

REPORT



- 과목명 : 객체지향프로그래밍
- 담당교수 : 엄진영 교수님
- 학과 : 컴퓨터공학과
- 학번 : 2019112130
- 이름 : 조 양 진

객체지향프로그래밍 14주차 실습과제 보고서

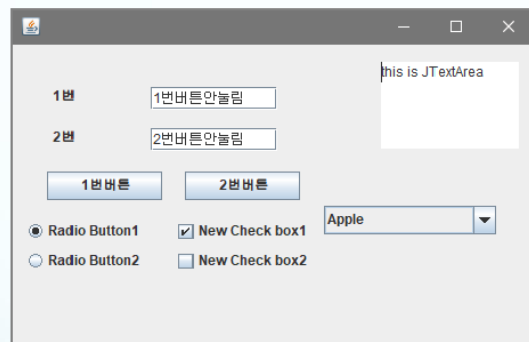
2019112130 조양진

목차

1. 문제 1 - 2 페이지
2. 문제 분석 - 2페이지
3. 코드 설명 - 4페이지
4. 결과 및 분석 - 8페이지
5. 소감 - 11페이지

문제 1.

아래와 같이 작동하는 프로그램을 작성하시오



기본화면



1번버튼을 눌렀을 경우

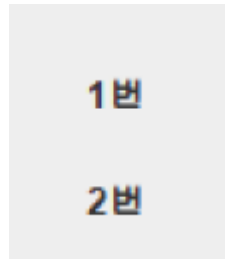
조건

- 1번버튼, 2번버튼을 눌렀을 경우 1번, 2번에 각각 버튼이 눌렸다는 멘트를 메소드를 이용해 구현
- 프로그램 실행시 기본화면의 Radio Button 1과 New Check box 1은 체크가 되어 있는 상태여야 함
- JLabel, JText 등 기본화면에 나와있는 각각의 구성 요소는 다 포함되어 있어야 하며 위치나 크기는 달라도 상관 없음.

- 문제 분석

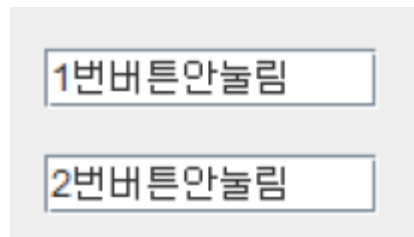
JAVA의 Swing을 사용하여 GUI를 구현해야 합니다. 우선 예시로 주어진 화면을 분석해보면 크게 7가지의 컴포넌트들이 보입니다.

1. JLabel



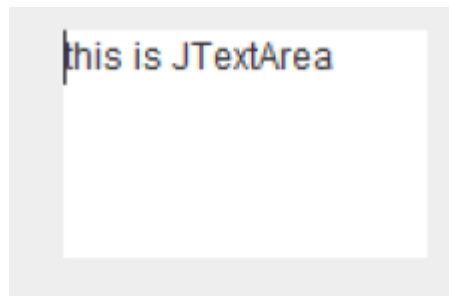
이 부분은 JLabel을 통해 1번, 2번을 작성해서 프레임 위에 그린 것으로 보입니다.

2. JTextField



JLabel 옆에 있는 텍스트 필드도 기본적으로는 "1번버튼안눌림"과 같이 설정되어 있다가 해당하는 버튼이 눌리면 "1번눌림"으로 바뀌는 부분입니다. 이는 JButton에서 설정해줘야 합니다.

3. JTextArea



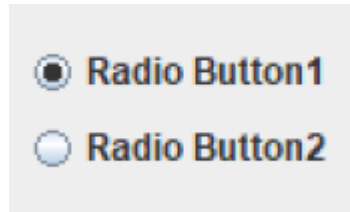
JTextArea안에 기본 텍스트를 설정해 줘야 합니다.

4. JButton



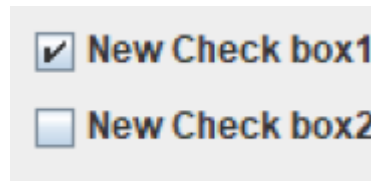
JLabel과 JTextField 밑에 JButton 2개를 만들어야 합니다. 또한 해당 버튼을 누르면 일어나는 행동을 addActionListener 메소드를 사용하여 지정해줘야 합니다.

5. JRadioButton



1번 JButton 밑에 있는 JRadioButton으로서 위에 있는 RadioButton1은 프로그램이 시작될 때부터 체크되어 있어야 합니다. 이는 해당 JButton객체의 setSelected 메소드를 사용하면 됩니다.

6. JCheckBox



JRadioButton 옆에 있는 체크박스는 JCheckBox로서 마찬가지로 New Check box1은 프로그램이 시작될 때부터 체크되어 있어야 합니다. 이도 setSelected 메소드를 사용하면 됩니다.

7. JComboBox



String형 배열을 리스트로 담고 있으며 드롭다운 형식 버튼을 만들어주는 이것은 JComboBox입니다.

이 7가지 컴포넌트들을 JFrame 객체 위에 add로 추가하여 실행하면 됩니다.

- 코드 설명 (week14 -Problem_01.java)

Line 3~4

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

GUI를 사용하기 위해서 swing과 awt를 import 해옵니다.

Line 8~12

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    Problem_01 gui = new Problem_01();
    gui.go();
}
```

프로그램을 실행시키기 위한 메인 함수는 간단히 본인 클래스 객체를 생성하고 go라는 메소드를 실행시키는 것이 다 입니다.

Line 15~21

```
public static void go() {
    JFrame frame = new JFrame();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JPanel panel1 = new JPanel();
    panel1.setLayout(null);
    frame.add(panel1);
    frame.setLayout(null);
}
```

go 메소드에서는 JFrame을 사용하여 프레임을 만들어주고 화면을 껴둘 때 어떻게 할지 정해줍니다. 또한 패널도 생성합니다

Line 23~27

```
// this is JTextArea
JTextArea text = new JTextArea(5,5);
text.setText("This is JTextArea");
text.setBounds(400,50,150,100);
frame.add(text);
```

JTextArea 컴포넌트를 생성합니다. 이후 기본 텍스트로 "This is JTextArea"라는 문구를 저장합니다. setBounds 메소드를 통해 절대 좌표로 프레임 안에 위치를 정해줍니다. (int x, int y, width, height) (x:400, y:50, 가로:150, 세로:100) 이후 프레임에 추가합니다.

Line 29~37

```
// 버튼이 눌렸는지 안눌렸는지 표기
JTextField textField1 = new JTextField();
JTextField textField2 = new JTextField();
textField1.setText("1번버튼안눌림");
textField2.setText("2번버튼안눌림");
textField1.setBounds(150,50,100,20);
textField2.setBounds(150,100,100,20);
frame.add(textField1);
frame.add(textField2);
```

JTextField 컴포넌트를 생성합니다. 기본 텍스트로 해당 버튼 안눌림을 저장하고 setBounds 메소드로 위치를 지정해 준 뒤 프레임에 추가합니다.

Line 39~45

```
// 1번, 2번 글자 그리기
JLabel label1 = new JLabel("1번");
label1.setBounds(100,50,20,20);
JLabel label2 = new JLabel("2번");
label2.setBounds(100,100,20,20);
frame.add(label1);
frame.add(label2);
```

JLabel을 사용하여 1번, 2번 글자를 그려줍니다 마찬가지로 위치를 지정하고 프레임에 추가합니다.

Line 47~58

```
// JButton 크기 조절용
Dimension dim = new Dimension(100,30);

// 1번 버튼과 2번 버튼
JButton button1 = new JButton("1번 버튼");
button1.setBounds(40,160,100,30);
button1.setSize(dim);
JButton button2 = new JButton("2번 버튼");
button2.setBounds(150,160,100,30);
button2.setPreferredSize(dim);
frame.add(button1);
frame.add(button2);
```

JButton을 사용하여 1번 2번 버튼을 만들고 위치를 지정해줍니다. 이후 Dimension 객체를 사용하여 해당 버튼의 크기를 setPreferredSize 메소드를 통해 지정해줍니다. 이후 프레임에 추가합니다.

Line 60~66

```
// 1번 버튼과 2번 버튼을 누르면 textField 안의 값 변경
button1.addActionListener(event →{
    textField1.setText("1번눌림");
});
button2.addActionListener(event→{
    textField2.setText("2번눌림");
});
```

button객체를 누르면 일어날 일을 addActionListener 메소드를 통해 지정해줍니다.

Line 68~78

```
// 라디오 버튼 생성
JRadioButton radio1 = new JRadioButton();
radio1.setBounds(40,200,120,30);
JRadioButton radio2 = new JRadioButton();
radio2.setBounds(40,230,120,30);
radio1.setText("Radio Button1");
// 프로그램 시작시부터 체크되어있음
radio1.setSelected(true);
radio2.setText("Radio Button2");
frame.add(radio1);
frame.add(radio2);
```

라디오 버튼 두개를 만들고 radio button1은 미리 체크를 해둡니다. 이후 프레임에 추가합니다.

Line 80~90

```
// 체크박스 버튼 생성
JCheckBox check1 = new JCheckBox();
check1.setBounds(160,200,120,30);
check1.setText("New Check box1");
JCheckBox check2 = new JCheckBox();
check2.setBounds(160,230,120,30);
check2.setText("New Check box2");
// 프로그램 시작시부터 체크되어 있음
check1.setSelected(true);
frame.add(check1);
frame.add(check2);
```

체크박스 버튼 두개를 만들고 new check box1은 미리 체크를 해둡니다. 이후 프레임에 추가합니다.

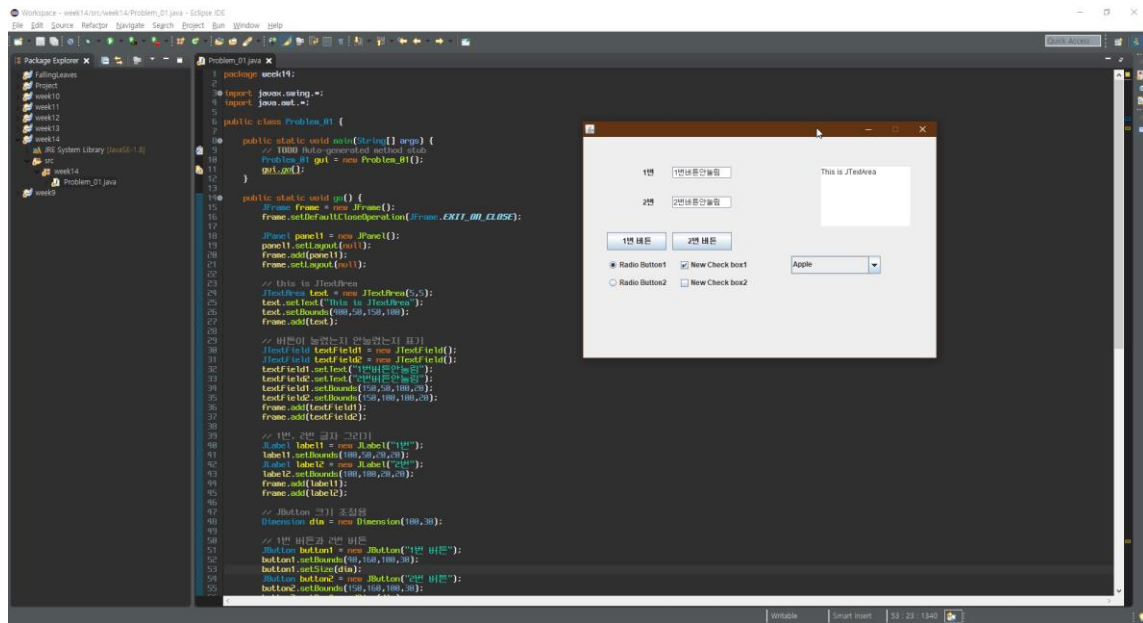
Line 92~101

```
// Drop-down list
// 리스트를 미리 만들고 JComboBox에 집어넣음
String[] listEntries = {"Apple", "Banana", "Gamma", "Delta", "Epsilon"};
JComboBox list = new JComboBox(listEntries);
list.setBounds(350,200,150,30);
frame.add(list);

frame.setSize(600,400);
frame.setResizable(false);
frame.setVisible(true);
```

드롭다운 선택창을 만들기 위해 String형 배열을 만들고 이를 JComboBox에 집어넣습니다. 이후 프레임에 추가합니다. 또한 프레임의 크기와 크기조정 불가를 설정해주는 것으로 go 메소드는 끝이 납니다.

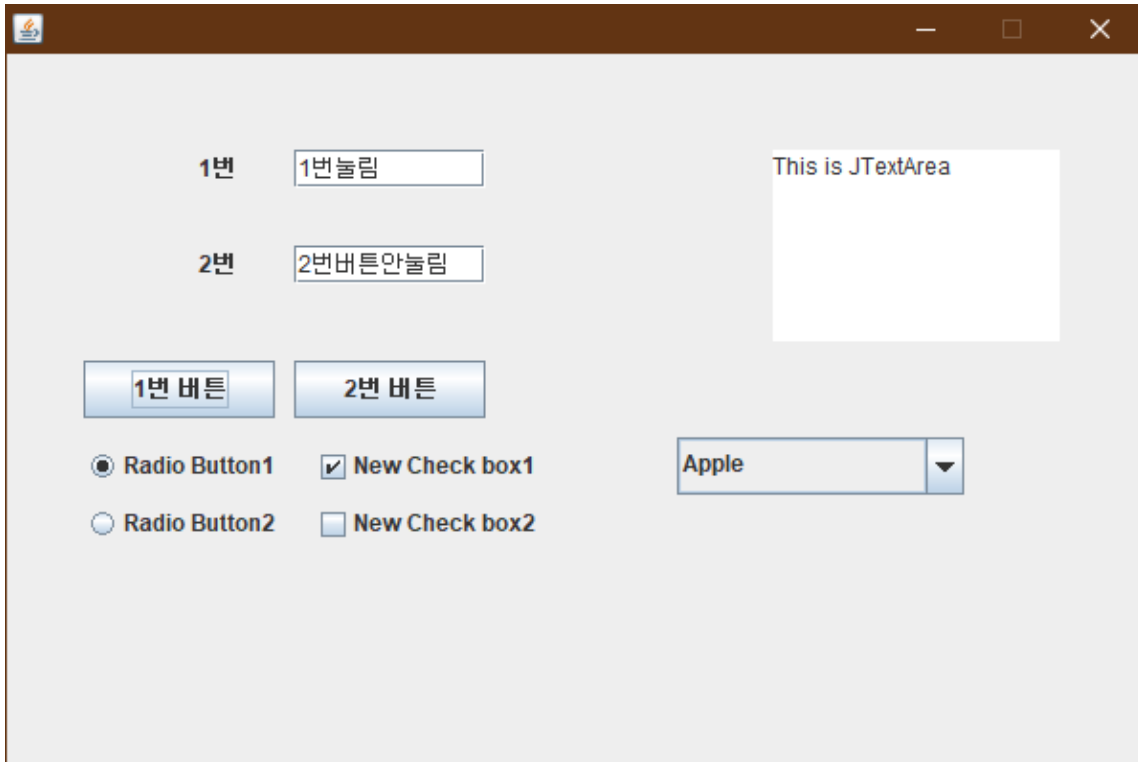
- 결과 및 분석



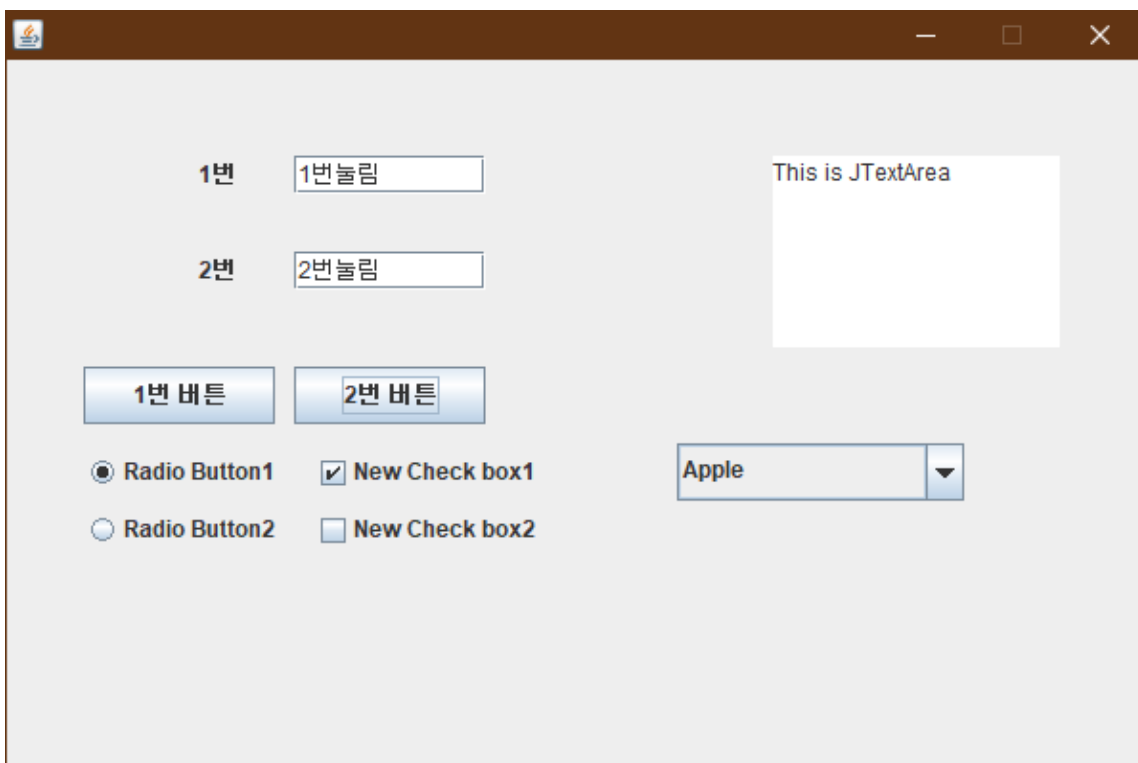
- eclipse 상에서 실행한 모습



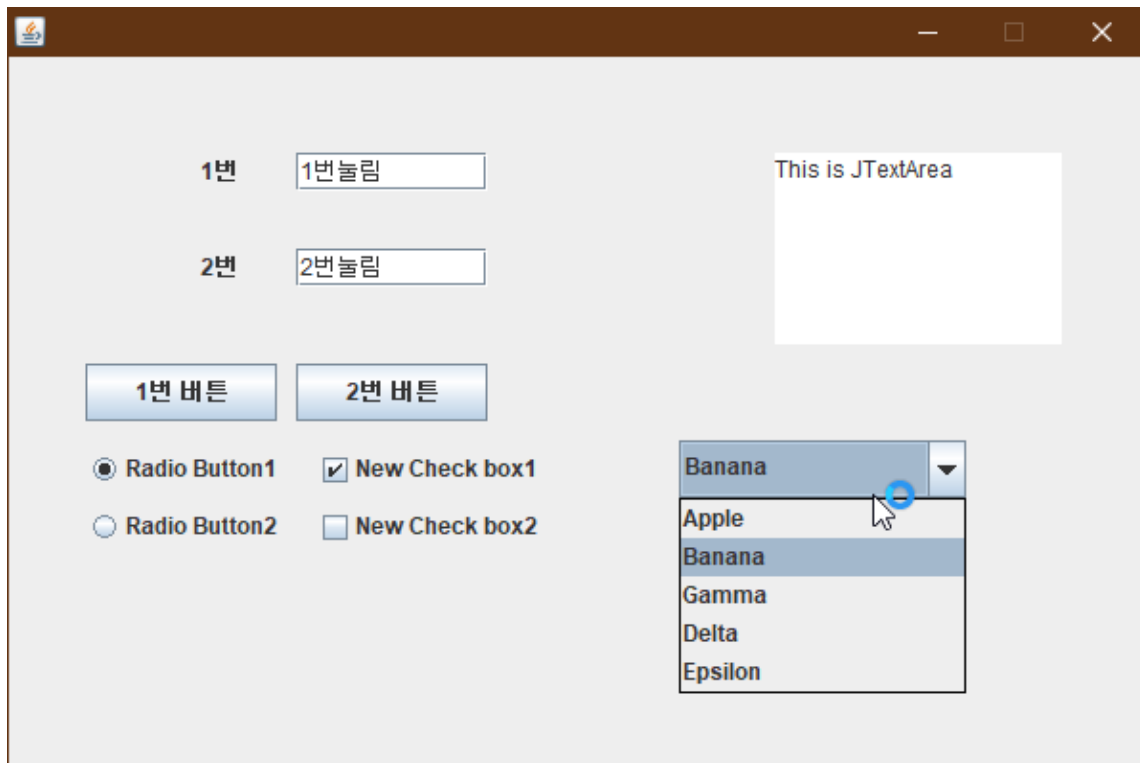
- 처음 실행시켰을 때의 상태



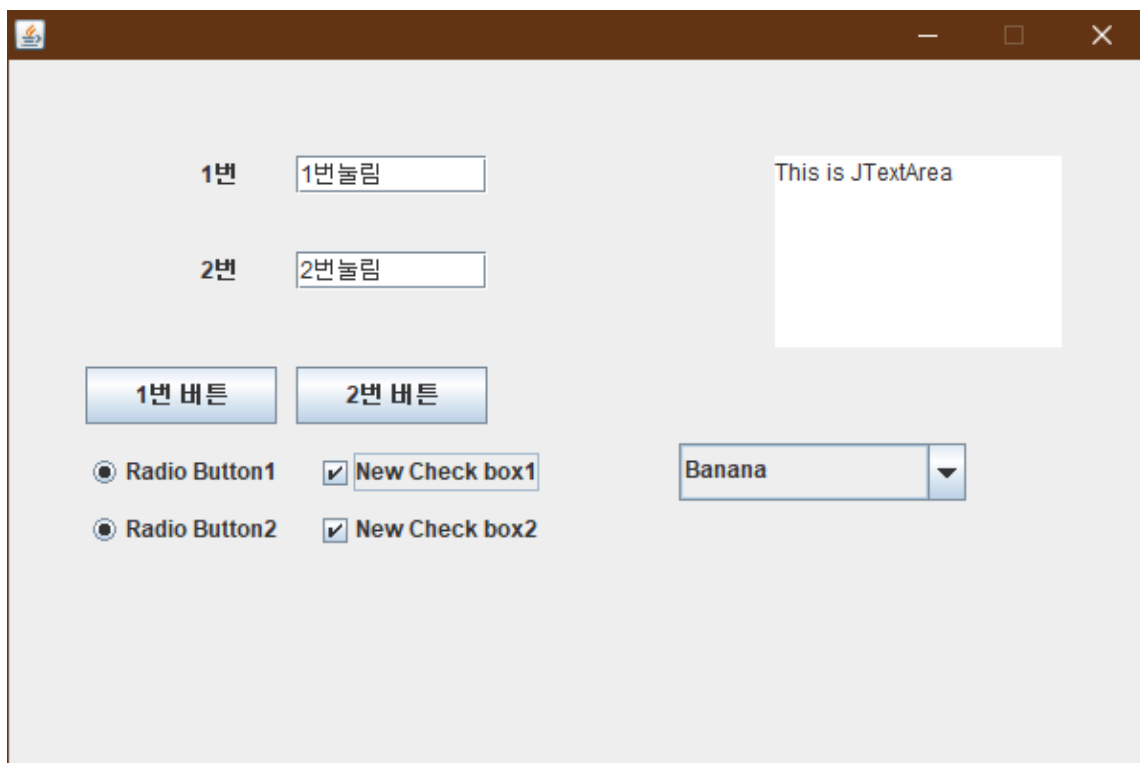
- 1번 버튼을 눌렀을 때의 상태



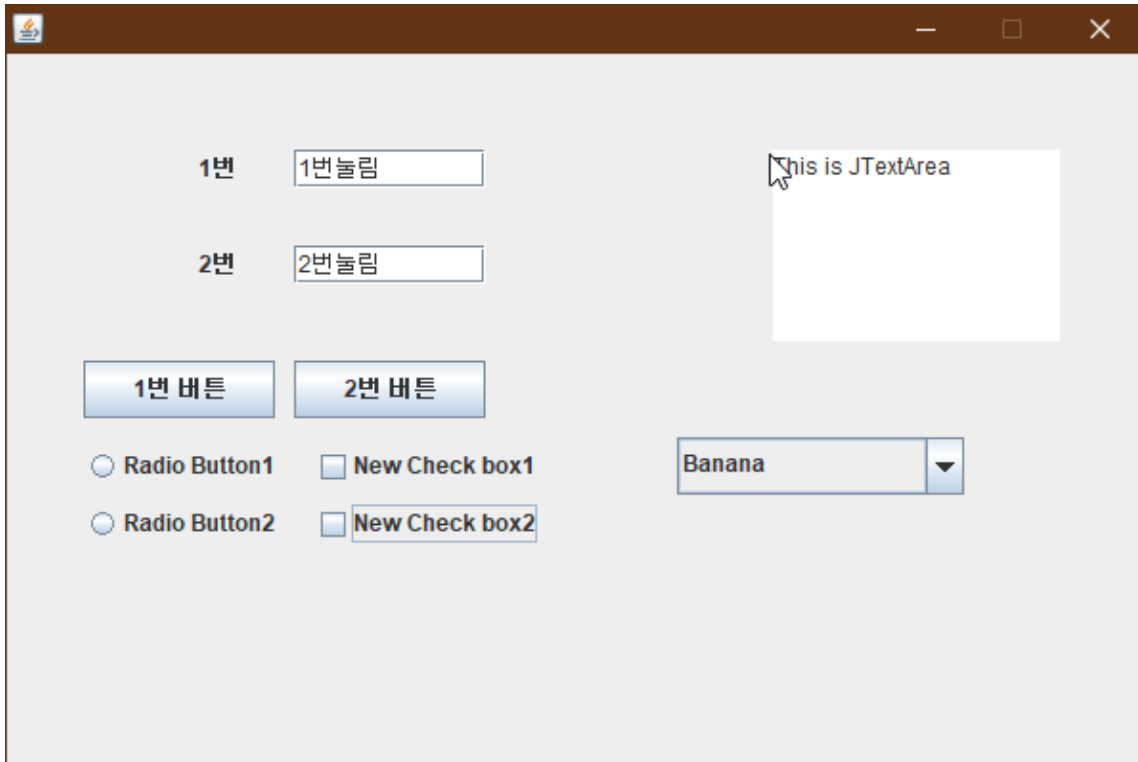
- 2번 버튼을 눌렀을 때



- 콤보박스에서 다른 것을 선택했을 때



- 다른 체크 버튼들을 눌러봤을 때



- 다른 체크 버튼들을 해제했을 때

- 소감

JAVA에서 GUI를 사용하는 것이 이렇게 간편하고 재밌을 수 있다는 것에 놀랐습니다. 앞으로 간단한 프로젝트를 만들 때 GUI가 필요하면 swing을 사용하여 사람들이 좀 더 편하게 사용할 수 있도록 만들 수 있을 것 같습니다. 또한 이 swing GUI는 OS가 달라도 잘 작동하는 것으로 알고 있어서 더욱 기대되는 바입니다.