BorderLayout

The JFrame's default layout manager

Here are what BorderLayout can offer to your containers:

- Default layout manager for JFrame but you may set it to be the layout manager for JPanels if needed
- Components are arranged in five regions : North/South/Center/East/West or three rows: North/West-Center-East/South
- Center region always tries to be biggest
- Not all regions need to be occupied. The component may grow/shrink to fill the available space
- BorderLayout.CENTER is the default region if not specified
- Each region can contain only one component. Therefore if multiple components to be added to the same region only the last component will be shown.

To set a layout manager for your container do the following:

1. Setting FlowLayout manager for JFrame (whose default layout is BorderLayout)

```
class EmptyFrame extends JFrame  {
        public EmptyFrame ( ) {
                setLayout (  new FlowLayout  ( ) ) ;  // now the components can be added and centered
                // OR setLayout (  new FlowLayout  (FlowLayout.LEFT ) ) ;  // left-justified
        }
}
```

2. Setting BorderLayout manager for JPanel (whose default layout is FlowLayout)

```
class  InformationPanel extends  JPanel )  {
        public  InformationPanel ( ) {
                setLayout  ( new BorderLayout ( ) ) ;  // now the components can be added in 5 regions
         }
```

```
}
```

3. To modify an existing layout for example from Center layout to Left layout for a JPanel

```
FlowLayout layout = (FlowLayout) infoPanel.getLayout ( );

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

The best way to understand those layout managers is to try out the code in different ways to see how they lay out the components. Once you know how they do it it becomes much easier for you to place the components in the position you'd like.

Below is a program to add 5 JButtons to 5 different regions of a Frame. As you will see the Center region grows to occupy most of the Frame. try to click on the region to experience how much space each region occupies.

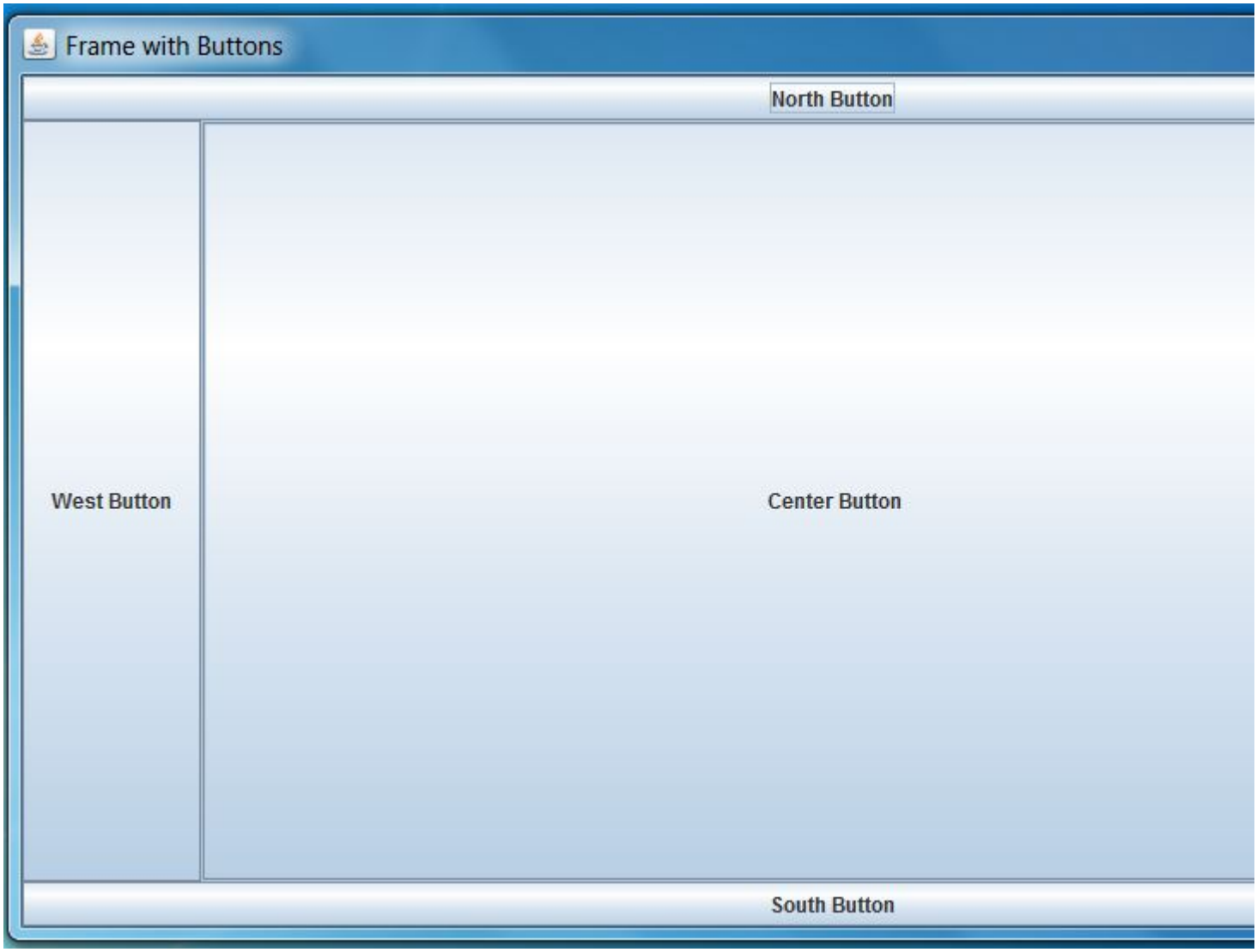
Program Sample

Only the bold-face code is of interest. The rest of the code is the same as the EmptyFrame program.

```
/**
   @version 1.0 2017-10-22
   @author  TP - FH Computer Science Department
 */

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

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

```
class EmptyFrame extends JFrame {

   // Frame size might be defined here
   // private final int WIDTH = 200 ;
   // private final int HEIGHT = 200 ;

   public EmptyFrame ()
   {
      // setting frame attributes ("look and feel")
      setTitle ("Frame with Buttons");

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

      //System.out.println ("W ="+ screenWidth +",  H=" +screenHeight );
      setSize (screenWidth/2, screenHeight/2);

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

      // adding 5 buttons to the Frame
      add (new JButton ("North Button"), BorderLayout.NORTH) ;
      add (new JButton ("South Button"), BorderLayout.SOUTH) ;
      add (new JButton ("Center Button"), BorderLayout.CENTER) ;
      add (new JButton ("West Button"), BorderLayout.WEST) ;
      add (new JButton ("East Button"), BorderLayout.EAST) ;

      // setting frame behavior
      //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      addWindowListener (new WindowAdapter ()
       {
          public void windowClosing (WindowEvent e) {
              System.exit(0);
          }
       }
      );
```

```
    }
} // end of class EmptyFrame
```



Later we will use the very same program as above but instead of add 5 buttons we will be:

- Adding 5 JPanels to the Frame to the five regions
- In each Panel we will be adding 3 JButtons (will be centered in the JPanel by default) to the North and South regions
- Other variations will be to modify the JPanel centered FLowLayout to LEFT OR RIGHT alignment