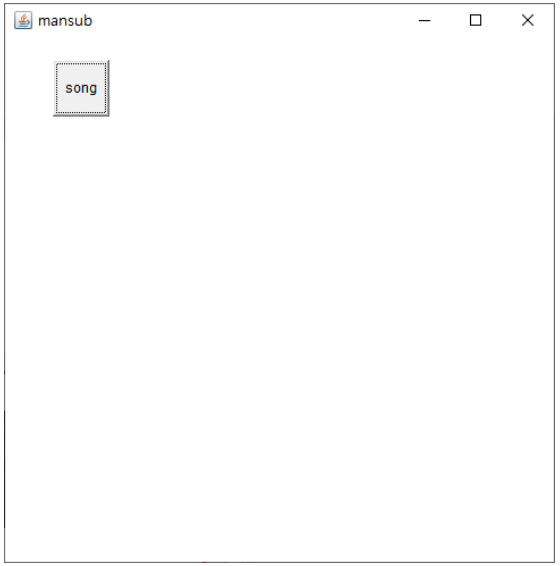
Lab 7

버튼 크기 및 위치 조절

```java
public class ButtonEx {

    public static void main(String[] args) {

        Frame f=new Frame( title: "mansub");
        Button btn=new Button( label: "song");
        f.setBounds( x: 100, y: 100, width: 500, height: 500);
        f.add(btn);

        btn.setBounds( x: 50, y: 50, width: 50, height: 50);
        f.setLayout(null);

        f.setVisible(true);

    }
}
```

setLayout(null)
or
setLayout(FlowLayout obj)

setBounds();



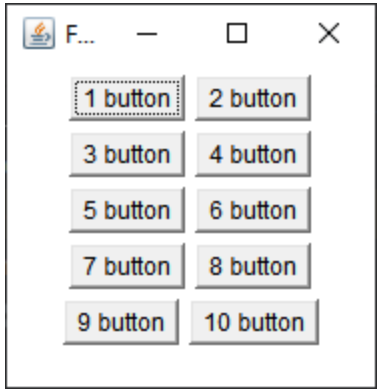
Layout

- 레이아웃에 따라 컴포넌트들을 배치하기 위한 객체

- **setLayout()을 이용하고 종류로는**
  - **FlowLayout**
  - **BorderLayout**
  - **GridLayout**
  - **등이 있다.**

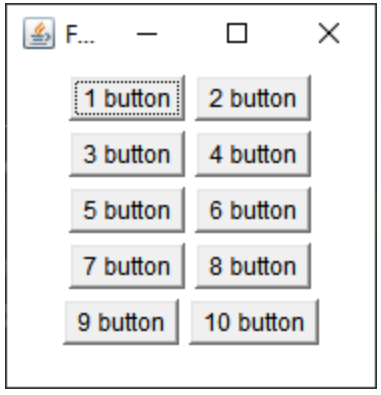
FlowLayout

- **panel**의 기본으로 컴포넌트 들을 수평으로 순서대로 나열하는 레이아웃
  - 수평으로 배치하다가 더 이상 공간이 없으면 다음 줄로 이동해서 배치



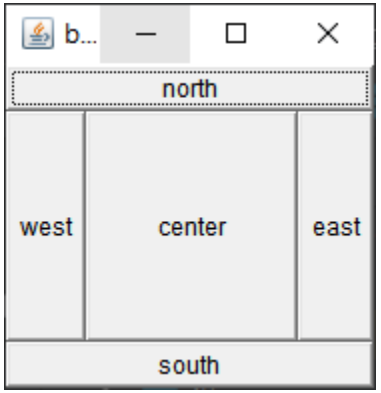
FlowLayout – example

```java
import java.awt.Button;
import java.awt.FlowLayout;
import java.awt.Frame;

public class Flow1 extends Frame
{
    FlowLayout f = new FlowLayout();
    Button btn[] = new Button[10];

    public Flow1(String str)
    {
        super(str);
        // FlowLayout
        setLayout(f);
        // 버튼 10개 생성
        for( int i = 0; i < 10; i++)
        {
            btn[i] = new Button( label: (i+1)+" button");
            add(btn[i]);
        }
        //width 값 바꾸면 버튼이 옆으로 늘어남
        setBounds( x: 100, y: 100, width: 200, height: 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Flow1( str: "Flow Test");
    }
}
```



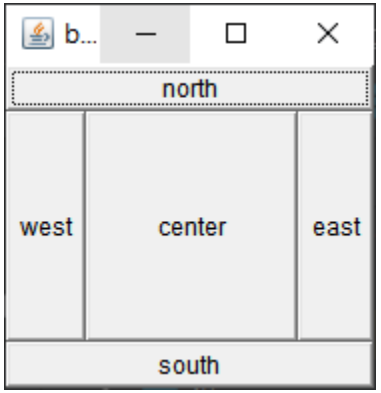
BorderLayout

- 5개의 영역으로 구분하여 보여줌
- 프레임은 기본적으로 BorderLayout



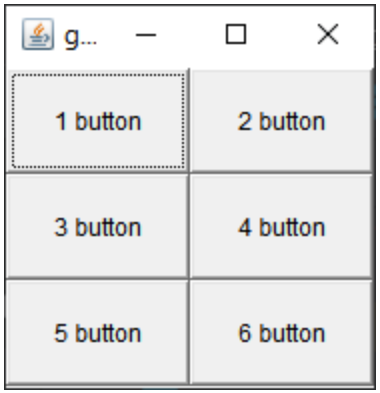
BoarderLayout – example

```java
import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.Frame;

public class Boarder1 extends Frame
{
    public Boarder1(String str)
    {
        super(str);
        // BorderLayout 생성
        setLayout(new BorderLayout());
        // 배치에 맞게
        add( name: "North", new Button( label: "north"));
        add( name: "West", new Button( label: "west"));
        add( name: "East", new Button( label: "east"));
        add( name: "Center", new Button( label: "center"));
        add( name: "South", new Button( label: "south"));

        setSize( width: 200, height: 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Boarder1( str: "boarder test");
    }
}
```



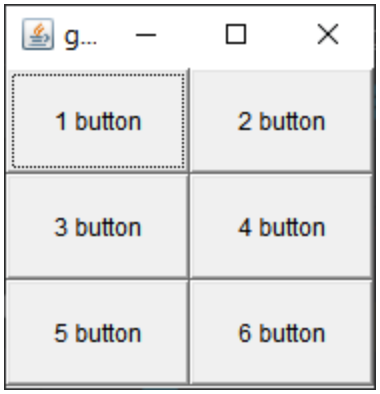
GridLayout

- 격자 모양으로 배치
- 행과 열의 수를 지정해서 배치
- 행과 열 수보다 컴포넌트의 수가 많으면 자동적으로 늘어남



GridLayout – example

```java
import java.awt.Button;
import java.awt.Frame;
import java.awt.GridLayout;

public class Grid1 extends Frame
{
    // 버튼 6개 선언
    Button btn[] = new Button[6];
    public Grid1(String str)
    {
        super(str);
        // GridLayout 생성(3 x 2)
        //row부터 인식하기 때문에 row값에 의해 형태가 결정난다
        //ex)row 1, col 2인 경우 => row 1 row 6형태로 됨
        setLayout(new GridLayout( rows: 6, cols: 1) );
        // 버튼 6개
        for(int i = 0; i <  6; i++)
        {
            btn[i] = new Button( label: i+1 + " button");
            add(btn[i]);
        }
        setSize( width: 200, height: 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Grid1( str: "gird test");
    }
}
```



AWT methods Example1

```
import java.awt.*;
public class LabelEx extends Frame {
    Label la1;
    Label la2;
    Label la3;
    Label la4;
    Font f1;
    Font f2;

    LabelEx() {
        super("Label Example");
        f1 = new Font("굴림체", Font.BOLD, 20);
        f2 = new Font("바탕체", Font.ITALIC, 25);
        la1 = new Label("RED");
        la2 = new Label("BLUE", Label.CENTER);
        la3 = new Label("GREEN", Label.RIGHT);
        la4 = new Label("PINK");
        setLayout(new GridLayout(4, 1));
        la1.setFont(f1);
        la2.setFont(f1);
        la3.setFont(f2);
        la4.setFont(f2);
        add(la1);
        add(la2);
        add(la3);
        add(la4);
        la1.setBackground(Color.red);
        la2.setBackground(Color.blue);
        la3.setBackground(Color.green);
        la4.setBackground(Color.pink);
        setSize(300, 200);
        setVisible(true);

    }

    public static void main(String[] ar) {
        LabelEx is = new LabelEx();
    }
}
```

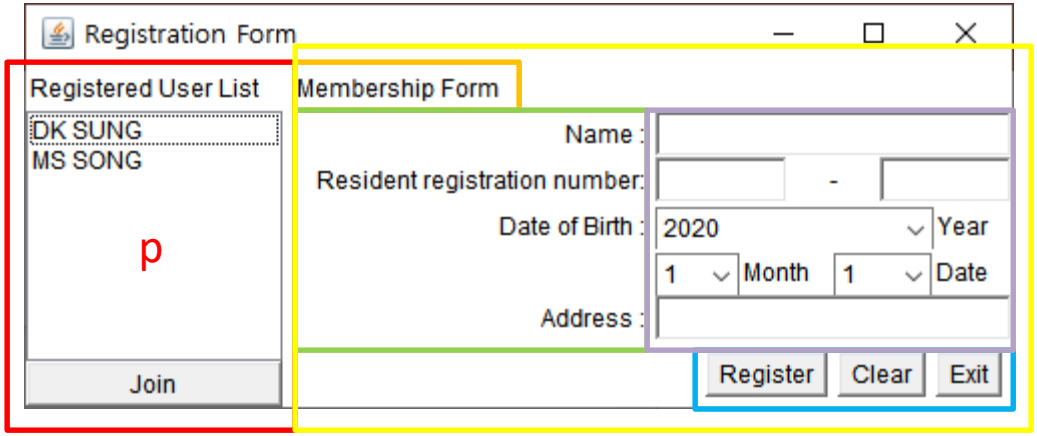
Result

AWT methods Example2

```
import java.awt.*;
import java.util.*;
import java.awt.List;

class Reg_Form extends Frame{
    private Label lb = new Label("Registered User List");
    private List li = new List(5, false);
    private Button bt = new Button("Join");

    private Label lb1 = new Label("Membership Form");
    private Label lb2 = new Label("Name : ", Label.RIGHT);
    private Label lb3 = new Label("Resident registration number: ", Label.RIGHT);
    private Label lb4 = new Label("Date of Birth : ", Label.RIGHT);
    private Label lb5 = new Label();
    private Label lb6 = new Label("Address : ", Label.RIGHT);

    private Button bt1 = new Button("Register");
    private Button bt2 = new Button("Clear");
    private Button bt3 = new Button("Exit");

    private TextField tf = new TextField();
    private TextField tf1 = new TextField(6);
    private Label lb7 = new Label("-", Label.CENTER);
    private TextField tf2 = new TextField(6);
    private Choice ch = new Choice();
    private Label lb8 = new Label("Year");
    private Choice ch1 = new Choice();
    private Label lb9 = new Label("Month");
    private Choice ch2 = new Choice();
    private Label lb10 = new Label("Date");
    private TextField tf3 = new TextField();

    public Reg_Form(String str){
        super(str);
        this.setLayout(new BorderLayout(5, 5));
        this.init();
        this.pack();
        this.setVisible(true);
    }
```

```
public void init(){
    //West
    Panel p = new Panel();
    p.setLayout(new BorderLayout());
    p.add("North", lb);
    p.add("Center", li);
    p.add("South", bt);
    this.add("West", p);
    //Center
    Panel p1 = new Panel();
    p1.setLayout(new BorderLayout());
    p1.add("North", lb1);
    Panel p2 = new Panel();
    p2.setLayout(new GridLayout(5, 1));
    p2.add(lb2);
    p2.add(lb3);
    p2.add(lb4);
    p2.add(lb5);
    p2.add(lb6);
    p1.add("West", p2);
    Panel p3 = new Panel();
    p3.setLayout(new FlowLayout(FlowLayout.RIGHT));
    p3.add(bt1);
    p3.add(bt2);
    p3.add(bt3);
    p1.add("South", p3);
    Panel p4 = new Panel();
    p4.setLayout(new GridLayout(5, 1));
    p4.add(tf);
    Panel p5 = new Panel();
    p5.setLayout(new BorderLayout());
    p5.add("West", tf1);
    p5.add("Center", lb7);
    p5.add("East", tf2);
    p4.add(p5);
        Panel p6 = new Panel();
        p6.setLayout(new BorderLayout());
        p6.add("Center", ch);
        p6.add("East", lb8);
        p4.add(p6);
        Panel p7 = new Panel();
        p7.setLayout(new GridLayout(1, 2));
        Panel p8 = new Panel();
        p8.setLayout(new BorderLayout());
        p8.add("Center", ch1);
        p8.add("East", lb9);
        p7.add(p8);
        Panel p9 = new Panel();
        p9.setLayout(new BorderLayout());
        p9.add("Center", ch2);
        p9.add("East", lb10);
        p7.add(p9);
        p4.add(p7);
        p4.add(tf3);
        p1.add("Center", p4);
        this.add("Center", p1);
        this.initData();
    }
    public void initData(){

        li.add("Heungmin Son");
        li.add("Jieun Lee");

        for(int i = 1; i <= 12; i++){
            ch1.add(i + "");
        }
        for(int i = 1; i <= 31; i++){
            ch2.add(i + "");
        }
        Calendar ca = Calendar.getInstance();
        int year = ca.get(Calendar.YEAR);
        for(int i = year; i >= 1900; i--){
            ch.add(i + "");
        }
    }
}

public class Example2{
    public static void main(String[] ar){
        Reg_Form es = new Reg_Form("Registration Form");
    }
}
```

Result


Registration Form
Registered User List
Membership Form
DK SUNG
MS SONG
p
Join
Name :
Resident registration number:
-
Date of Birth :
2020
Year
1
Month
1
Date
Address :
Register
Clear
Exit

AWT methods Example1_Modify (WindowListener & ActionListener)

```
import java.awt.*;
import java.awt.event.*;

import static java.lang.Thread.sleep;

public class LabelEx extends Frame {
    Label la1;
    Label la2;
    Label la3;
    Label la4;
    Font f1;
    Font f2;
    Button b1;
    int count = 0;

    class MyListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {

            if(e.getSource() == b1) {
                switch(count%4) {
                    case 0:
                        la1.setAlignment((la1.getAlignment()+1)%3);
                        break;
                    case 1:
                        la2.setAlignment((la2.getAlignment()+1)%3);
                        break;
                    case 2:
                        la3.setAlignment((la3.getAlignment()+1)%3);
                        break;
                    case 3:
                        la4.setAlignment((la4.getAlignment()+1)%3);
                        break;
                    default:
                        break;
                }
                b1.setLabel("Press_"+(++count));
            }
        }
    }
```

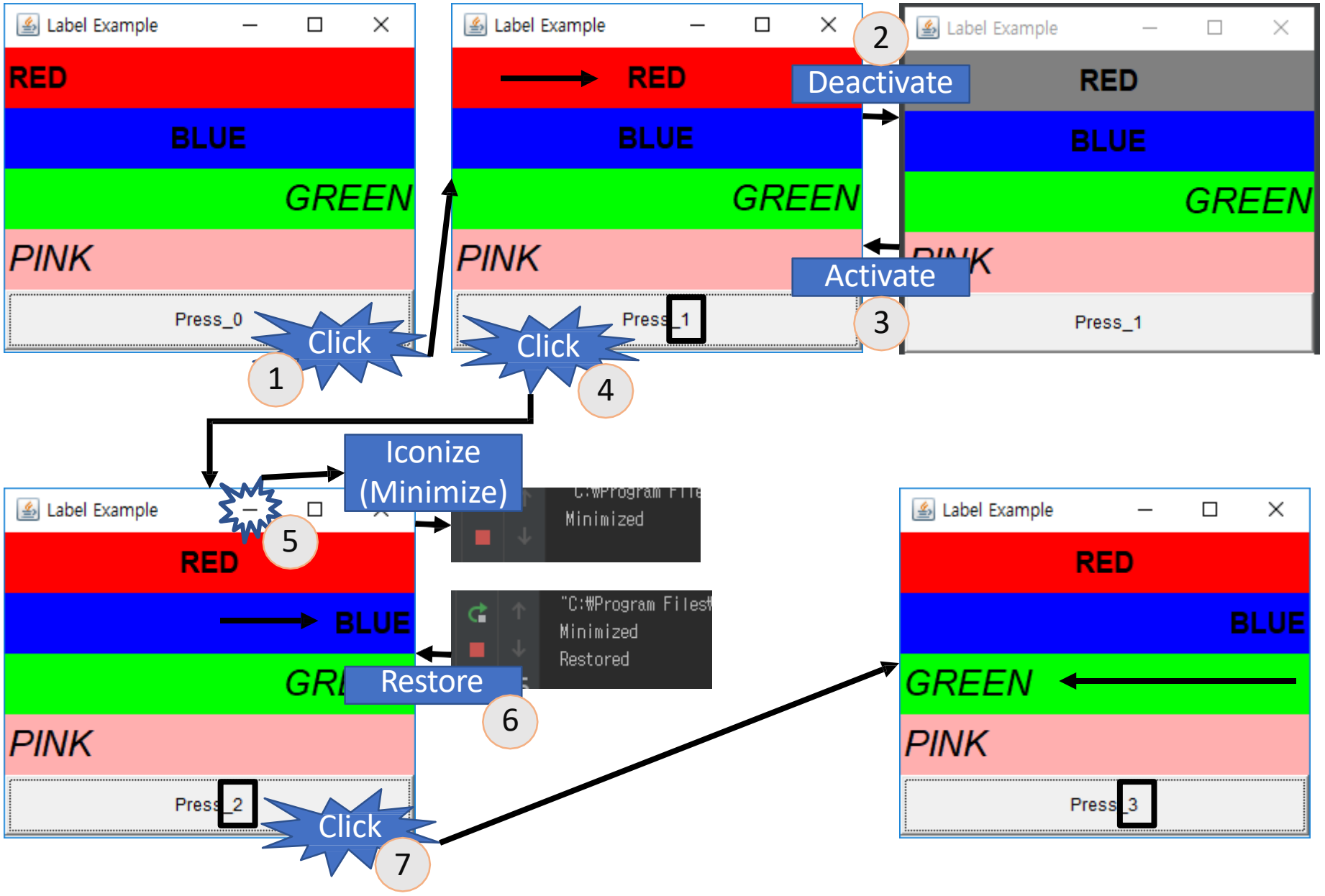
```
class WinEvent implements WindowListener{
    //윈도우가 활성화 될때
    public void windowActivated(WindowEvent e){
        la1.setBackground(Color.red);
    }
    //윈도우가 닫힌 다음에 호출
    public void windowClosed(WindowEvent e){}

    //윈도우가 닫히려고 할때 (X 버튼 누를때)
    public void windowClosing(WindowEvent e){
        System.exit(0);// X를 누르면 종료된다.
    }
    //윈도우가 비활성화 될때
    public void windowDeactivated(WindowEvent e){
        la1.setBackground(Color.GRAY);
    }
    //윈도우가 정상화 되었을때
    public void windowDeiconified(WindowEvent e){
        System.out.println("Restored");
    }
    //윈도우가 아이콘화 되었을때
    public void windowIconified(WindowEvent e){
        System.out.println("Minimized");
    }
    //윈도우가 화면에 처음 나타날때
    public void windowOpened(WindowEvent e){
        la4.setBackground(Color.red);
        try {
            sleep(350);
        } catch (InterruptedException e1) {
            e1.printStackTrace();
        }
        la4.setBackground(Color.pink);
        try {
            sleep(350);
        } catch (InterruptedException e1) {
            e1.printStackTrace();
        }
        la4.setBackground(Color.red);
        try {
            sleep(350);
        } catch (InterruptedException e1) {
            e1.printStackTrace();
        }
        la4.setBackground(Color.pink);
    }
```

AWT methods Example1_Modify (WindowListener & ActionListener)

```
    LabelEx() {
        super("Label Example");

        this.addWindowListener(new WinEvent());// 윈도우 이벤트 등록
        MyListener buttonlisten = new MyListener(); //for ButtonAction

        f1 = new Font("굴림체", Font.BOLD, 20);
        f2 = new Font("바탕체", Font.ITALIC, 25);
        la1 = new Label("RED");
        la2 = new Label("BLUE", Label.CENTER);
        la3 = new Label("GREEN", Label.RIGHT);
        la4 = new Label("PINK");
        b1 = new Button("Press_"+count);
        b1.addActionListener(buttonlisten);

        setLayout(new GridLayout(5, 1));
        la1.setFont(f1);
        la2.setFont(f1);
        la3.setFont(f2);
        la4.setFont(f2);
        add(la1);
        add(la2);
        add(la3);
        add(la4);
        add(b1);
        la1.setBackground(Color.red);
        la2.setBackground(Color.blue);
        la3.setBackground(Color.green);
        la4.setBackground(Color.pink);
        setSize(300, 250);
        setVisible(true);
    }

    public static void main(String[] args) {
        LabelEx is = new LabelEx();
    }
}
```

main에 LabelEx obj 2개 생성하면 windowDeactivated() & windowActivated() 함수가 어떻게 동작하는지 알 수 있음

Result


Label Example
RED
BLUE
GREEN
PINK
Press_0
Click
1
Press_1
2
Deactivate
3
Activate
4
Iconize
(Minimize)
5
Minimized
"C:₩Program Files
Restored
Restore
6
Press_2
Click
7
Press_3

```
if(e.getSource() == b1) {
    switch(count%4) {
        case 0:
            la1.setAlignment((la1.getAlignment()+1)%);
            break;
```

1. Click Button
   - Red goes to the center & Button number changed
2. Deactivate the window
   - Color red changes to gray
3. Reactivate the window
   - Color gray changes back to red
4. Click Button
   - Blue goes to the right & Button number changed
5. Iconize (Minimize) the window
   - System.out.println(“Minimized”)
6. Restore the window
   - System.out.println(“Restored”)
7. Click Button
   - Green goes to the left & Button number changed

.getAlignment()
Return value:
0 (left) 1 (center) 2 (right)