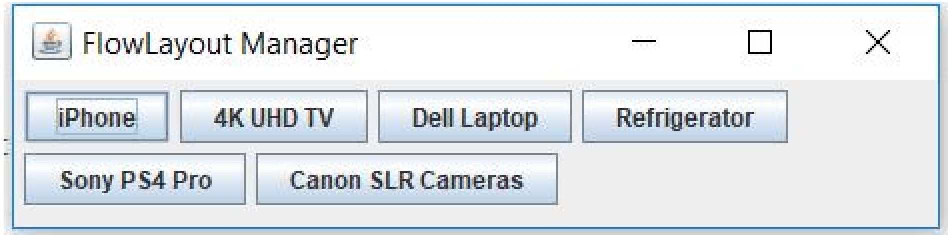
FlowLayout is the simplest layout manager for a container as the UI components will flow "naturally" onto its container from left to right and top to bottom. Basically FlowLayout arranges UI components horizontally in the order of component insertions until no more components fit on the current row. The FlowLayout manager will leave each UI component at its preferred size. By **default** the components are placed in the **center of the panel (all components will be center aligned) as seen below:**



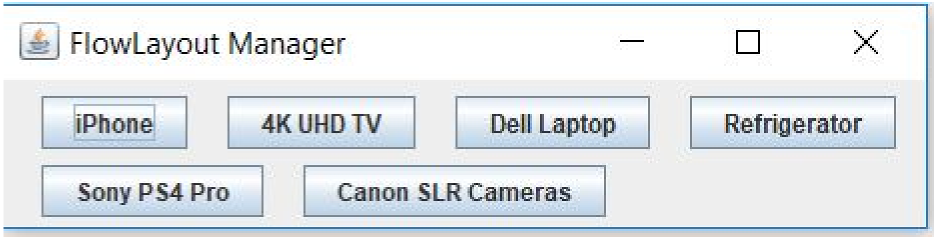
If there is no more room left on the current row (limited by panel's size) to fit the next component the remaining components will be moved to the next row.



Users have the option to set the components left or right justified however.



Additionally it's not uncommon that users might want to leave some margins or **gaps** between the components and the container as well as between the components based on their own setup instead of taking the default margins provided by the FlowLayout. We can specify those gaps (horizontal gaps hgap and vertical gaps vgap) in pixels. Below is the screenshot of the same GUI only that we set the vertical gap (vgap) to 8 pixels and horizontal gaps (hgap) to 20 pixels.



FlowLayout is the default layout for JPanel. When placing UI components on a JPanel if you don't explicitly specify a layout manager for it the FlowLayout manager will take over the responsibility of arranging the components (note that each JPanel can have different layout managers if desired). Note that if you wish to use FlowLayout on any container (JFrame for example) you can always set it as the container layout of your choice.

Configure FlowLayout to a JPanel (or any container)

We must instantiate a new FlowLayout object with the desired configuration (left justified for example) then assign it to the desired panel or frame.

JPanel paymentPanel = new JPanel () ;

// now configure or assign a FlowLayout of your choice to the panel

FlowLayout layout = new FlowLayout (FlowLayout.LEFT); // instantiating FlowLayout object which is left aligned.

paymentPanel.setLayout (layout);

The alignment properties are:

- LEFT
- RIGHT
- CENTER
- LEADING
- TRAILING

In case we want to specify the gaps we will use a different FlowLayout constructor:

FlowLayout layout = new FlowLayout (FlowLayout.LEFT, 5, 20);   // left alignment, 5 pixels horizontally and 20 pixels vertically.

Modify existing FlowLayout

You can get a layout manager from a JPanel and reconfigure it according to your needs.

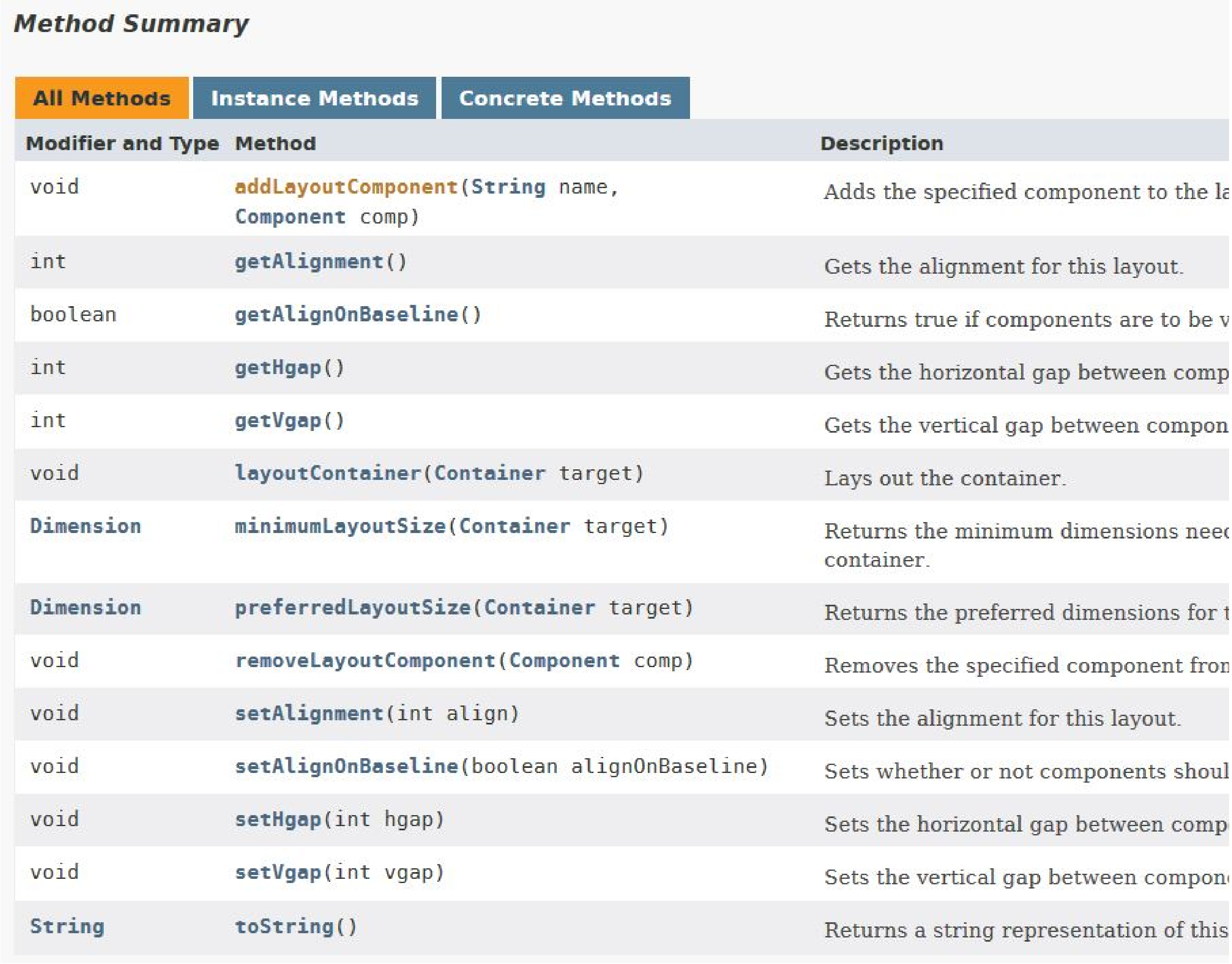
```
JPanel  paymentPanel  = new JPanel ();

FlowLayout layout =  (FlowLayout) paymentPanel,getLayout ():

layout.setAlignment (FlowLayout.RIGHT);
```

FlowLayout's methods

Basically additional setup or configuration you can do with a FlowLayout manager is to set components' alignment and horizontal and vertical gaps between components and between components and their containers. Below is the list of available FlowLayout methods from **Java docs (https://docs.oracle.com/javase/10/docs/api/java/awt/FlowLayout.html)** .

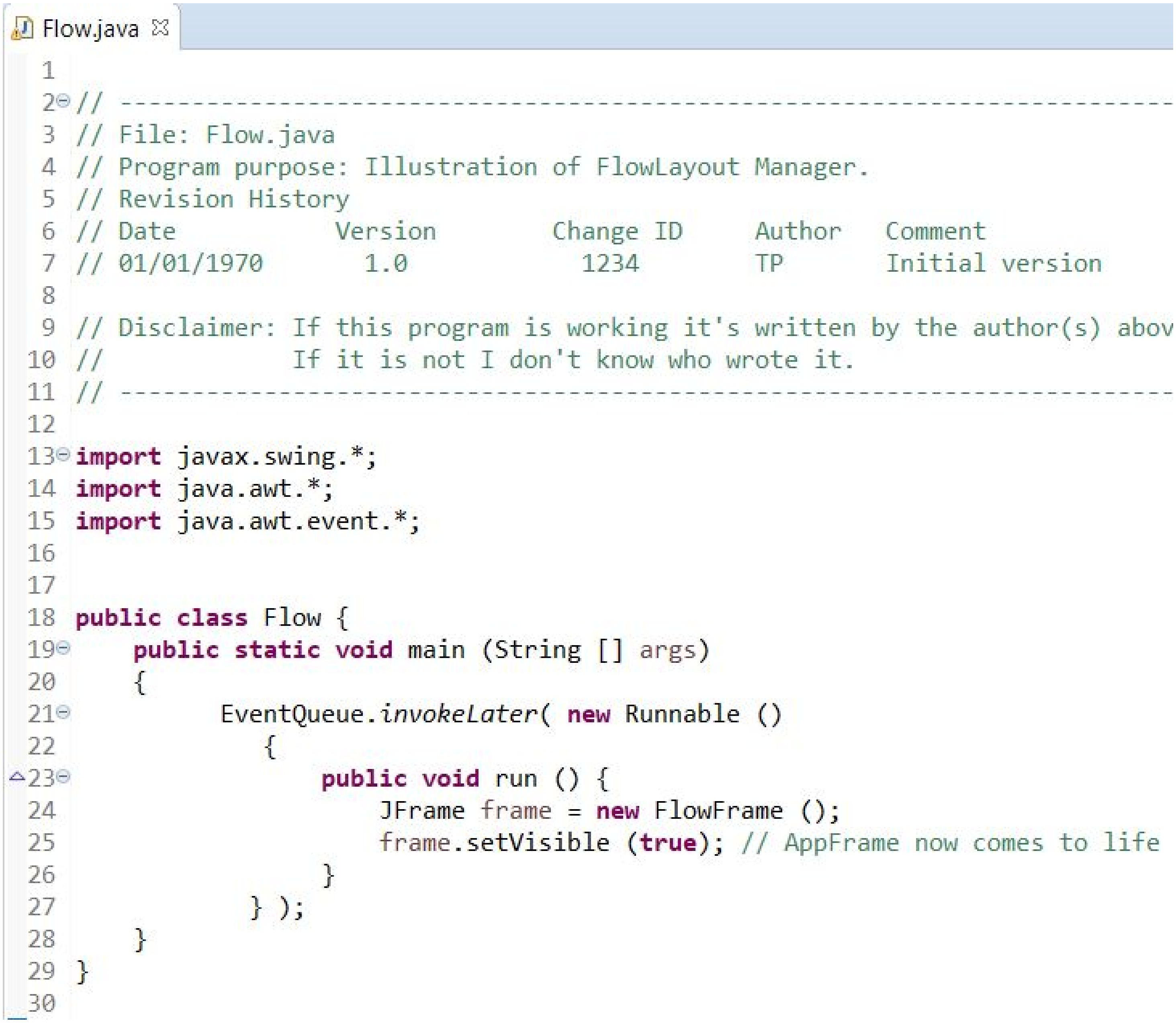


Method Summary

Modifier and Type	Method	Description
void	addLayoutComponent(String name, Component comp)	Adds the specified component to the la
int	getAlignment()	Gets the alignment for this layout.
boolean	getAlignOnBaseline()	Returns true if components are to be v
int	getHgap()	Gets the horizontal gap between comp
int	getVgap()	Gets the vertical gap between compon
void	layoutContainer(Container target)	Lays out the container.
Dimension	minimumLayoutSize(Container target)	Returns the minimum dimensions neec container.
Dimension	preferredLayoutSize(Container target)	Returns the preferred dimensions for t
void	removeLayoutComponent(Component comp)	Removes the specified component fron
void	setAlignment(int align)	Sets the alignment for this layout.
void	setAlignOnBaseline(boolean alignOnBaseline)	Sets whether or not components shoul
void	setHgap(int hgap)	Sets the horizontal gap between comp
void	setVgap(int vgap)	Sets the vertical gap between compon
String	toString()	Returns a string representation of this

The Sample Program

You can manually type the sample program below to your IDE and play around with it by adding different number of UI components (JButton controls) and setting different alignment and hgap, vgap. No program copy/paste is available I apologize. Have fun!

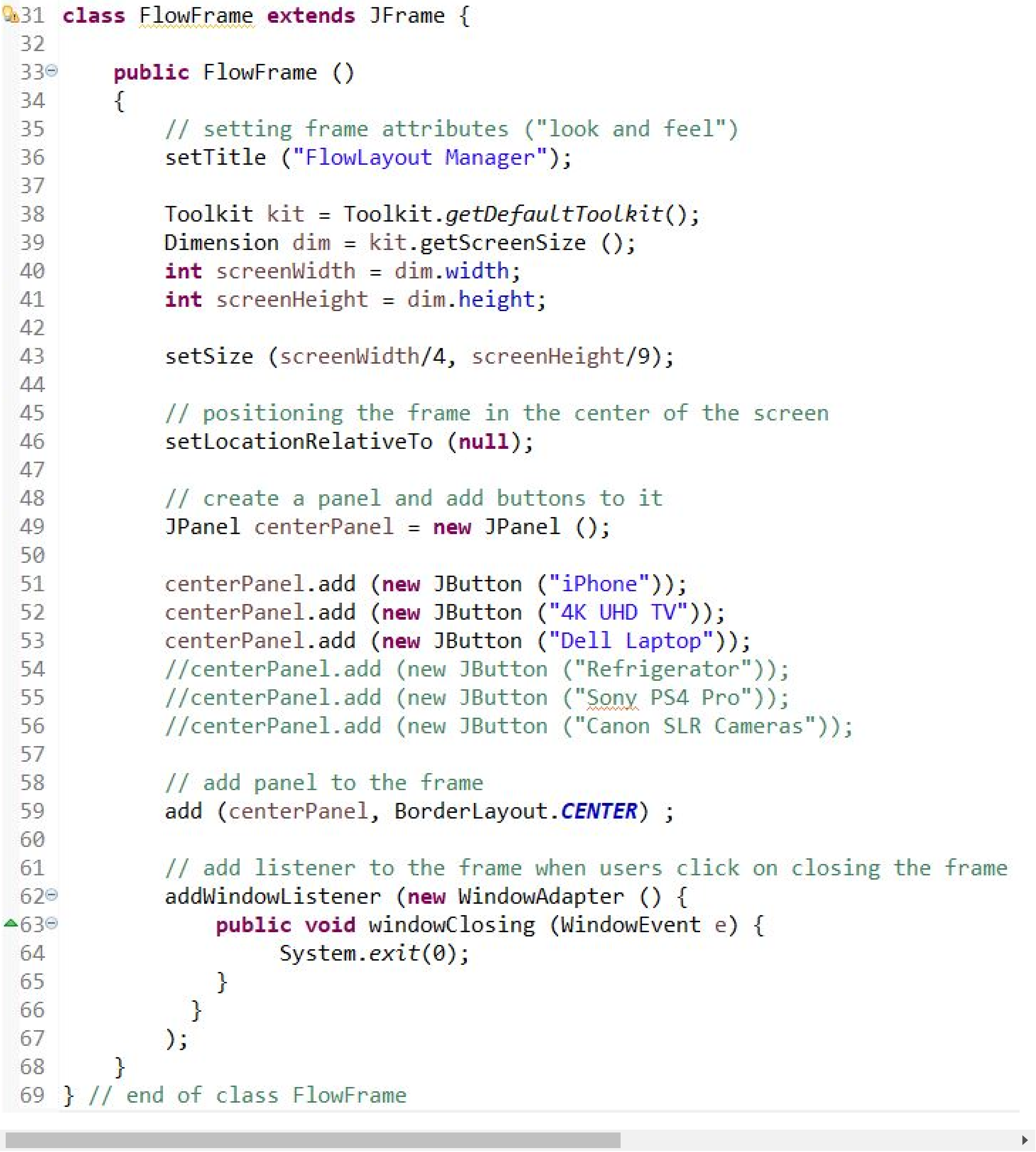


```
// -------------------------------------------------------------------------
// File: Flow.java
// Program purpose: Illustration of FlowLayout Manager.
// Revision History
// Date              Version        Change ID     Author   Comment
// 01/01/1970         1.0            1234         TP       Initial version

// Disclaimer: If this program is working it's written by the author(s) abov
//             If it is not I don't know who wrote it.
// -------------------------------------------------------------------------

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


public class Flow {
    public static void main (String [] args)
    {
          EventQueue.invokeLater( new Runnable ()
             {
                 public void run () {
                     JFrame frame = new FlowFrame ();
                     frame.setVisible (true); // AppFrame now comes to life
                 }
            } );
    }
}
```



```java
class FlowFrame extends JFrame {

    public FlowFrame ()
    {
        // setting frame attributes ("look and feel")
        setTitle ("FlowLayout Manager");

        Toolkit kit = Toolkit.getDefaultToolkit();
        Dimension dim = kit.getScreenSize ();
        int screenWidth = dim.width;
        int screenHeight = dim.height;

        setSize (screenWidth/4, screenHeight/9);

        // positioning the frame in the center of the screen
        setLocationRelativeTo (null);

        // create a panel and add buttons to it
        JPanel centerPanel = new JPanel ();

        centerPanel.add (new JButton ("iPhone"));
        centerPanel.add (new JButton ("4K UHD TV"));
        centerPanel.add (new JButton ("Dell Laptop"));
        //centerPanel.add (new JButton ("Refrigerator"));
        //centerPanel.add (new JButton ("Sony PS4 Pro"));
        //centerPanel.add (new JButton ("Canon SLR Cameras"));

        // add panel to the frame
        add (centerPanel, BorderLayout.CENTER) ;

        // add listener to the frame when users click on closing the frame
        addWindowListener (new WindowAdapter () {
            public void windowClosing (WindowEvent e) {
                 System.exit(0);
            }
          }
        );
    }
} // end of class FlowFrame
```