



세종대학교

Report

명품 JAVA 프로그래밍

제출일 2018.11.17

전공 디지털콘텐츠학과

학번 16013093

이름 박상우

Hw 3_1

GridLayout을 이용해 버튼을 10개 배치하고 각 버튼의 색을 바꾸는 문제입니다.

```
JPanel p = new JPanel();
p.setLayout(new GridLayout(1, 10));
JButton tmp[] = new JButton[10];

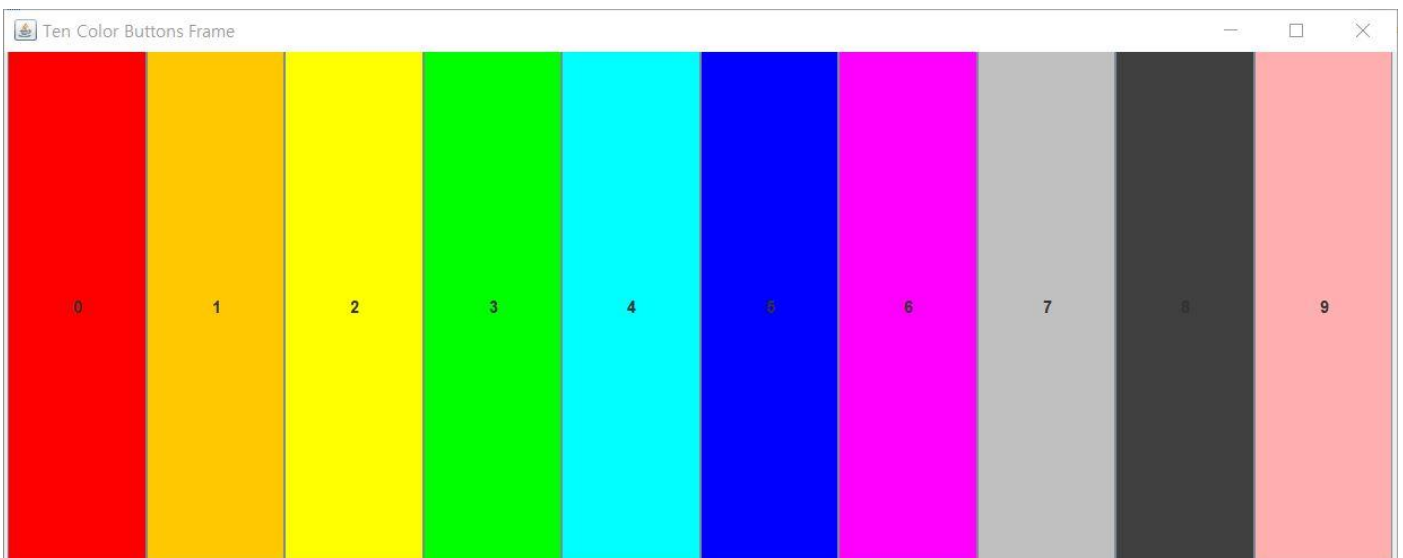
for(int i = 0; i < 10; i++)
{
    String str = i + "";
    tmp[i] = new JButton(str);
    p.add(tmp[i]);
}
```

위와 같은 방법으로 10개의 버튼은 만들었

```
tmp[0].setBackground(Color.RED);
tmp[1].setBackground(Color.ORANGE);
tmp[2].setBackground(Color.YELLOW);

.....
```

각 버튼의 setBackground() 메소드를 호출해 색을 바꾸었습니다.



Hw3_2

계산기와 같은 모양의 프레임을 작성하는 문제입니다.

위쪽의 JLabel, JTextField를 포함하는 JPanel을 붙이고 가운데에 16개의 4X4버튼, 아래쪽에 위쪽과 마찬가지로 JLabel, JTextField를 포함하는 JPanel을 붙여 모양을 완성시켰습니다.

JTextField는 생성자로 칸 수를 지정할 수 있고 400x300 창에서 30칸 정도가 적당한 입력창 크기였습니다.

가운데 16개의 버튼 중 마지막 연산기호 부분, 위, 아래패널의 배경색도 설정했습니다.

```
JPanel topPanel = new JPanel();
JLabel text1 = new JLabel("수식입력");
JTextField in = new JTextField(30);
topPanel.setBackground(Color.LIGHT_GRAY);
topPanel.add(text1);
topPanel.add(in);
add(topPanel, BorderLayout.NORTH);
```

위와 같은 방법으로 위 아래에 패널을 추가했습니다.

가운데 버튼의 경우

```
buttonPanel.setLayout(new GridLayout(4, 4));
JButton tmp[] = new JButton[16];
```

로 4x4버튼을 만들고 반복문을 돌며 각 버튼들을 설정했습니다.



Hw3_3

왼쪽 방향키를 누를 시 Love Java가 한 문자씩 이동하는 프로그램입니다.

```
class KeyShift extends JPanel implements KeyListener
```

패널에 인터페이스로 키리스너를 추가해 구현했습니다.

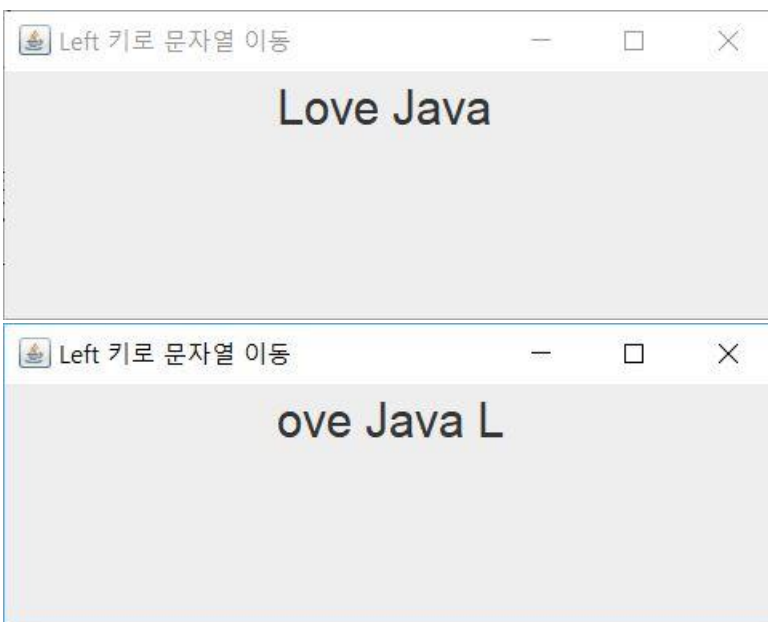
```
KeyShift() {  
    l = new JLabel("Love Java ");  
    l.setFont(new Font("Arial", Font.PLAIN, 30));  
    add(l);  
  
    this.setFocusable(true);  
    this.requestFocus();  
    this.addKeyListener(this);  
}
```

생성자에서 JLabel을 초기화 하고 글씨를 크게 표시하기위해 폰트를 수정했습니다.

키입력을 받기위해 KeyListener를 추가하고 포커스도 설정했습니다.

```
@Override  
public void keyPressed(KeyEvent e) {  
    if (e.getKeyCode() == KeyEvent.VK_LEFT) {  
        String firstChar = l.getText().substring(0, 1);  
        String str = l.getText().substring(1);  
  
        l.setText(str + firstChar);  
    }  
}
```

왼쪽키가 눌리면 JLabel의 첫 문자를 잘라서 fistChar에 저장하고 나머지를 str에 저장해 str + firstChar 순서로 다시 JLabel을 설정해 한칸씩 이동하도록 구현했습니다.



Hw3_4

랜덤위치로 움직이는 JLabel 만들기 프로그램입니다.

```
RandomJLabel() {  
    setLayout(null);  
    l = new JLabel("c");  
    l.setSize(10, 10);  
    l.setLocation(100, 100);  
    add(l);  
    addMouseListener(this);  
}
```

배치관리자를 지우고 JLabel을 초기화 합니다.

마우스리스너를 추가해 마우스클릭을 입력 받을 수 있게 했습니다.

```
@Override  
public void mouseClicked(MouseEvent arg0) {  
    int x = (int) (Math.random() * 500);  
    int y = (int) (Math.random() * 500);  
  
    l.setLocation(x, y);  
}
```

프레임 크기를 500, 500으로 설정했기에 0~500사이 숫자를 랜덤으로 얻어 그 위치로 이동하게 했습니다.

