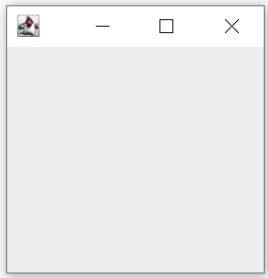


## GUI

1. (윈도우 크기, 위치) 다음 프로그램 소스를 입력하고 실행하시오.

- 실습 #1: 아래 코드에서 `setSize(400, 400);` 문장과 `setLocationRelativeTo(null);` 문장의 순서를 변경한 경우 윈도우가 어떻게 표시되는지 확인하시오. 힌트: 생성자 내에서 `System.out.println(getSize());`를 실행해 보시오.

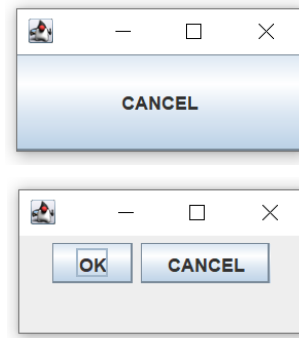


```
public class Test extends JFrame {  
    public Test() {  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setSize(400, 400); // 윈도우 가로, 세로 (width, height) 크기 (픽셀 단위) 설정  
        setLocationRelativeTo(null); // 윈도우 위치를 화면 중앙으로 설정  
    }  
    public static void main(String[] args) {  
        //Test test=new Test();  
        //test.setVisible(true);  
        new Test().setVisible(true); // 위 두 문장 실행과 동일  
    }  
}
```

2. (**JFrame,ContentPane, BorderLayout**) 다음 프로그램 소스를 입력하고 실행하시오. 아래 코드의 `getContentPane();`은 JFrame 객체의 ContentPane을 반환하는데, JFrame 객체의 ContentPane은 BorderLayout 설정된 JPanel 객체이다.

- 실습 #1: 아래 코드에서 `container.setLayout(new FlowLayout());` 문장의 주석을 해제한 후 실행하였을 때 윈도우 화면(아래 두 윈도우 중 하단 윈도우)을 확인하시오.

```
public class Test extends JFrame {
    public Test() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(200,100);
        setLocationRelativeTo(null);
        Container container=getContentPane();
        // container.setLayout(new FlowLayout());
        container.add(new JButton("OK"));
        container.add(new JButton("CANCEL"));
    }
    public static void main(String[] args) {
        new Test().setVisible(true);
    }
}
```

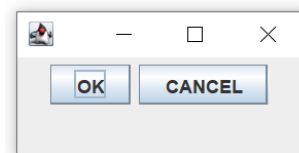


3. (**JPanel, FlowLayout**) 다음 프로그램 소스를 입력하고 실행하시오. `new JPanel();`은 FlowLayout 설정된 JPanel 객체를 반환한다. 컨테이너인 JPanel에 FlowLayout 방식으로 두 개의 JButton 객체를 올린 다음, 그 JPanel 객체를 JFrame의 내용창(contentPane)에 추가한다.

- 실습 #1: 아래 코드 실행 후 표시되는 윈도우의 크기를 변경하면서 내부 버튼들의 변화를 관찰하면서 FlowLayout의 동작 방식을 이해하시오.

- 실습 #2: 아래 코드에서 `add(jPanel);` 문장을 `setContentPane(jPanel);` 문장으로 변경한 후 실행하였을 때 윈도우 화면을 확인하시오.

```
public class Test extends JFrame {
    public Test() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(200,100);
        setLocationRelativeTo(null);
        JPanel jPanel=new JPanel(); // FlowLayout()으로 설정된 JPanel 객체 반환
        jPanel.add(new JButton("OK"));
        jPanel.add(new JButton("CANCEL"));
        add(jPanel); // setContentPane(jPanel);
    }
    public static void main(String[] args) {
        new Test().setVisible(true);
    }
}
```

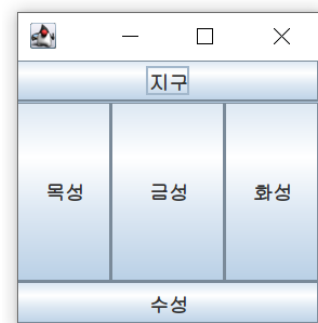


4. (**BorderLayout**) 다음 프로그램 소스를 입력하고 실행하시오. 기본 JPanel 객체는 FlowLayout으로 설정되어 있으므로 `jPanel.setLayout(new BorderLayout());`을 통해 BorderLayout으로 변경한다. 아래 코드는 BorderLayout 설정된 JPanel 객체에 지구 버튼을 북쪽에, 수성 버튼을 남쪽에, 금성 버튼을 중앙에, 화성 버튼을 동쪽에 목성 버튼을 서쪽에 배치한 후, 그 JPanel 객체를 JFrame의 내용창에 추가한다.

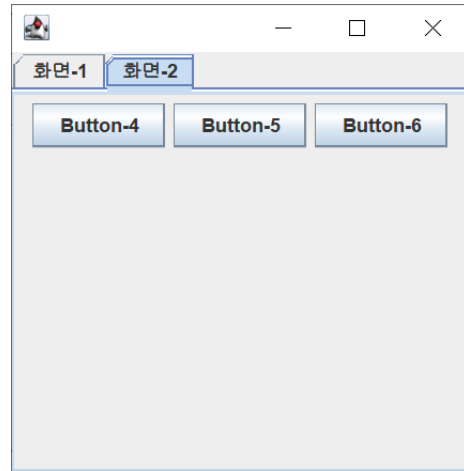
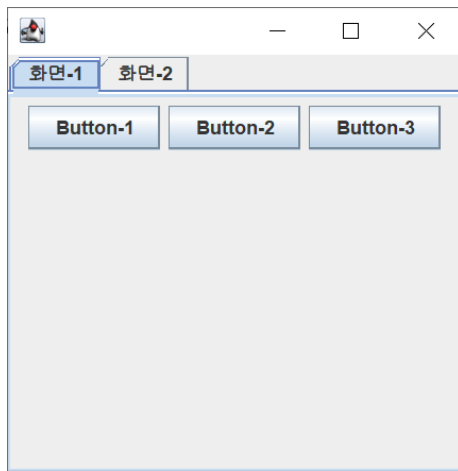
- 실습 #1: 아래 코드 실행 후 표시되는 윈도우의 크기를 변경하면서 내부 버튼들의 변화를 관찰하면서 BorderLayout의 동작 방식을 이해하시오.

- 실습 #2: 아래 코드에서 동,서,남,북,중앙의 각 영역 버튼 추가 문장(들)을 주석처리한 후 윈도우의 실행 화면을 확인하면서 BorderLayout의 동작을 익히시오.

```
public class Test extends JFrame {  
    public Test() {  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setSize(400,400);  
        JPanel jPanel=new JPanel(); // FlowLayout()  
        jPanel.setLayout(new BorderLayout());  
        jPanel.add(new JButton("지구"),BorderLayout.NORTH);  
        jPanel.add(new JButton("수성"),BorderLayout.SOUTH);  
        jPanel.add(new JButton("금성"),BorderLayout.CENTER);  
        jPanel.add(new JButton("화성"),BorderLayout.EAST);  
        jPanel.add(new JButton("목성"),BorderLayout.WEST);  
        add(jPanel);  
    }  
    public static void main(String[] args) {  
        new Test().setVisible(true);  
    }  
}
```

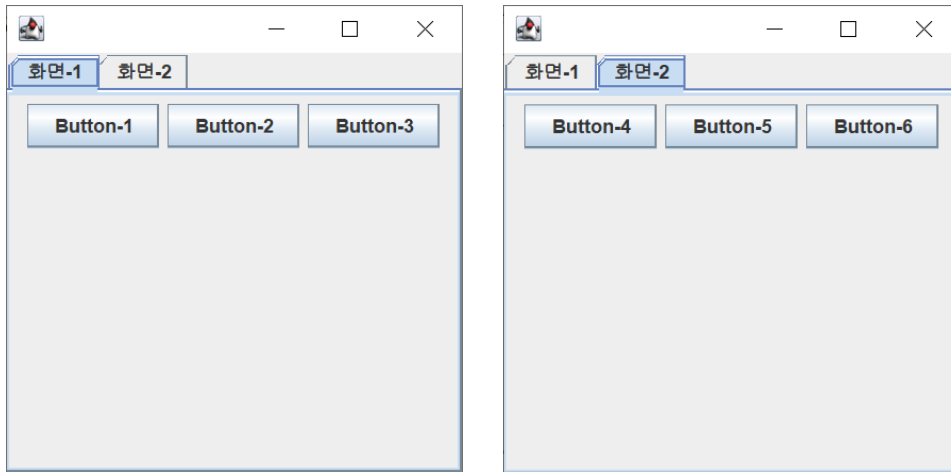


5. (JTabbedPane) 다음 프로그램 소스를 입력하고 실행하시오.



```
public class Test extends JFrame {
    public Test() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 300);
        setLocationRelativeTo(null);
        JTabbedPane jTabbedPane=new JTabbedPane();
        jTabbedPane.add("화면-1", myPane1());
        jTabbedPane.add("화면-2", myPane2());
        add(jTabbedPane);
    }
    private JPanel myPane1() {
        JPanel jPanel=new JPanel();
        for (int i = 1; i <= 3; i++) jPanel.add(new JButton("Button-"+i));
        return jPanel;
    }
    private JPanel myPane2() {
        JPanel jPanel=new JPanel();
        for (int i = 4; i <= 6; i++) jPanel.add(new JButton("Button-"+i));
        return jPanel;
    }
    public static void main(String[] args) {
        new Test().setVisible(true);
    }
}
```

6. (JTabbedPane) 다음 프로그램 소스를 입력하고 실행하시오.



```
public class Test extends JFrame {
    JTabbedPane jTabbedPane=new JTabbedPane();
    public Test() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 300);
        setLocationRelativeTo(null);
        jTabbedPane.add("화면 1", new TestTab1());
        jTabbedPane.add("화면 2", new TestTab2());
        add(jTabbedPane);
    }
    public static void main(String[] args) {
        new Test().setVisible(true);
    }
}

public class TestTab1 extends JPanel {
    public TestTab1() {
        for (int i = 1; i <= 3; i++) add(new JButton("Button-"+i));
    }
}

public class TestTab2 extends JPanel {
    public TestTab2() {
        for (int i = 4; i <= 6; i++) add(new JButton("Button-"+i));
    }
}
```

## References

Java Platform, Standard Edition (Java SE). <https://docs.oracle.com/javase/>

김윤명. (2008). 뇌를 자극하는 Java 프로그래밍. 한빛미디어.

남궁성. 자바의 정석. 도우출판.

김윤명. (2010). 뇌를 자극하는 JSP & Servlet. 한빛미디어.

황기태, 김효수 (2015). 명품 Java Programming. 생능출판사.