

{ JAVA }

선린인터넷  
웹 운영과

# 그래픽 사용자 인터페이스 : GUI-3

- JPanel 컨테이너
- JLabel 컴포넌트

## JFrame 복습

### ❖ 모든 스윙 컴포넌트를 담는 최상위 GUI 컨테이너

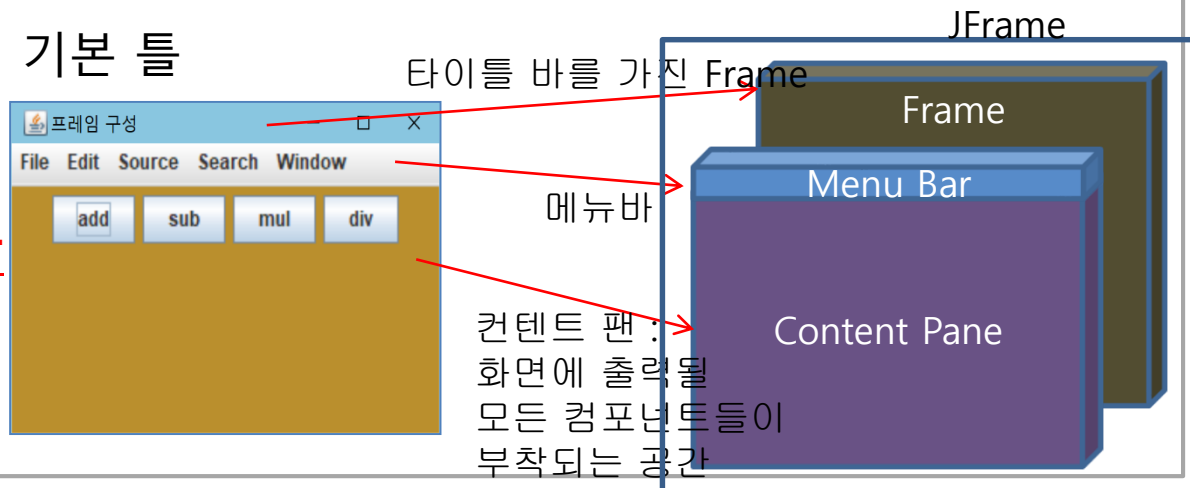
- 컴포넌트가 화면에 보이려면 프레임에 부착되어야 함
- 프레임을 닫으면 프레임 내 모든 컴포넌트가 보이지 않게 됨

### ❖ JFrame 기본구성

- 프레임 - 스윙 프로그램의 기본 틀
- 메뉴바 - 메뉴 부착

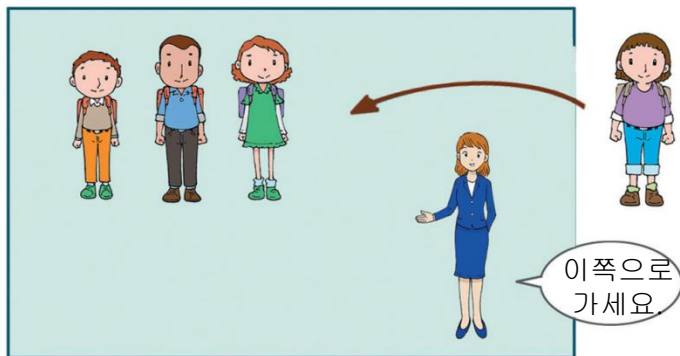


- 컨텐츠팬 - 컴포넌트 부착

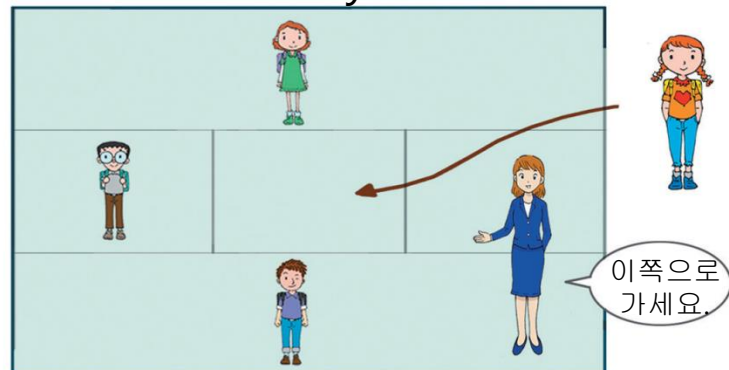


## 컨테이너와 배치 개념 복습

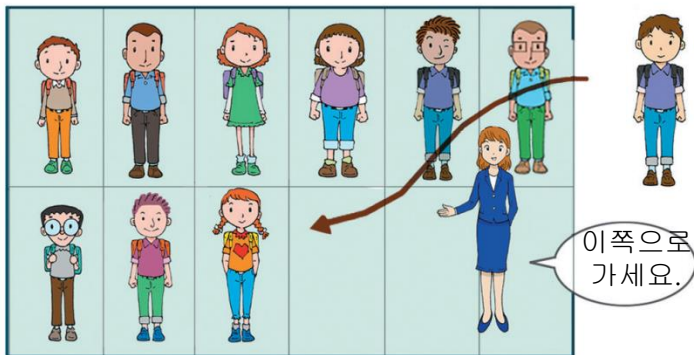
FlowLayout



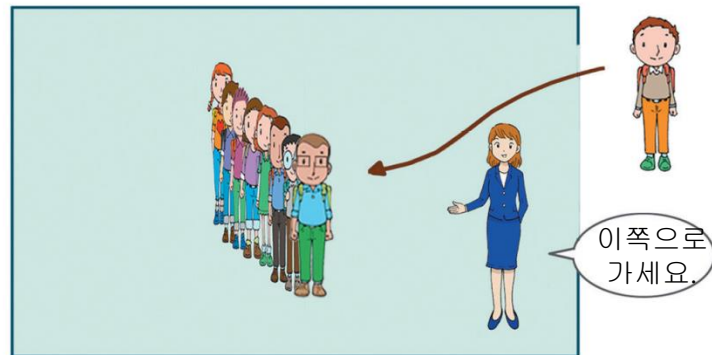
BorderLayout



GridLayout



Null : 레이아웃 없음



# JPanel 컨테이너

□ JPanel 클래스 : 교과서 181쪽

- 필요한 button, label 등의 컴포넌트들을 그룹별로 묶을 때 사용하는 컨테이너
  - JPanel에 컴포넌트 추가시 메소드 : JPanel변수.add(Component변수);
  - JPanel의 색상 설정 : JPanel변수.setBackground(Color.각종색상상수);
  - ※ JAVA API에서 JPanel 클래스의 다른 메소드들을 확인해보자.
- 화면에 보여지기 위해서는 다른 패널 또는 프레임(JFrame) 등 컨테이너에 추가돼야 함  
즉, JPanel 혼자서는 실행될 수 없고, JFrame같은 컨테이너에 포함되어야 함

## 자바의 GUI (Graphical User Interface)

### ❖ JPanel 컨테이너 : 컨테이너의 일종

- **컴포넌트들을 포함**할 수 있도록 설계된 **컨테이너**의 일종
- 패널을 통해 **유지 보수** 및 **배치 관리**에 유리한 경우가 많음

### ❖ JLabel 컴포넌트

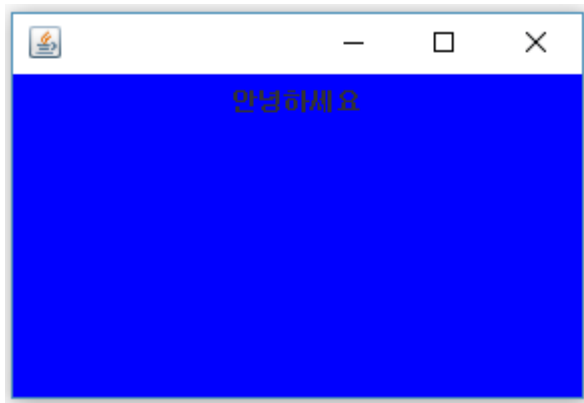
- 편집이 불가능한 **텍스트를 표시**하기 위한 컴포넌트
- 컴포넌트들의 캡션 표시, 도움이 되는 **정보** 또는 계산 결과 **표시** 등으로 사용
- 텍스트와 이미지를 동시에 표시할 수 있음
- 레이블에 나타난 **텍스트의 속성 변경 가능** : 폰트의 종류, 크기, 색상 등

1번째

# JPanel을 만들어보자

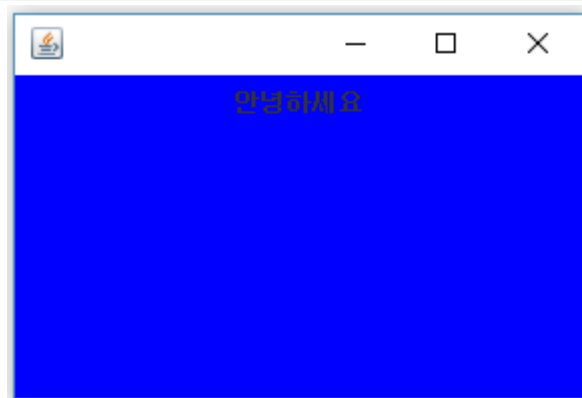
- JPanel 배경색상: 파란색
- JFrame에 붙여 화면(300\*200) 출력
- JLabel도 붙여보자 ← “안녕하세요”

Exam1 : 레이블 1개가 부착된 JPanel을 만들어보자. → JPanel은 JFrame에 부착



Exam1 : 레이블 1개가 부착된 JPanel을 만들어보자. → JPanel은 JFrame에 부착

```
1 import java.awt.Color;
2 import javax.swing.*;
3 public class Exam1 extends JFrame {
4     public Exam1() {
5         setSize(300, 200);
6         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
7
8         JPanel panel = new JPanel(); // panel 컨테이너 생성
9         // 참고: JPanel은 기본 배치관리자가 FlowLayout 객체임
10        panel.add(new JLabel("안녕하세요")); //panel에 라벨 부착
11        panel.setBackground(Color.blue); //panel 배경색 설정
12        add(panel); // JFrame의 콘텐츠팬에 panel컨테이너를 부착
13
14        setVisible(true);
15    }
16    public static void main(String[] args) {
17        new Exam1();
18    }
19 }
```



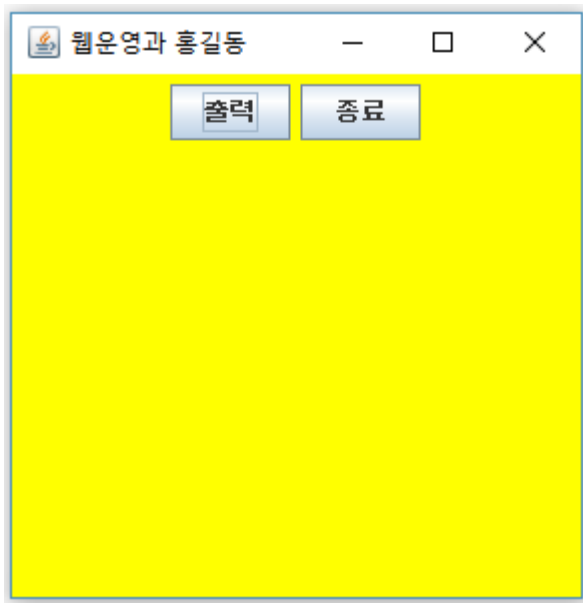


## 2번째

# JPanel을 만들어보자

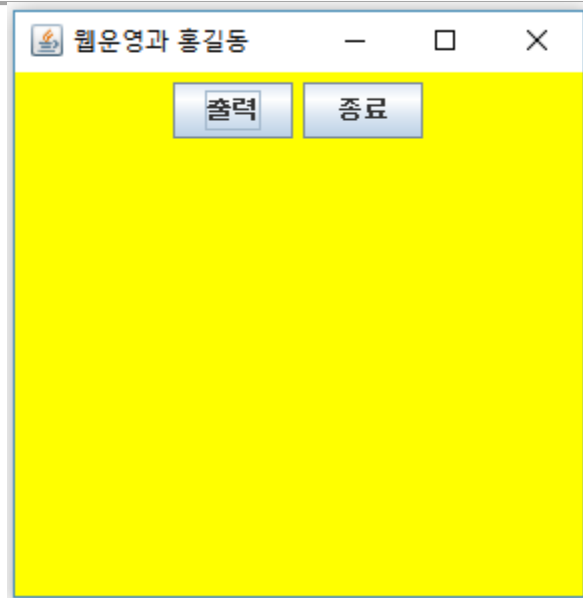
- JPanel 배경색상: 노란색
- JFrame에 붙여 화면(300\*300) 출력
- JButton 2개 부착 ← "출력", "종료"
- JFrame타이틀 ← "웹운영과 홍길동"

Exam2 : 버튼 2개가 부착된 JPanel을 만들어보자. → JPanel은 JFrame에 부착



Exam2 : 버튼 2개가 부착된 JPanel을 만들어보자. → JPanel은 JFrame에 부착

```
1 import java.awt.Color;
2 import javax.swing.*;
3 public class Exam2 extends JFrame {
4     public Exam2(String title) {
5         super(title);
6         setSize(300, 300);
7         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
8
9         JPanel panel = new JPanel(); // panel 컨테이너 생성
10        // 참고: JPanel은 기본 배치관리자가 FlowLayout 객체임
11        panel.add(new JButton("출력")); //panel에 라벨 부착
12        panel.add(new JButton("종료")); //panel에 라벨 부착
13        panel.setBackground(Color.YELLOW); //panel 배경색 설정
14        add(panel); // JFrame의 콘텐츠팬에 panel컨테이너를 부착
15
16        setVisible(true);
17    }
18    public static void main(String[] args) {
19        new Exam2("웹운영과 홍길동");
20    }
21 }
```



## 3번째

# JPanel을 만들어보자

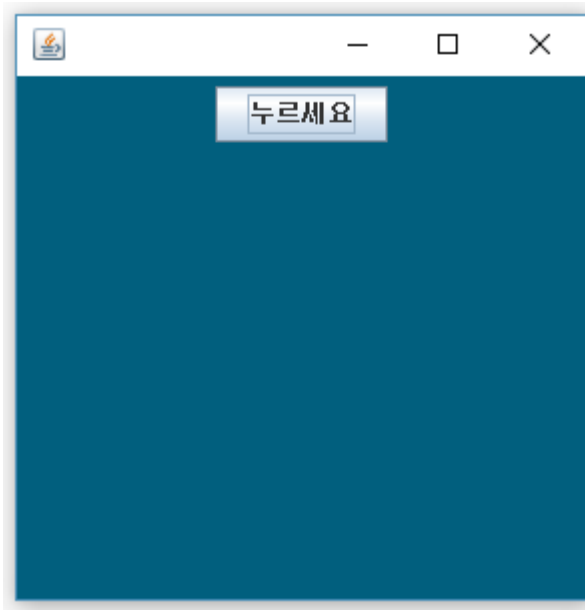
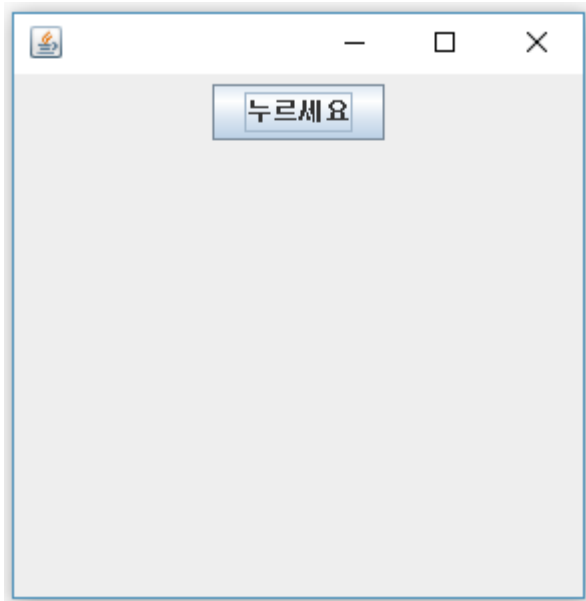
- JPanel 배경색상: 처음에는 없음
- JFrame에 붙여 화면(300\*300) 출력
- JButton 1개 부착 ← “누르세요”  
→ 버튼 누르면 JPanel 색상이 랜덤 변함

## Exam3 : 버튼 누르면 JPanel 배경색이 랜덤으로 변경되게...

임의의 색을 생성하는 코드는 다음과 같이 작성해도 됩니다. 참고하세요.

```
int r = (int)(Math.random()*256);  
int g = (int)(Math.random()*256);  
int b = (int)(Math.random()*256);  
Color c = new Color(r,g,b); // c는 랜덤색
```

## Exam3 : 버튼 누르면 JPanel 배경색이 랜덤으로 변경되게...



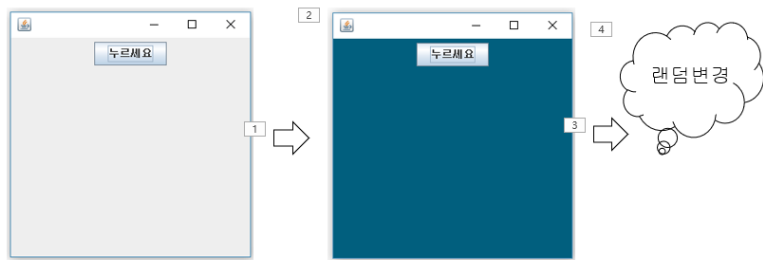
## GUI 맛보기

## Exam3 : 버튼을 누르면 JPanel배경색상이 랜덤으로 변하게...

```

1 import java.awt.Color;
2 import java.awt.event.*;
3 import javax.swing.*;
4 public class Exam3 extends JFrame implements ActionListener{
5     JPanel panel;
6     public Exam3() {
7         setSize(300,300);
8         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9
10        panel = new JPanel(); // panel 컨테이너 생성
11        JButton button = new JButton("누르세요"); button.addActionListener(this);
12        panel.add(button); // 패널에 버튼 부착
13        add(panel); // 콘텐츠 팬에 "panel 컨테이너" 부착
14
15        setVisible(true);
16    }
17    @Override
18    public void actionPerformed(ActionEvent arg0) {
19        int r = (int)(Math.random()*256), g = (int)(Math.random()*256);
20        int b = (int)(Math.random()*256);
21        panel.setBackground(new Color(r,g,b)); // panel의 배경색 랜덤 설정
22    }
23    public static void main(String[] args) {
24        new Exam3();
25    }
26 }

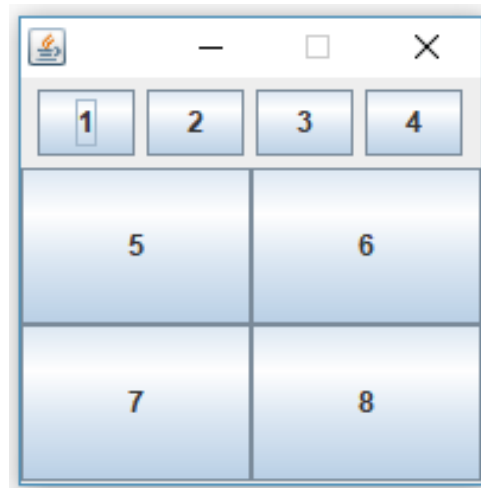
```



4번째

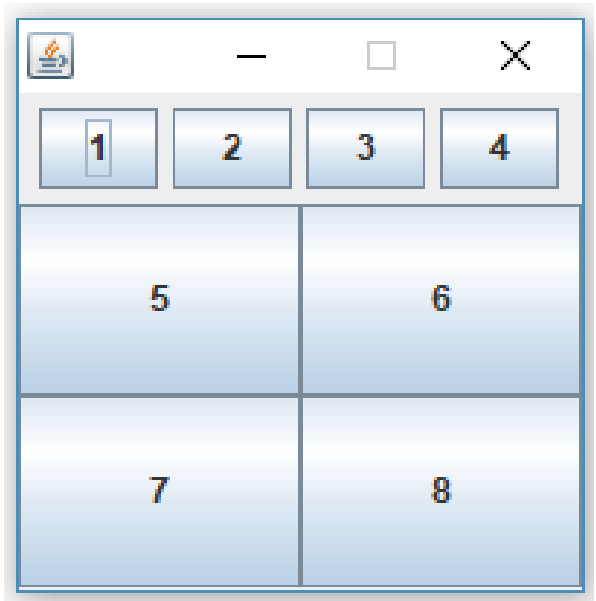
# JPanel을 만들어보자

- Frame 사이즈 200\*200



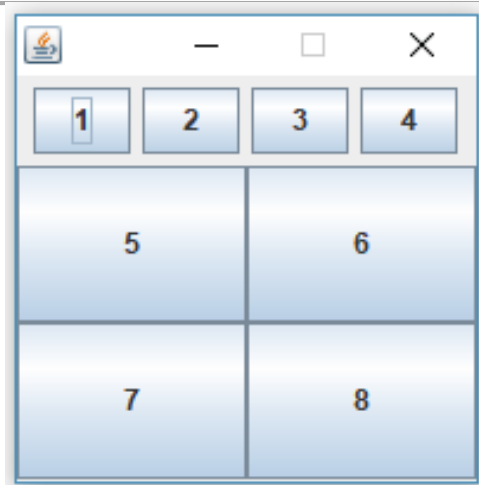


Exam4



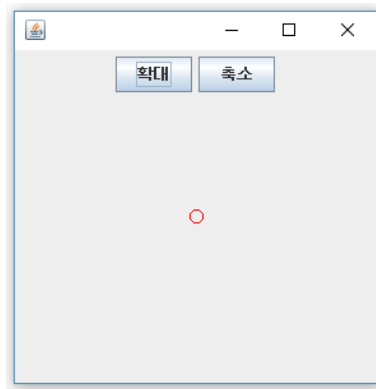
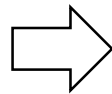
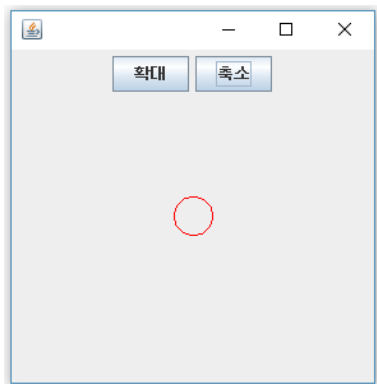
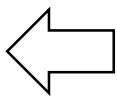
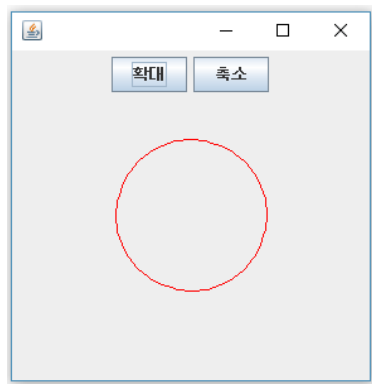
## Exam4

```
1 import java.awt.*;
2 import javax.swing.*;
3 public class Exam4 extends JFrame {
4     JPanel p1, p2;
5     public Exam4() {
6         setSize(200, 200); // frame 사이즈를 200*200 픽셀로 정함
7         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
8         setResizable(false); // frame의 크기 조절을 못 하게 설정함
9
10        p1 = new JPanel(); // 패널 기본 배치관리자: FlowLayout()
11        p2 = new JPanel(new GridLayout(2, 2));
12        add(p1, BorderLayout.NORTH); // 컨테인트팬에 p1패널을 부착
13        add(p2, BorderLayout.CENTER); // 컨테인트팬에 p2패널을 부착
14
15        for(int i=1;i<5;i++) { // p1,p2패널에 버튼을 각각 부착
16            p1.add(new JButton(Integer.toString(i)));
17            p2.add(new JButton(Integer.toString(i+4)));
18        }
19
20        setVisible(true); // frame을 보이게 설정함
21    }
22    public static void main(String[] args) {
23        new Exam4();
24    }
25 }
```



5번째

# JPanel을 만들어보자



# GUI 맛보기

## Exam5 Exam5클래스

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4 public class Exam5 extends JFrame {
5     public Exam5() {
6         setSize(300, 300);
7         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
8         add(new Canvas());
9         setVisible(true);
10    }
11    public static void main(String[] args) {
12        new Exam5();
13    }
14 }
```

## GUI 맛보기

## Exam5 Canvas클래스

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4 public class Canvas extends JPanel implements ActionListener {
5     private int radius = 5;
6     JButton up, down;
7     public Canvas() {
8         up = new JButton("확대"); up.addActionListener(this); add(up);
9         down = new JButton("축소"); down.addActionListener(this); add(down);
10    }
11    public void up() { radius += 5; repaint(); }
12    public void down() { radius -= 5; repaint(); }
13    @Override
14    public void actionPerformed(ActionEvent e) {
15        if(e.getSource() == up) up();
16        else down();
17    }
18    @Override
19    protected void paintComponent(Graphics g) {
20        super.paintComponent(g);
21        g.setColor(Color.RED);
22        g.drawOval(getWidth()/2-radius, getHeight()/2-radius, radius*2, radius*2);
23    }
24 }
```

