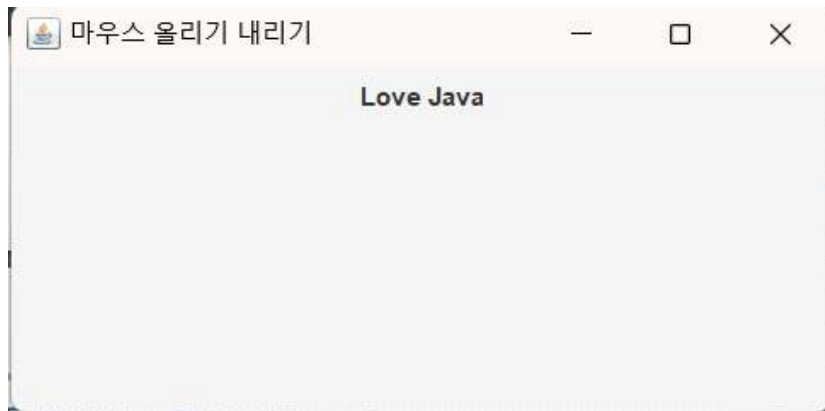


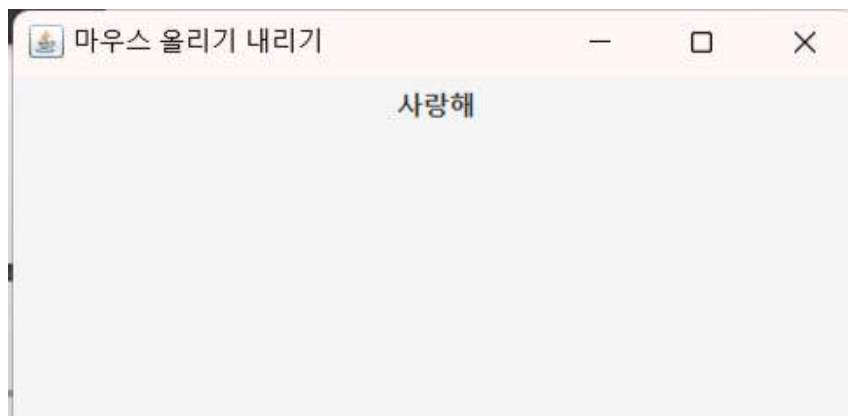
<실습 1> MouseEventFrame

-코드 및 실행결과

```
1  /*작성자: 임다희(2312282)
2   *작성일: 2024-11-08
3   *Lab8-1 MouseEventFrame*/
4
5  import javax.swing.JPanel;
6  import java.awt.*;
7  import javax.swing.*;
8  import java.awt.event.*;
9  //GUI 구성 및 이벤트 처리에 필요한 패키지를 import한다.
10
11 public class MouseEventFrame extends JFrame {
12
13     JLabel text;
14     JPanel panel;
15
16     public MouseEventFrame() {
17
18         setSize(400, 200);
19         setTitle("마우스 클릭기 내리기");
20         // 프레임의 크기와 제목을 지정한다.
21
22         text = new JLabel("Love Java");
23         // JLabel 컴포넌트 text. 초기 내용은 "Love Java"이다.
24
25         panel = new JPanel();
26
27         text.addMouseListener(new MouseAdapter() {
28             // JLabel 컴포넌트에 MouseAdapter를 익명 클래스로 구현해
29             // 마우스 이벤트를 받을 수 있도록 한다.
30
31             public void mouseEntered(MouseEvent e) {
32                 text.setText("사랑해");
33             } // 사용자가 JLabel 컴포넌트 text에 마우스를 들어가게 한 경우
34             // text의 내용을 "사랑해"로 설정한다.
35
36             public void mouseExited(MouseEvent e) {
37                 text.setText("Love Java");
38                 // 사용자가 text에서 마우스를 나가게 할 경우
39                 // text의 내용을 다시 "Love JAVA"로 설정한다.
40             }
41
42         });
43
44         panel.add(text);
45         add(panel);
46         // text를 패널에 추가하고 패널을 프레임에 추가한다.
47
48         setVisible(true);
49         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
50     }
51
52     public static void main(String[] args) {
53         new MouseEventFrame();
54     }
55 }
56
57 }
```



-JLabel 컴포넌트에 마우스를 올렸을 경우

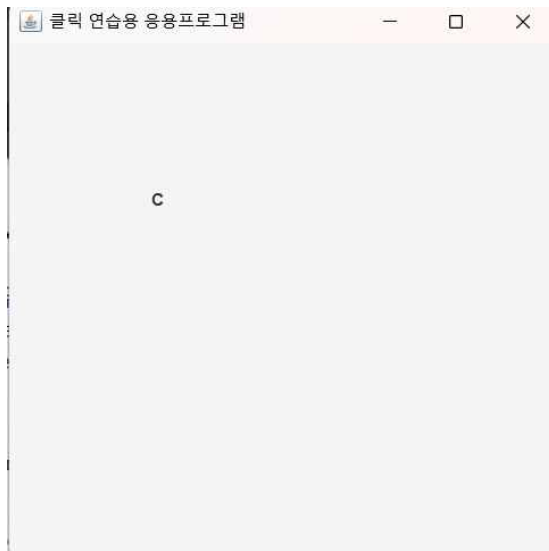


-JLabel 컴포넌트에서 마우스를 내릴 경우

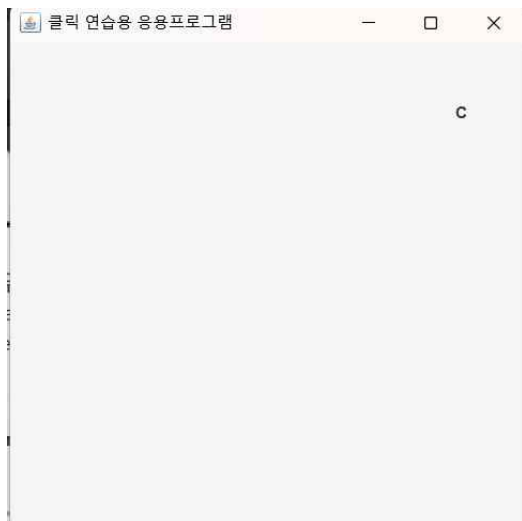
<실습 2> CilckPracticeFrame

-코드 및 실행결과

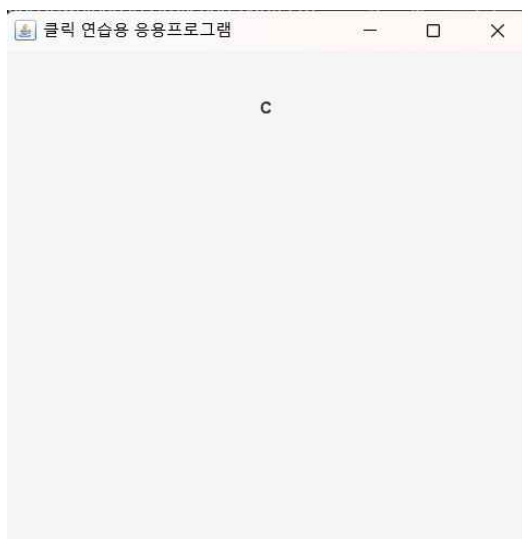
```
1 *작성자: 임다희(2312282)
2 *작성일: 2024-11-08
3 *Lab8-2 CilckPracticeFrame*/
4
5 import java.awt.*;
6 import javax.swing.*;
7 import java.awt.event.*;
8
9 public class ClickPracticeFrame extends JFrame{
10
11     private JLabel c;
12
13     public ClickPracticeFrame() {
14         setTitle("클릭 연습용 응용프로그램");
15         setSize(400, 400);
16
17         setLayout(null); //절대 위치로 요소들을 배치하기 위해 배치 관리자를 null로 설정한다.
18         setVisible(true);
19         //ContentPane의 크기를 받아오기 위해 setVisible 설정을 사전에 해준다.
20
21         c=new JLabel("C"); //"C"를 출력하는 JLabel c
22         c.setBounds(100,100,20,20); //c의 초기 위치와 크기 설정.
23         c.addMouseListener(new MouseAdapter() {
24             //c에 MouseListener를 추가하고 MouseAdapter를 구현한다.
25             public void mouseClicked(MouseEvent e) {
26                 //컴포넌트 c를 클릭하는 이벤트 e가 발생하면 다음과 같이 실행하도록 한다.
27                 int x=(int)(Math.random()*(getContentPane().getWidth()-20+1));
28                 //c의 새로운 x좌표를 지정한다.
29                 //ContentPane의 가로 크기에서 c의 가로 크기를 뺀 값에 1을 더하고
30                 //이를 Math.random()으로 생성한 0 이상 1 미만의 난수 값과 곱한다.
31                 int y=(int)(Math.random()*(getContentPane().getHeight()-20+1));
32                 //c의 새로운 y좌표를 지정한다.
33                 //ContentPane의 세로 크기에서 c의 세로 크기를 뺀 값에 1을 더하고
34                 //이를 Math.random()으로 생성한 0 이상 1 미만의 난수 값과 곱한다.
35                 c.setLocation(x,y);
36                 //c의 위치를 생성된 가로, 세로 좌표 난수값에 따라 새로 지정한다.
37             }
38         });
39
40         add(c); //c를 프레임에 추가한다.
41
42         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
43     }
44
45     public static void main(String[] args) {
46         new ClickPracticeFrame();
47     }
48
49
50
51
```



<- 처음 실행 시 화면.



<- C 1회 클릭 시 화면(위치 랜덤 이동)



<- C 2회 클릭 시 화면(위치 랜덤 이동)

<실습 3> MouseDraggingFrame

-코드 및 실행결과

```
1  /*작성자: 임다희(2312282)
2   *작성일: 2024-11-08
3   *Lab8-3. MouseDraggingFrame*/
4
5  import java.awt.*;
6  import javax.swing.*;
7  import java.awt.event.*;
8  //GUI 구성 및 이벤트 처리에 필요한 패키지를 import한다.
9
10 public class MouseDraggingFrame extends JFrame{
11
12     public MouseDraggingFrame() {
13         setTitle("드래깅동안 YELLOW");
14         setSize(400, 200); // 프레임의 크기와 제목을 지정한다.
15
16         getContentPane().setBackground(Color.green);
17         //ContentPane의 초기 배경색을 초록색으로 지정한다.
18
19         getContentPane().addMouseMotionListener(new MouseAdapter() {
20             //ContentPane에 MouseMotionListener를 추가하고 MouseAdapter를 구현한다.
21             public void mouseDragged(MouseEvent e) {
22                 getContentPane().setBackground(Color.yellow);
23                 //마우스를 드래그하는 이벤트가 발생할 시 ContentPane의 배경색을 노란색으로 바꾼다.
24             }
25             public void mouseMoved(MouseEvent e) {
26                 getContentPane().setBackground(Color.green);
27                 //마우스를 클릭하지 않고 단순 이동하는 이벤트가 발생할 시
28                 //ContentPane의 배경색은 초록색으로 지정된다.
29             }
30         });
31
32         getContentPane().addMouseListener(new MouseAdapter() {
33             //ContentPane에 MouseListener를 추가하고 MouseAdapter를 구현한다.
34             public void mousePressed(MouseEvent e) {
35                 getContentPane().setBackground(Color.yellow);
36             }
37             //ContentPane 내에서 클릭 이벤트가 발생할 시 배경색을 노란색으로 바꾼다.
38             public void mouseReleased(MouseEvent e) {
39                 getContentPane().setBackground(Color.green);
40             }
41             //ContentPane 내에서 클릭 해제 이벤트가 발생할 시 배경색을 다시 초록색으로 바꾼다.
42         });
43
44         setVisible(true);
45         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
46     }
47
48     public static void main(String[] args) {
49         new MouseDraggingFrame();
50     }
51 }
52 }
53 }
```



-클릭이나 드래그가 이루어지지 않았을 때의 배경 색: 초록색.



-클릭이나 드래그가 이루어질 때의 배경 색: 노란색.

<실습 4> MoveStringFrame

-코드 및 실행결과

```
1  /*작성자: 임다희(2312282)
2   *작성일: 2024-11-08
3   *Lab8-4 MoveStringFrame*/
4
5  import java.awt.*;
6  import javax.swing.*;
7  import java.awt.event.*;
8  //GUI 구성 및 이벤트 처리에 필요한 패키지를 import한다.
9
10 public class MoveStringFrame extends JFrame {
11
12     private JLabel text; // 회전시킬 문자열을 내용으로 가지는 JLabel 컴포넌트.
13     private int count = 0; // 문자열 분할의 기준이 되는 정수타입의 변수 count
14
15     public MoveStringFrame() {
16         setTitle("Left 키로 문자열 회전");
17         setSize(400, 200);
18         // 프레임의 크기와 제목을 지정한다.
19
20         JPanel panel = new JPanel();
21
22         String T = "Love Java"; // 회전시킬 문자열의 내용은 Love Java이다.
23         text = new JLabel(T);
24
25         text.addMouseListener(new MouseAdapter() {
26             // text 컴포넌트에 MouseListener를 추가하고 MouseAdapter를 구현한다.
27             public void mouseClicked(MouseEvent e) {
28                 // text 컴포넌트가 클릭되었을 경우 다음과 같이 실행한다.
29                 if (count < T.length()) {
30                     // 변수 count의 크기가 문자열의 길이보다 작을 경우(회전이 완전히 진행되어 원본과 같아진 경우가 아니다)
31                     count++; // count를 1 증가시킨다.
32                     text.setText(T.substring(count, T.length()).concat(T.substring(0, count)));
33                     // text의 내용을 새로 설정한다.
34                     // 문자열 T의 count번째 문자부터 마지막 번째 문자(T.length()-1번째)까지로 이루어진 substring,
35                     // 0번째 문자부터 count-1번째 문자까지로 이루어진 substring을 concat으로 합친다.
36                     // 이 문자열을 text의 새로운 내용으로 설정해 실행 이전보다 한 문자만큼 왼쪽으로 밀린 문자열을 표시하도록 한다.
37                 }
38
39                 else { // 회전이 완전히 진행되어 원본과 같아진 경우(count==T.length())
40                     count = 1; // count를 1로 설정한다.
41                     text.setText(T.substring(count, T.length()).concat(T.substring(0, count)));
42                     // 앞의 방법과 똑같이 왼쪽으로 한 문자만큼 밀린 문자열을 출력한다.
43                     // count가 다시 T.length() 값과 같아질 때까지 앞의 if문 과정이 반복된다.
44                 }
45             }
46         });
47
48         panel.add(text);
49         add(panel);
50
51         setVisible(true);
52         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
53     }
54
55     public static void main(String[] args) {
56         new MoveStringFrame();
57     }
58 }
59 }
60 }
```




-처음 실행 시 표시되는 문자열. (count==0일 때의 상태)



-1회 클릭 시 표시되는 문자열. (count==1일 때의 상태)

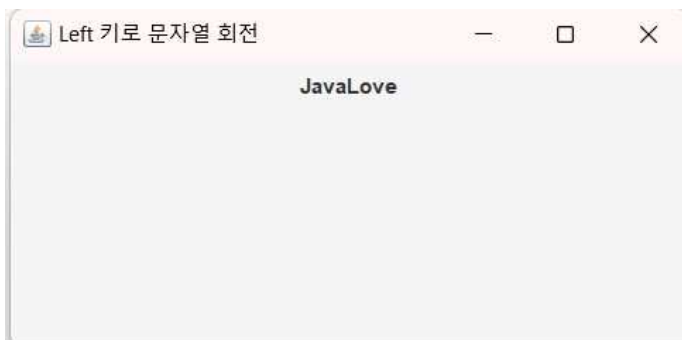


-2회 클릭 시 표시되는 문자열. (count==2일 때의 상태)



-3회 클릭 시 표시되는 문자열. (count==3일 때의 상태)

-4회 클릭 시 표시되는 문자열. (count==4일 때의 상태)



-5회 클릭 시 표시되는 문자열. (count==5일 때의 상태)



-6회 클릭 시 표시되는 문자열. (count==6일 때의 상태)



-7회 클릭 시 표시되는 문자열. (count==7일 때의 상태)



-8회 클릭 시 표시되는 문자열. (count==8일 때의 상태)



-9회 클릭 시 표시되는 문자열. (count==9==T.length()일 때의 상태)

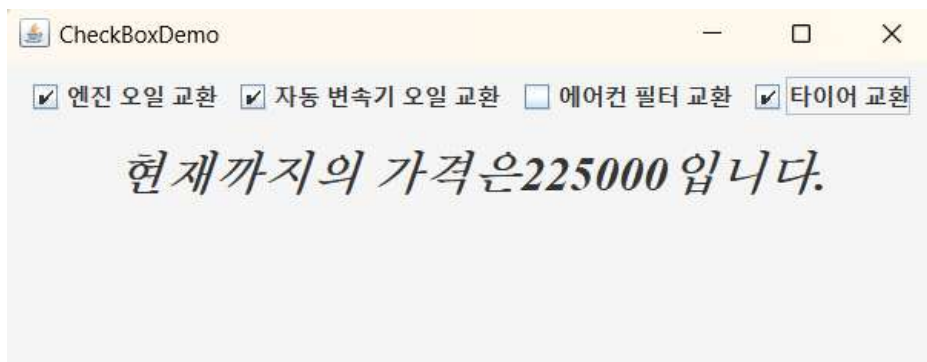
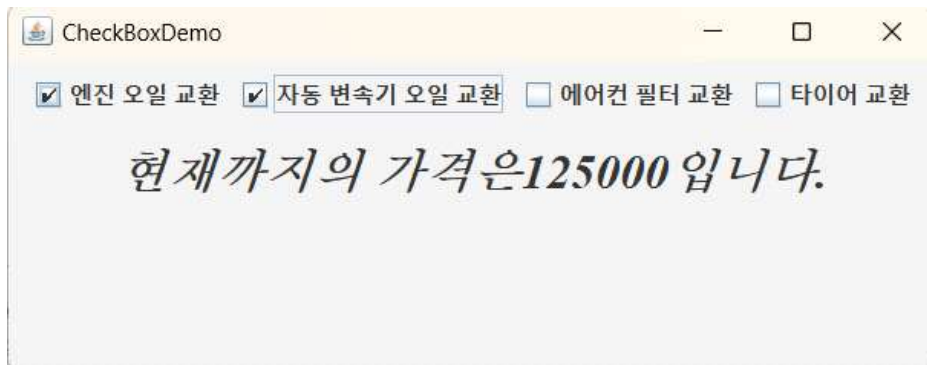
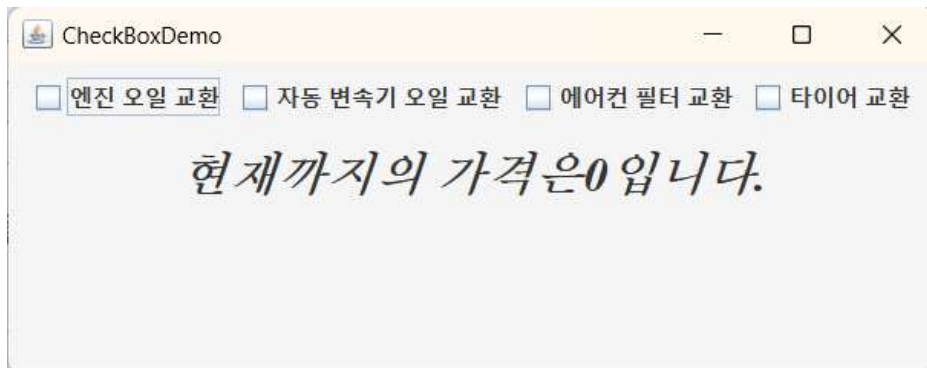


-10회 클릭 시 표시되는 문자열. (count==1인 상태로 돌아가며 이 이후로 동일하게 반복됨)

<실습 5> AutoRepair

-코드 및 실행결과

```
1 1. /* 작성자: 임다희(2312282)
2 2. *작성일: 2024-11-08
3 3. *Lab8-5 AutoRepair*/
4
5 5. import java.awt.*;
6 6. import javax.swing.*;
7 7. import java.awt.event.*;
8 8. //GUI 구성 및 이벤트 처리에 필요한 패키지를 import한다.
9 |
10 10. public class AutoRepair extends JFrame {
11 |
12 |     JCheckBox[] checkBoxes = new JCheckBox[4]; // JCheckBox 배열을 저장하는 크기 4의 배열 checkBoxes
13 |     int[] prices = new int[4]; // 크기 4의 정수 배열 prices. 각 수리 선택지에 해당하는 가격을 저장한다.
14 |     int total = 0; // 전체 가격 값을 저장하는 변수.
15 |
16 16. public AutoRepair() {
17 |     setTitle("CheckBoxDemo");
18 |     setSize(500, 200);
19 |     // 프레임의 크기와 제목을 지정한다.
20 |     setLayout(new BorderLayout()); // 프레임의 배치 관리자를 BorderLayout으로 지정한다.
21 |
22 |     JPanel panel1 = new JPanel(new FlowLayout(FlowLayout.CENTER));
23 |     // CheckBox들을 배치할 JPanel panel1. panel1에 배치되는 요소들은 가운데 정렬된 FlowLayout 형식으로 배치된다.
24 |     JPanel panel2 = new JPanel();
25 |     // 전체 가격값을 나타내는 JLabel을 배치할 JPanel panel2
26 |
27 |     JLabel text = new JLabel("현재까지의 가격은" + total + "입니다.");
28 |     text.setFont(new Font("Serif", Font.BOLD | Font.ITALIC, 25));
29 |     // 현재 전체 가격값을 나타내는 JLabel text. 글씨체의 크기와 스타일을 조절한다.
30 |
31 |     checkBoxes[0] = new JCheckBox("엔진 오일 교환");
32 |     checkBoxes[1] = new JCheckBox("자동 변속기 오일 교환");
33 |     checkBoxes[2] = new JCheckBox("에어컨 필터 교환");
34 |     checkBoxes[3] = new JCheckBox("타이어 교환");
35 |     // 서비스 항목을 나타내는 JCheckBox 4개를 만들어 checkBoxes의 원소로 저장한다.
36 |
37 |     prices[0] = 45000;
38 |     prices[1] = 80000;
39 |     prices[2] = 30000;
40 |     prices[3] = 100000;
41 |     // 서비스 항목에 해당하는 가격을 prices의 원소로 저장한다.
42 |
43 |     for (int i = 0; i < checkBoxes.length; i++) {
44 |         panel1.add(checkBoxes[i]);
45 |     } // panel1에 checkBoxes의 원소들을 차례로 추가한다.
46 |
47 |     panel2.add(text); // panel2에 text를 추가한다.
48 |
49 |     for (int j = 0; j < checkBoxes.length; j++) {
50 |         // checkBoxes의 모든 원소들에 대해 해당 작업을 똑같이 실행한다.
51 |         final int i = j;
52 52. checkBoxes[j].addItemListener(new ItemListener() {
53 |             // checkBoxes의 j번째 원소에 ItemListener를 추가하고 익명 클래스로 구현한다.
54 54. public void itemStateChanged(ItemEvent e) {
55 |             // 체크박스의 상태가 변하는 이벤트가 발생할 경우
56 |             if (e.getStateChange() == 1) { // 체크박스가 체크된 상태로 변화하는 이벤트일 경우
57 |                 total += prices[i]; // 전체 가격에 체크박스의 서비스 항목에 대응하는 가격을 더한다.
58 |             } else // 체크박스가 체크 해제된 상태로 변화하는 이벤트일 경우
59 |                 total -= prices[i]; // 전체 가격에서 체크박스의 서비스 항목에 대응하는 가격을 뺀다.
60 |             text.setText("현재까지의 가격은" + total + "입니다.");
61 |             // text의 내용을 변화된 total값을 반영해 다시 설정한다.
62 |         }
63 |     });
64 | }
65 |
66 | add(panel1, "North"); // panel1을 프레임의 North 위치에 배치한다.
67 | add(panel2, "Center"); // panel2를 프레임의 Center 위치에 배치한다.
68 |
69 | setVisible(true);
70 | setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
71 | }
72 |
73 73. public static void main(String[] args) {
74 |     new AutoRepair();
75 | }
76 |
77 }
```



<실습 6> RadioImageTest

-코드 및 실행결과

```
1  /*작성자: 임다희(2312282)
2   *작성일: 2024-11-09
3   *Lab8-6 RadioImageTest*/
4
5  import java.awt.*;
6  import javax.swing.*;
7  import java.awt.event.*;
8  import javax.swing.border.EmptyBorder;
9  //GUI 구성 및 이벤트 처리에 필요한 패키지를 import한다.
10
11 public class RadioImageTest extends JFrame {
12
13     JRadioButton[] buttons = new JRadioButton[5]; // JRadioButton 타입을 저장하는 크기 5의 배열 buttons
14
15     public RadioImageTest() {
16         setTitle("RadioButtonDemo");
17         setSize(550, 400); // 프레임의 크기와 제목을 지정한다.
18         setLayout(new BorderLayout()); // 프레임의 배치 관리자를 BorderLayout으로 지정한다.
19         String[] animals = { "bird", "cat", "dog", "rabbit", "pig" };
20         // 볼러를 사진 파일의 이름을 나타내는 문자열들을 저장하는 배열 animals.
21
22         buttons[0] = new JRadioButton(animals[0]);
23         buttons[1] = new JRadioButton(animals[1]);
24         buttons[2] = new JRadioButton(animals[2]);
25         buttons[3] = new JRadioButton(animals[3]);
26         buttons[4] = new JRadioButton(animals[4]);
27         // animals의 각 문자열을 내용으로 가지는 JRadioButton 5개를 생성해 buttons 배열에 저장한다.
28
29         ButtonGroup group = new ButtonGroup(); // 생성한 JRadioButton을 하나로 묶을 ButtonGroup group.
30
31         JPanel panel1 = new JPanel(new GridLayout(5, 1, 0, 10));
32         // JRadioButton들을 배치할 JPanel. 5행 1열의 GridLayout로 배치하며, 요소 간의 y축 간격을 10으로 지정한다.
33         JLabel label = new JLabel(); // 사진 파일을 볼러와 ImageIcon으로 붙일 JLabel 생성.
34
35         for (int i = 0; i < buttons.length; i++) {
36             // buttons 배열의 모든 JRadioButton 원소에 대해 다음과 같이 실행한다.
37             final int j = i;
38             group.add(buttons[j]); // 모든 JRadioButton을 ButtonGroup group에 추가한다.
39             panel1.add(buttons[j]); // 모든 JRadioButton을 panel1에 추가한다.
40             buttons[j].addActionListener(new ActionListener() {
41                 // 모든 JRadioButton에 ActionListener을 추가하고 ActionListener을 익명 클래스로 구현한다.
42                 public void actionPerformed(ActionEvent e) {
43                     // j번째 JRadioButton에서 액션 이벤트가 발생하면
44                     ImageIcon icon = new ImageIcon(animals[j] + ".png");
45                     label.setIcon(icon);
46                     //animals의 j번째 원소를 파일 이름으로 가지는 사진을 볼러와 ImageIcon으로 설정하고,
47                     //관찰자 둔 JLabel label의 아이콘으로 설정한다.
48                 }
49             });
50         }
51
52         add(panel1, "West"); //panel1을 프레임의 West 위치에 배치한다.
53         add(label, "Center"); //label를 프레임의 Center 위치에 배치한다.
54
55         panel1.setBorder(new EmptyBorder(0, 20, 0, 20));
56         //panel1의 외부 간격을 지정한다.
57
58         setVisible(true);
59         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
60     }
61
62     public static void main(String[] args) {
63         new RadioImageTest();
64     }
65 }
66
67
```

