

1811943 황성미 11일차 과제

▼ 7.1 AWT Event classes

event components(classes)들에 의해서 이벤트가 발생함

(ActionEvent, AdjustmentEvent, ComponentEvent, ContainerEvent, FocusEvent, ItemEvent, KeyEvent, MouseEvent, MouseMotionEvent, TextEvent, WindowEvent)

EventObject : Top-level class for event

ActionEvent : 버튼 클릭 object, TextArea 또는 TextField에 text input 가능, 더블 클릭, MenuItem object를 선택

ItemEvent : 체크박스, 리스트, 콤보박스, 라디오 객체로부터 아이템 선택

MouseEvent, MouseMotionEvent : 마우스 움직임

KeyEvent : 키보드

▼ 표로 정리한 이벤트

AWT Components	AWT Event types and Listener										
	action	adjustment	component	container	focus	item	key	mouse	mouse motion	text	window
Button	X		X		X		X	X	X		
Canvas			X		X		X	X	X		
Checkbox			X		X	X	X	X	X		
Checkbox MenuItem						X					
Choice			X		X	X	X	X	X		
Component			X		X		X	X	X		
Container			X	X	X		X	X	X		
Dialog			X	X	X		X	X	X		X
Frame			X	X	X		X	X	X		X
Label			X		X		X	X	X		

AWT Components	AWT Event Types and Listener										
	action	adjustment	component	container	focus	item	key	mouse	mouse motion	text	window
List	X		X		X	X	X	X	X		
MenuItem	X										
Panel			X	X	X		X	X	X		
Scrollbar		X	X		X		X	X	X		
ScrollPane			X	X	X		X	X	X		
TextArea			X		X		X	X	X	X	
TextComponent			X		X		X	X	X	X	
TextField	X		X		X		X	X	X	X	
Window			X	X	X		X	X	X		X

▼ Method

Listener Interface(상속받아서) 이를 구현해야함

ActionListener : actionPerformed(ActionEvent)

ItemListener : itemStateChanged(ItemEvent)

KeyListener : keyPressed(KeyEvent)

MouseListener : mouseClicked(MouseEvent), mousePressed(MouseEvent),
mouseReleased(MouseEvent)

MouseMotionListener : mouseDragged(MouseEvent),
mouseMoved(MouseEvent)

TextListener : textValueChanged(TextEvent)

WindowListener : windowClosed(WindowEvent)

▼ Swing Event

AWT와 비슷

Swing Component	Swing Event 및 Listener						
	action	caret	change	document, undoable edit	item	list selection	window
button	X		X		X		
check box	X		X		X		
color chooser			X				
combobox	X				X		
dialog							X
editor pane		X		X			
frame							X
internal frame							
list						X	
menu							
menu item	X		X		X		
file chooser	X						

Component	Swing 이벤트 및 Listener						
	action	caret	change	document, undoable edit	item	list selection	window
option pane							
password filed	X	X		X			
popupmenu							
progress bar			X				
radio button	X		X		X		
slider			X				
tabbed pane			X				
table						X	
text area		X		X			
text field	X	X		X			
text pane		X		X			
toggle button	X		X		X		

▼ Method

ActionListener : actionPerformed(ActionEvent)

ChangeListener : stateChanged(ChangeEvent)

ItemListener : itemStateChanged(ItemEvent)

KeyListener : keyPressed(KeyEvent)

MouseListener : mouseClicked(MouseEvent)

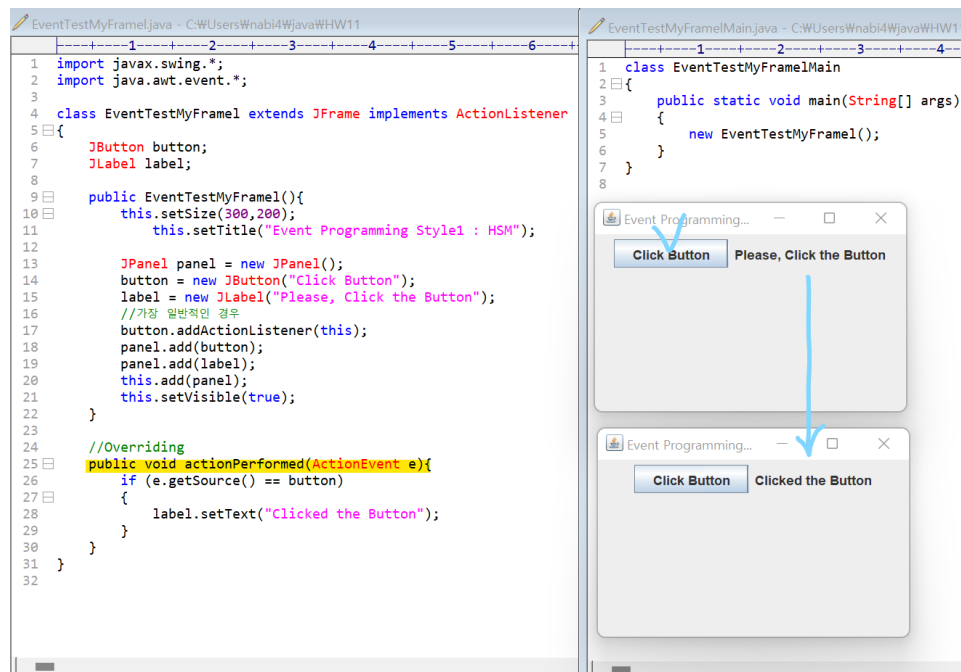
▼ 7.2 Event Programming

Handling Codes(인터페이스로 정의된 것을 구현하는 부분)

▼ 구현하는 방법 5가지

1. 상속받아서 구현하는 방법
- 2-3. inner 클래스로 구현하는 방법(이름이 있는 클래스 / 없는 클래스)
4. 이벤트를 처리하는 클래스 파일을 따로 만드는 방법
5. 자바의 람다 표현법을 이용해 방법3을 좀 더 편하게 구현한 방법

▼ Lab 1(상속받아서 구현)



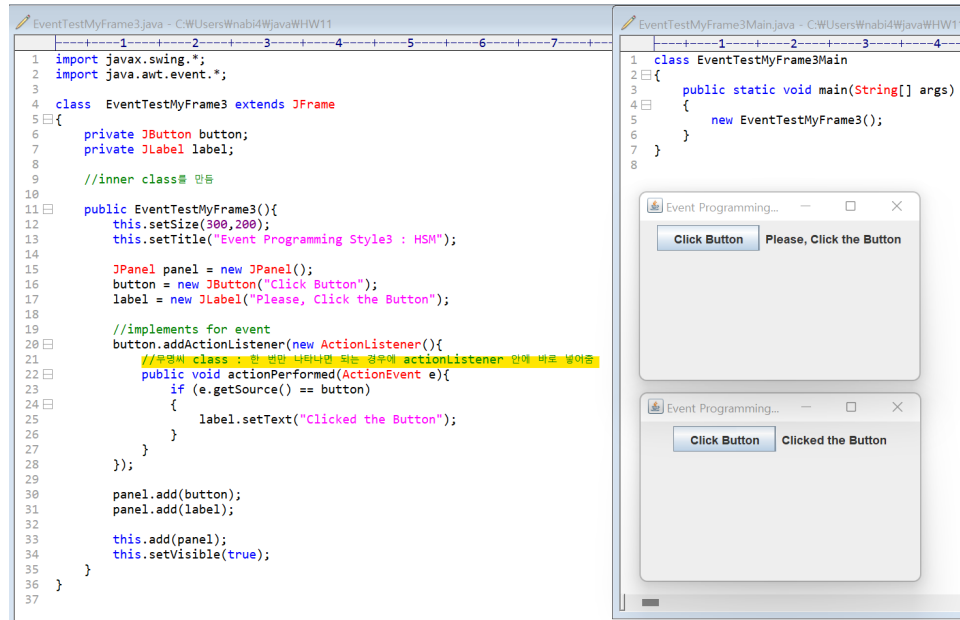
상속받아 이벤트 핸들링을 구현한 방법



actionPerformed(노란색 부분)을 구현하지 않았을 때 나타나는 메시지

단점 : 메서드가 많을 때 필요없는 메서드들도 다 null로 처리해주어야하는 불편함이 있음. 해당 코드가 어디에 쓰이는건지 한 눈에 파악하기 어려움.

▼ Lab3

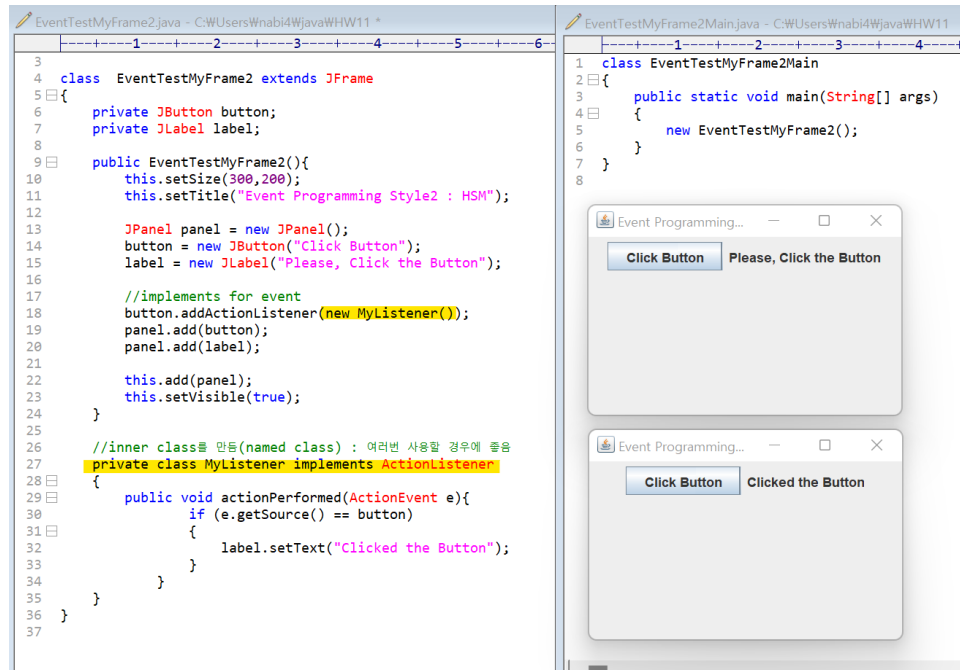


No-name inner class로 이벤트 핸들링을 구현한 방법

해당 이벤트가 발생할 때 실행시킬 함수를 바로 써주기 때문에 한 눈에 파악하기 쉬움

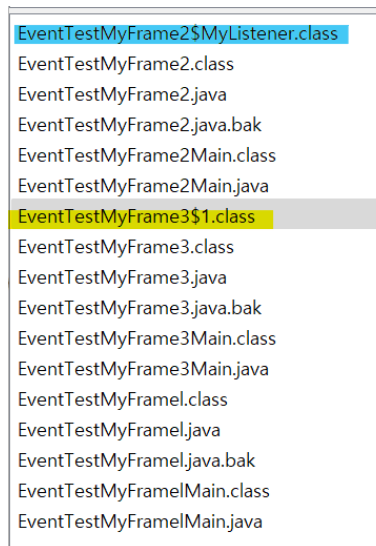
⇒ 똑같은 기능을 여러번 사용할 때, 무명씨로 하면 중복 코드가 많이 생김. 이런 경우에는 이름있는 클래스를 만들어 함수 이름을 호출시켜서 이벤트 핸들링을 하는 것이 더 편함(Lab2)

▼ Lab2



Named Inner class

ActionListener에 이름을 주어 함수 이름을 호출하는 것으로 이벤트를 발생시킬 수 있도록 이벤트 핸들링을 구현한 방법

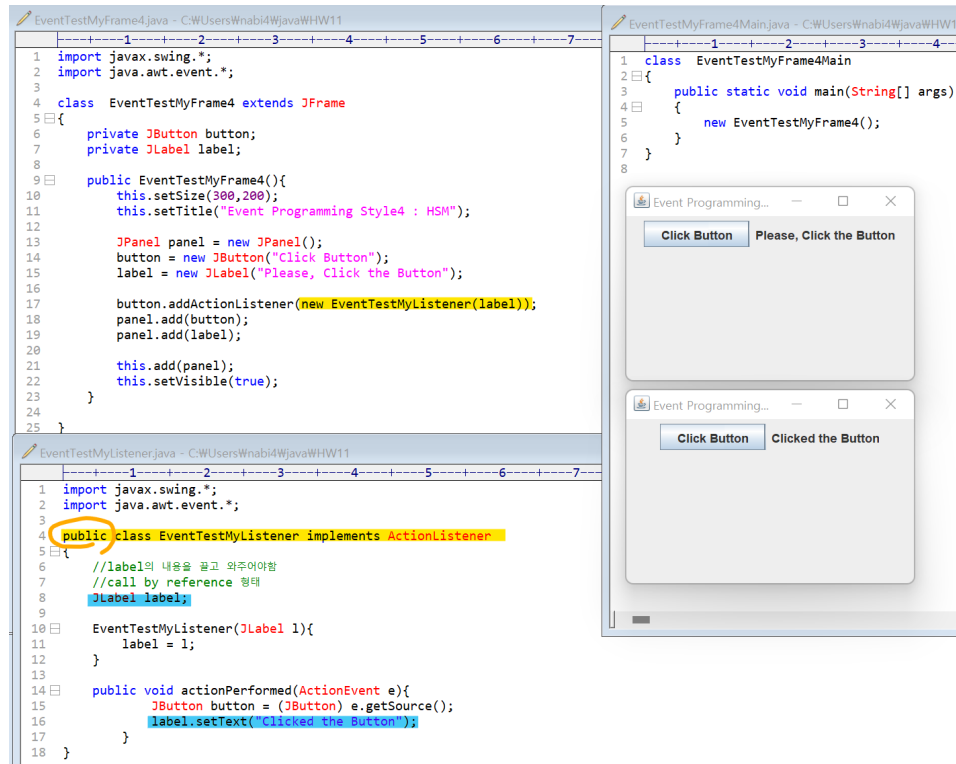


무명씨(파일 이름\$1), 이름있는 클래스(파일이름\$클래스이름)

▼ Lab4

이벤트를 처리하는 부분을 하나의 클래스 파일로 만듦

여러 클래스에서 해당 메서드를 사용할 수 있게 하기 위함



Independent class

여러 클래스에서 사용할 수 있어야 하므로 public으로 선언해주어야 함
 이때 해당 이벤트가 처리될 부분(위 예제에선 JLabel)을 call by reference 형식으로 불러와서 코드를 구현해야함

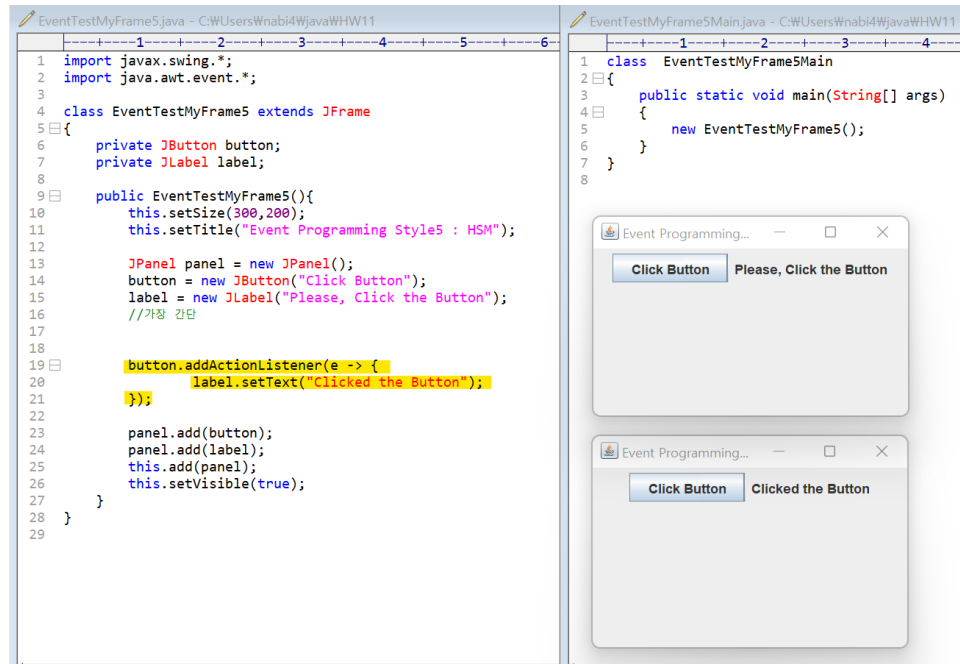
▼ 7.2.5 Lambda Expression

value만 보내주면 코드가 수행될 수 있도록 하는 자바의 lambda 표현 특징을 이용.

(argument) → {body}

method invocation의 원리를 이용

▼ Lab5



argument가 어떻게 처리될 것인지 다 아는 경우 이 방법을 이용하면 편리함

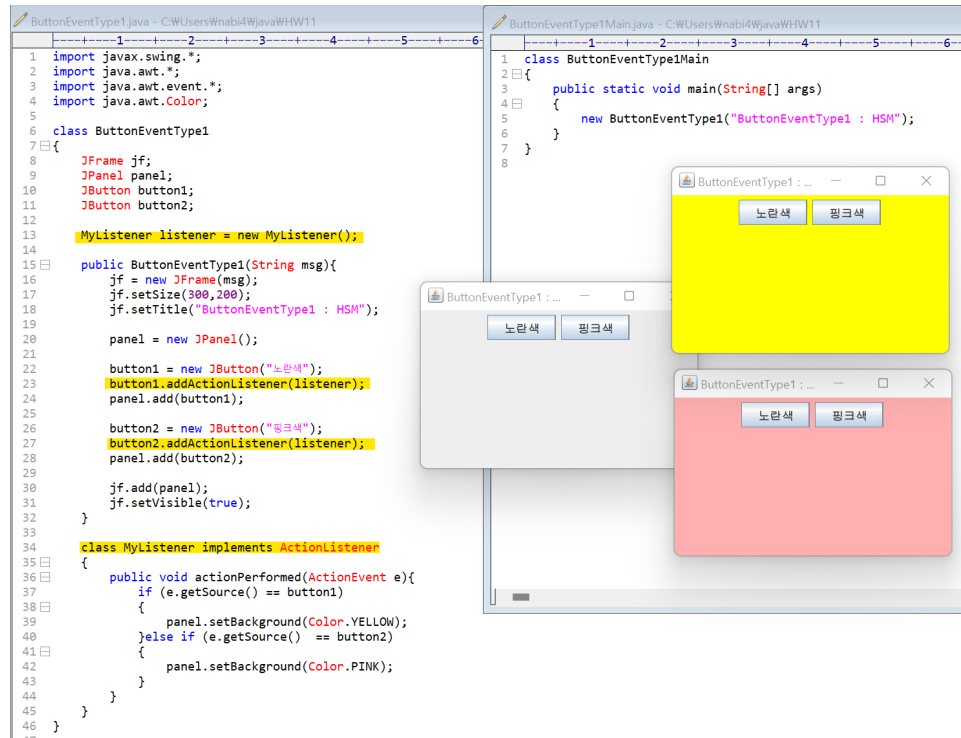
▼ 7.3 Event Types

사용자가 버튼을 클릭하는 경우, 사용자가 메뉴 항목을 선택하는 경우, 사용자가 Mouse를 사용하는 경우, 사용자가 특정 key를 사용하는 경우, 사용자가 텍스트필드에서 엔터키를 누른 경우 등 여러 상황의 이벤트가 발생

▼ 7.3.1 Button Event Types

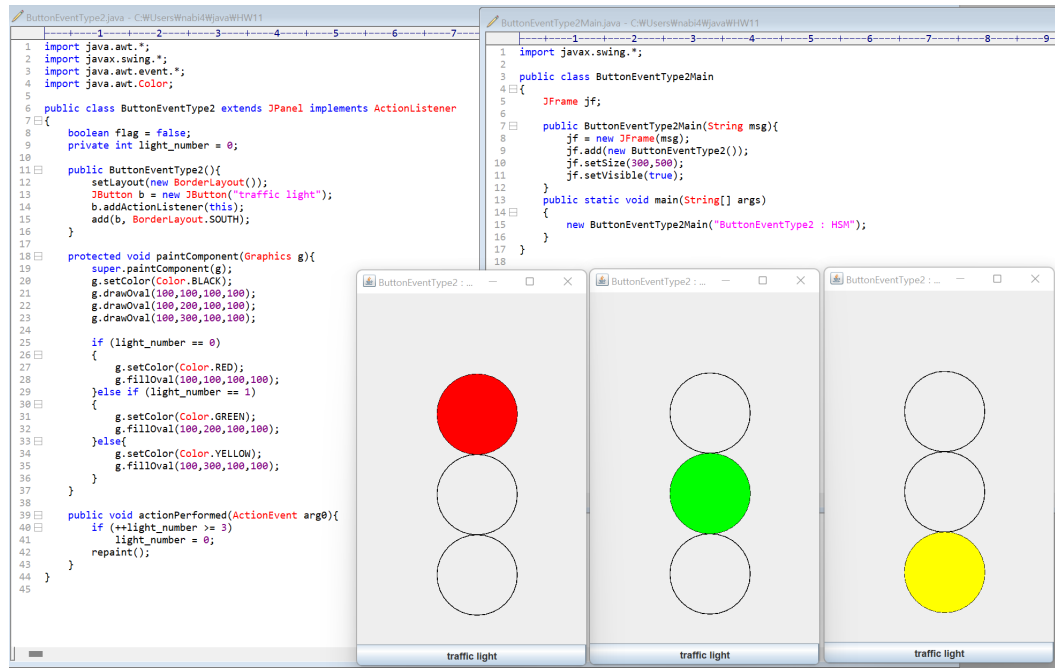
addActionListener() 메서드 불러준 후, 이 인터페이스를 구현하기 위해 actionPerformed() 가 반드시 선언되어야 함

▼ Lab1



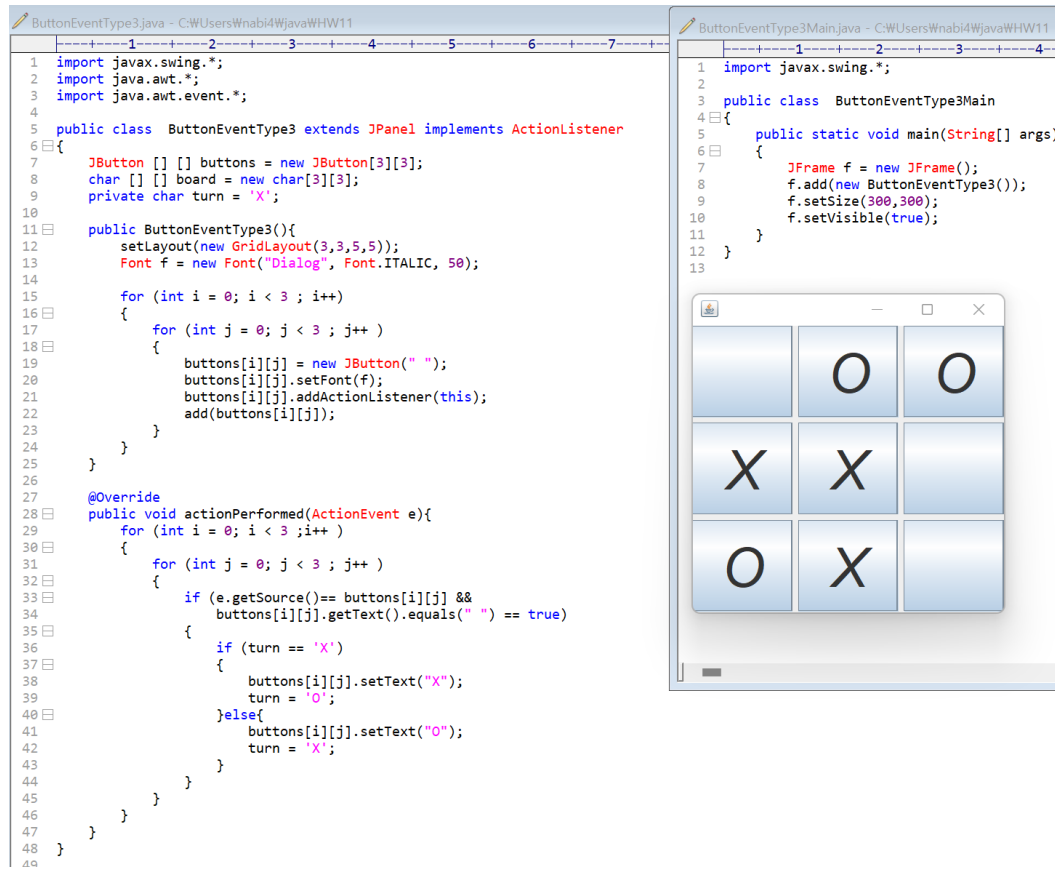
같은 것을 쓰기 때문에 위에 한 번만 new 객체만 생성하고 addActionListener에서 listener을 불러주
었음. MyListener에서 이벤트 핸들링 부분을 구현해주면 버튼을 클릭할 때 마다 정의한 이벤트가 발생
함

▼ Lab2



화면을 구성한 후, light_number의 숫자에 따라 색상이 바뀌도록 코드를 구현함. actionPerformed에
는 버튼을 클릭할 때마다 숫자를 1씩 증가시키고 3이상이 되면 0으로 초기화해줌

▼ Lab3



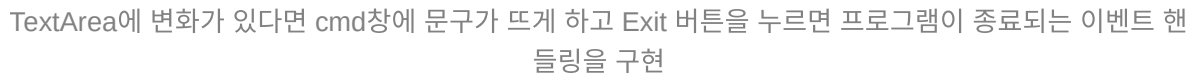
GridLayout(int rows, int cols, int hgap, int vgap) : 지정된 수의 행과 열, 레이아웃 간 여백

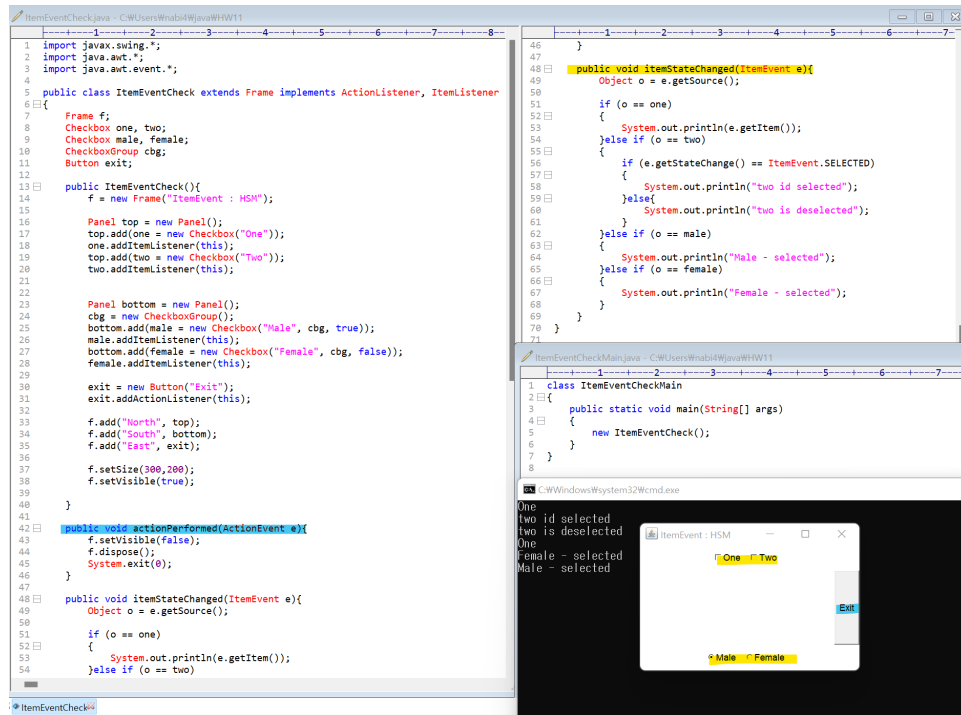
(위 코드에서 board 배열도 필요없음!)

처음에 버튼을 클릭할 때 X가 나오게 한 후, X가 나오면 그 이후엔 O, O가 나오면 그 이후엔 X가 뜨게 하는 이벤트 핸들링

buttons[i][j].getText().equals(" ") == true : 클릭하지 않은 버튼만 클릭할 수 있게 함

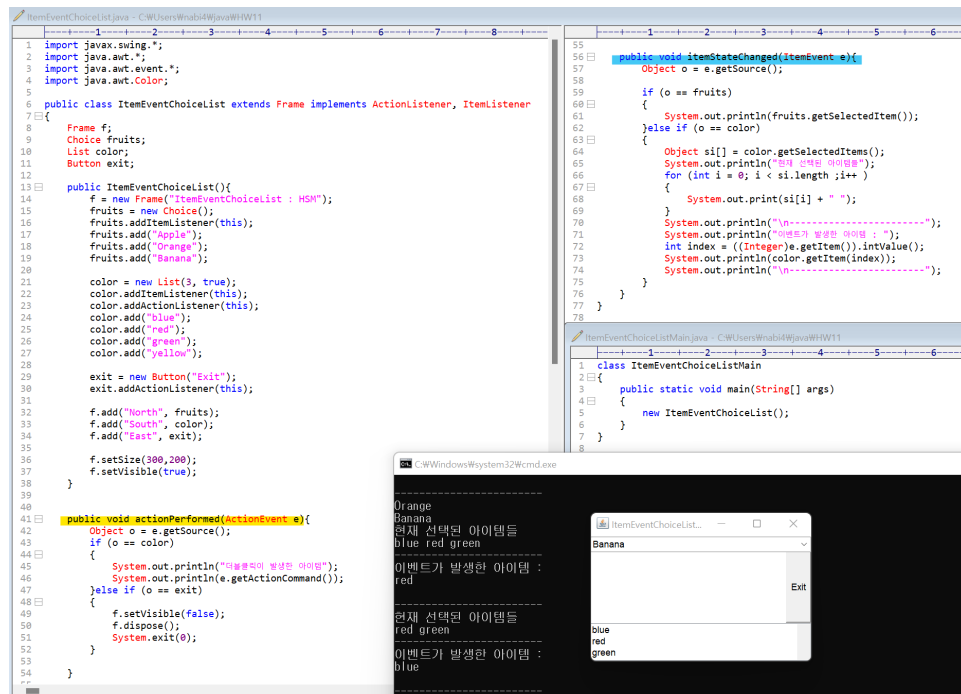
▼ 7.3.2 Text Event Types





화면을 먼저 구성해준 후, 해당 버튼들이 클릭될 때 실행할 이벤트를 구현해줌

▼ Lab2

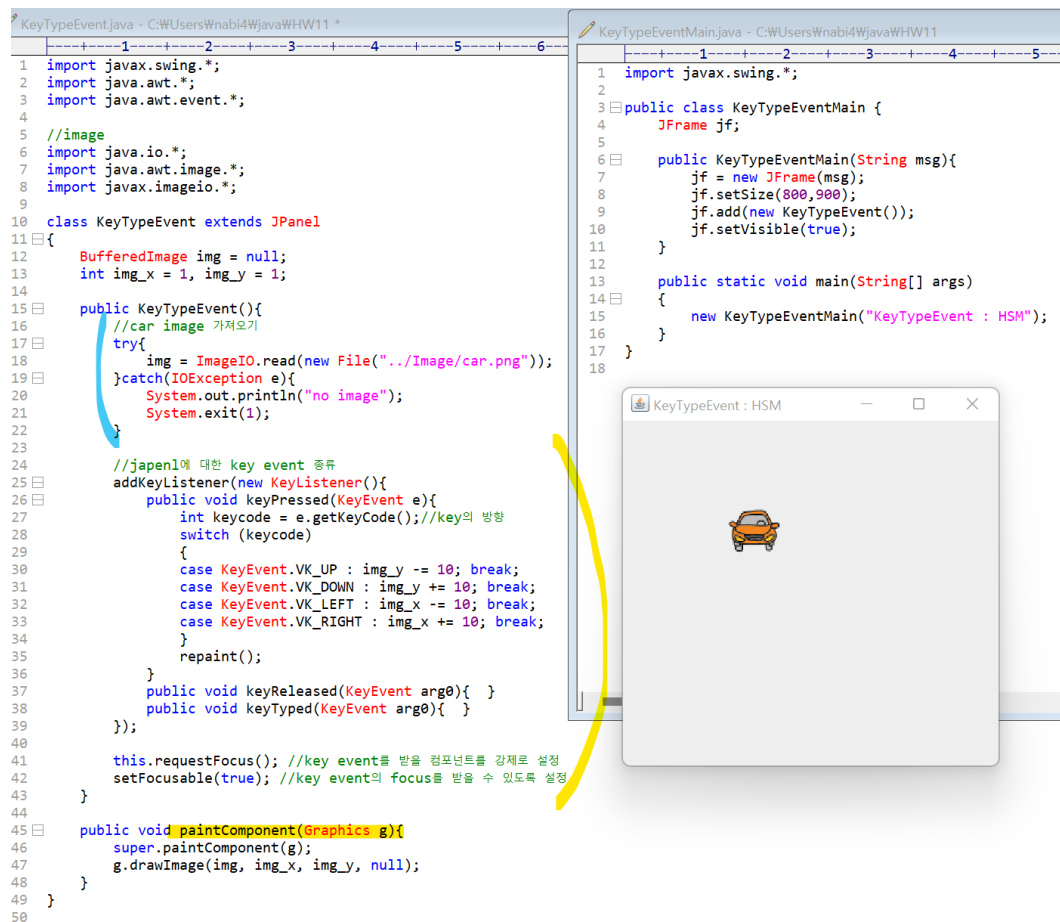


▼ 7.3.4 Key Event

▼ Method

keyTyped(KeyEvent e), keyPressed(KeyEvent e), keyReleased(KeyEvent e)

▼ Lab1



키보드를 이용해서 불러온 이미지를 움직이게끔 하는 이벤트

*drawImage(Image img, int x, int y, ImageObserver observer);

▼ 7.3.5 Mouse Event

<https://diyall.tistory.com/608>

Mouse Event method

▼ Method

mouseEntered : 마우스가 컴포넌트 위에 올라갈 때

mouseExited : 마우스가 컴포넌트에서 내려올 때

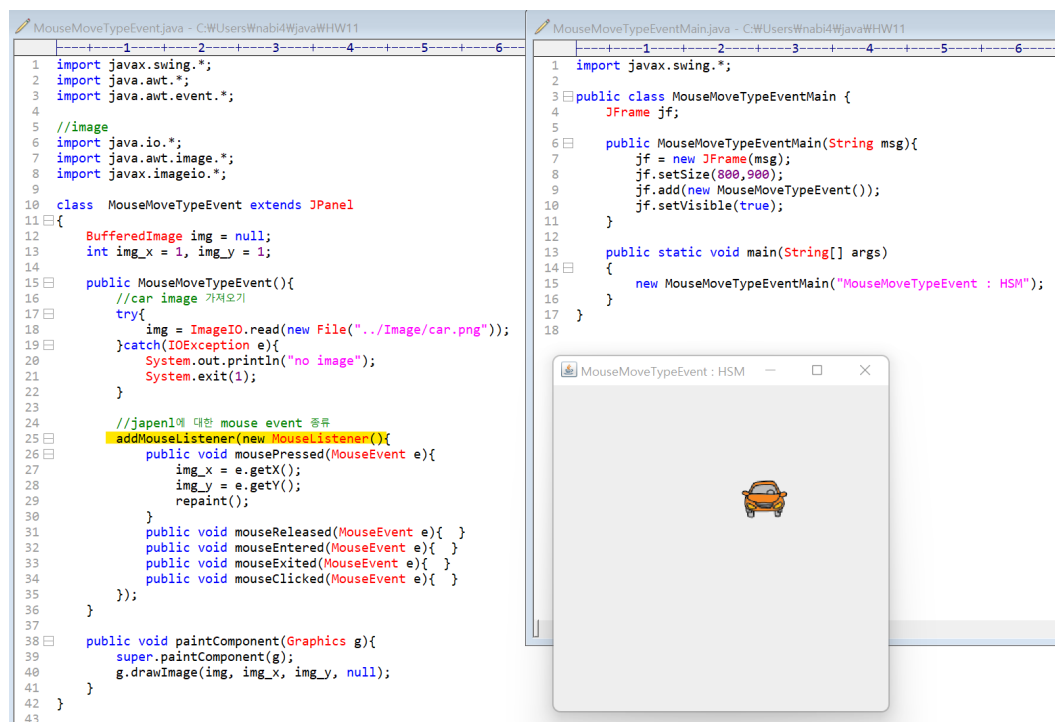
mouseReleased : 눌러진 버튼이 떼어질 때

mouseClicked : 마우스가 컴포넌트를 클릭했을 때

mouseDragged : 마우스가 드래그되는 동안

mousePressed : 마우스 버튼이 눌러졌을 때

▼ Lab1

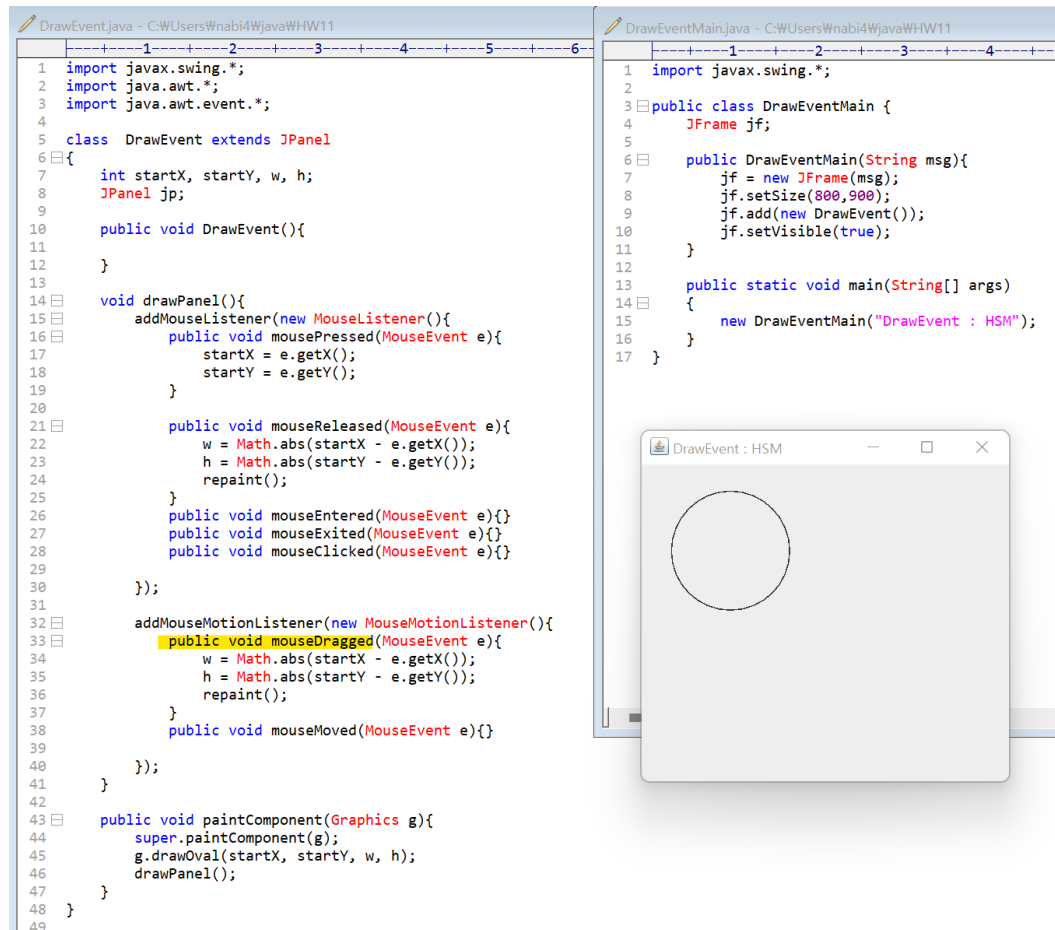


```
MouseMoveTypeEvent.java - C:\Users\Wnabi4\java\HW11
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4
5 //image
6 import java.io.*;
7 import java.awt.image.*;
8 import javax.imageio.*;
9
10 class MouseMoveTypeEvent extends JPanel
11 {
12     BufferedImage img = null;
13     int img_x = 1, img_y = 1;
14
15     public MouseMoveTypeEvent(){
16         //car image 가져오기
17         try{
18             img = ImageIO.read(new File("../Image/car.png"));
19         }catch(IOException e){
20             System.out.println("no image");
21             System.exit(1);
22         }
23
24         //japenl에 대한 mouse event 종류
25         addMouseListener(new MouseListener(){
26             public void mousePressed(MouseEvent e){
27                 img_x = e.getX();
28                 img_y = e.getY();
29                 repaint();
30             }
31             public void mouseReleased(MouseEvent e){ }
32             public void mouseEntered(MouseEvent e){ }
33             public void mouseExited(MouseEvent e){ }
34             public void mouseClicked(MouseEvent e){ }
35         });
36     }
37
38     public void paintComponent(Graphics g){
39         super.paintComponent(g);
40         g.drawImage(img, img_x, img_y, null);
41     }
42 }
43
```

```
MouseMoveTypeEventMain.java - C:\Users\Wnabi4\java\HW11
1 import javax.swing.*;
2
3 public class MouseMoveTypeEventMain {
4     JFrame jf;
5
6     public MouseMoveTypeEventMain(String msg){
7         jf = new JFrame(msg);
8         jf.setSize(800,900);
9         jf.add(new MouseMoveTypeEvent());
10        jf.setVisible(true);
11    }
12
13    public static void main(String[] args)
14    {
15        new MouseMoveTypeEventMain("MouseMoveTypeEvent : HSM");
16    }
17 }
18
```

마우스를 클릭할 때마다 그 위치에 불러온 이미지가 이동하게끔 하는 이벤트
마우스 버튼이 눌러졌을 때 이벤트가 발생함

▼ Lab2(Mouse Motion event)



마우스를 드래그하여 원하는 크기의 도형을 그릴 수 있음
mouseDragged 메서드 이용

▼ HW7(나만의 계산기)

: 화면을 구성함과 동시에 addActionListener를 달아주는 형태로 코드 구현

각 버튼에 따라 실행될 이벤트는 non-named class와 재사용될 이벤트는 named class를 사용해 구현했음

```

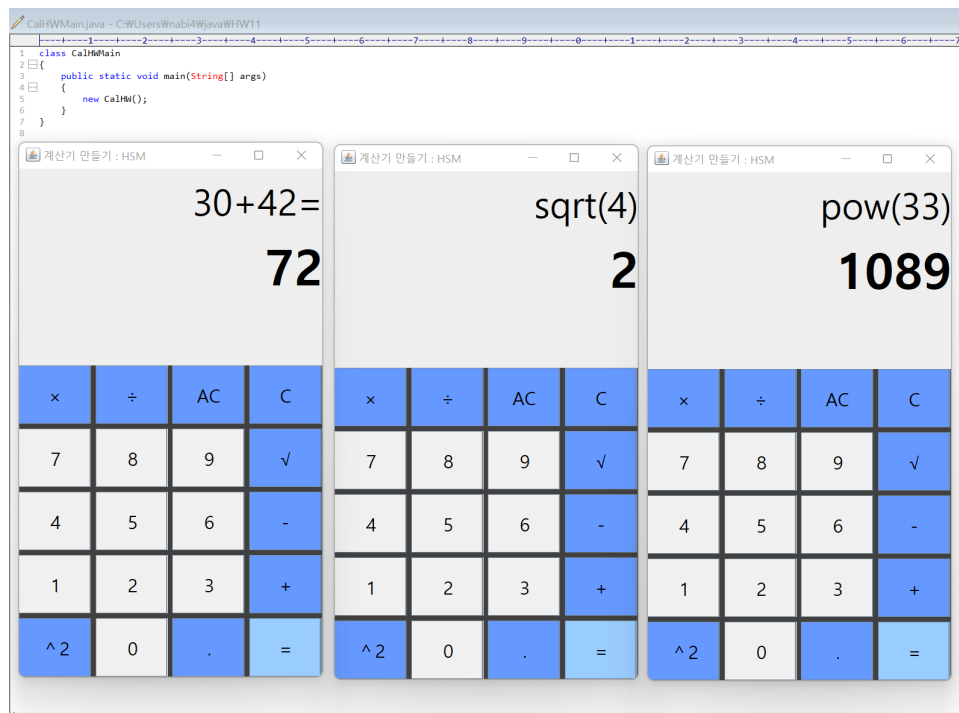
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4
5 public class CalHW extends JFrame {
6
7     JLabel label;
8     JLabel info;
9     double result = 0;
10    String math = "";
11    double num = 0;
12    String tostring = "";
13
14    public CalHW() {
15        setTitle("계산기 만들기 : HSM");
16
17        setLayout(new BorderLayout(5, 5));
18
19        //입력 값과 결과값이 출력되는 위 패널 생성 후 콘테이너에 추가
20        UpPanel UP = new UpPanel();
21        add(UP, BorderLayout.EAST);
22
23        //버튼을 넣을 아래 패널 생성 후 콘테이너에 추가
24        DownPanel DP = new DownPanel();
25        add(DP, BorderLayout.SOUTH);
26
27        setSize(350, 600);
28        setVisible(true);
29    }
30
31    //화면 구성
32    //사용자가 누른 값을 보여주는 화면
33    class UpPanel extends JPanel {
34        public UpPanel() {
35            setLayout(new GridLayout(3, 1));
36            info = new JLabel("");
37            label = new JLabel("0");
38
39            info.setFont(new Font("맑은 고딕", 0, 40));
40            info.setForeground(Color.BLACK);
41            info.setHorizontalAlignment(JLabel.RIGHT);
42
43            label.setFont(new Font("맑은 고딕", Font.BOLD, 55));
44            label.setForeground(Color.BLACK);
45            label.setHorizontalAlignment(JLabel.RIGHT);
46
47            add(info);
48            add(label);
49        }
50    }
51
52    //숫자, 기호 보여주는 화면
53    class DownPanel extends JPanel {
54        public DownPanel() {
55            JButton[] bt = new JButton[20];
56            setLayout(new GridLayout(5, 4, 5, 5));
57            setBackground(Color.DARK_GRAY);
58
59            bt[0] = new JButton("x");
60            bt[1] = new JButton("+");
61            bt[2] = new JButton("AC");
62            bt[3] = new JButton("C");
63
64            bt[4] = new JButton("7");
65            bt[5] = new JButton("8");
66            bt[6] = new JButton("9");
67            bt[7] = new JButton("v");
68
69            bt[8] = new JButton("4");
70            bt[9] = new JButton("5");
71            bt[10] = new JButton("6");
72            bt[11] = new JButton("-");
73
74            bt[12] = new JButton("1");
75            bt[13] = new JButton("2");
76            bt[14] = new JButton("3");
77            bt[15] = new JButton("=");
78
79            bt[16] = new JButton("^ 2");
80            bt[17] = new JButton("0");
81            bt[18] = new JButton(".");
82            bt[19] = new JButton("/");
83
84            for(int i = 0; i < 19; i++) {
85
86                //화면 구성
87                bt[i].setPreferredSize(new Dimension(0, 65));
88                bt[i].setFont(new Font("맑은 고딕", 0, 20));
89                bt[i].setForeground(Color.BLACK);
90                bt[i].setBackground(new Color(240, 240, 240));
91
92                add(bt[i]);
93
94                //이벤트 핸들링(non-named class & named class이름)
95                if(i < 15 && i < 15 && i % 4 != 3) { //숫자 버튼
96                    bt[i].addActionListener(new ActionListener() {
97                        //non-name class event
98                        @Override
99                        public void actionPerformed(ActionEvent e) {
100
101                            JButton b = (JButton)e.getSource();
102                            if(label.getText() == "0") {
103                                label.setText("");
104                            }
105                            String labeltext = label.getText();
106                            String text = b.getText();
107                            String newtext = labeltext + text;
108
109                            for(int i = 0; i < 19; i++) {
110
111                                //화면 구성
112                                bt[i].setPreferredSize(new Dimension(0, 65));
113                                bt[i].setFont(new Font("맑은 고딕", 0, 20));
114                                bt[i].setForeground(Color.BLACK);
115                                bt[i].setBackground(new Color(240, 240, 240));
116
117                                add(bt[i]);
118
119                                //이벤트 핸들링(non-named class & named class이름)
120                                if(i < 15 && i < 15 && i % 4 != 3) { //숫자 버튼
121                                    bt[i].addActionListener(new ActionListener() {
122                                        //non-name class event
123                                        @Override
124                                        public void actionPerformed(ActionEvent e) {
125
126                                            JButton b = (JButton)e.getSource();
127                                            if(label.getText() == "0") {
128                                                label.setText("");
129                                            }
130                                            String labeltext = label.getText();
131                                            String text = b.getText();
132                                            String newtext = labeltext + text;
133                                            int n = newtext.length();
134                                            if(n < 10) {
135                                                label.setText(newtext);
136                                            }
137                                            if(info.getText().contains("/") == true) {
138                                                info.setText("");
139                                                label.setText(b.getText());
140                                                result = 0;
141                                                num = 0;
142                                            }
143                                        }
144                                    });
145                                }
146                            }
147                        }
148                    });
149                }
150            }
151
152            //계산 결과
153            bt[16].setPreferredSize(new Dimension(150, 204, 255));
154            bt[16].addActionListener(new ActionListener() {
155                //non-name class event
156                @Override
157                public void actionPerformed(ActionEvent e) {
158                    int n = label.getText().length();
159                    if(n == 0) {
160                        label.setText("0");
161                        info.setText("");
162                    }
163                    result = 0;
164                }
165            });
166
167            //계산기 버튼
168            bt[17].setPreferredSize(new Dimension(150, 204, 255));
169            bt[17].addActionListener(new ActionListener() {
170                //non-name class event
171                @Override
172                public void actionPerformed(ActionEvent e) {
173                    int n = label.getText().length();
174                    if(n > 0) {
175                        setBackspace(label.getText().length() - 1);
176                    }
177                    if(label.getText() == "") {
178                        label.setText("0");
179                    }
180                }
181            });
182
183            //결과 버튼
184            bt[18].setPreferredSize(new Dimension(150, 204, 255));
185            bt[18].addActionListener(new ActionListener() {
186                //non-name class event
187                @Override
188                public void actionPerformed(ActionEvent e) {
189                    String labeltext = label.getText();
190                    String text = b.getText();
191                    String newtext = labeltext + text;
192                    int n = newtext.length();
193                    if(label.getText().contains(b.getText()) == false && n < 10) {
194                        label.setText(newtext);
195                    }
196                }
197            });
198
199            //결과 버튼
200            bt[19].setPreferredSize(new Dimension(150, 204, 255));
201            bt[19].addActionListener(new ActionListener() {
202                //non-name class event
203                @Override
204                public void actionPerformed(ActionEvent e) {
205                    String labeltext = label.getText();
206                    String text = b.getText();
207                    String newtext = labeltext + text;
208                    int n = newtext.length();
209                    if(label.getText().contains(b.getText()) == false && n < 10) {
210                        label.setText(newtext);
211                    }
212                }
213            });
214        }
215    }
216
217    private void setBackspace(int length) {
218        String labeltext = label.getText();
219        String text = labeltext.substring(0, length);
220        label.setText(text);
221    }
222
223    private void setResult() {
224        String labeltext = label.getText();
225        String text = labeltext;
226        String newtext = labeltext + text;
227        int n = newtext.length();
228        if(label.getText().contains("/") == true && n < 10) {
229            label.setText(newtext);
230            result = Double.parseDouble(newtext);
231            num = 0;
232        }
233    }
234
235    private void setInfo() {
236        String labeltext = label.getText();
237        String text = labeltext;
238        String newtext = labeltext + text;
239        int n = newtext.length();
240        if(label.getText().contains("/") == true && n < 10) {
241            label.setText(newtext);
242            info.setText(newtext);
243        }
244    }
245
246    private void setNum() {
247        String labeltext = label.getText();
248        String text = labeltext;
249        String newtext = labeltext + text;
250        int n = newtext.length();
251        if(label.getText().contains(".") == true && n < 10) {
252            label.setText(newtext);
253        }
254    }
255
256    private void setClear() {
257        String labeltext = label.getText();
258        String text = labeltext;
259        String newtext = labeltext + text;
260        int n = newtext.length();
261        if(label.getText().contains("/") == true && n < 10) {
262            label.setText(newtext);
263            info.setText("");
264            result = 0;
265            num = 0;
266        }
267    }
268
269    private void setCancel() {
270        String labeltext = label.getText();
271        String text = labeltext;
272        String newtext = labeltext + text;
273        int n = newtext.length();
274        if(label.getText().contains("/") == true && n < 10) {
275            label.setText(newtext);
276            info.setText("");
277            result = 0;
278            num = 0;
279        }
280    }
281
282    private void setPower() {
283        String labeltext = label.getText();
284        String text = labeltext;
285        String newtext = labeltext + text;
286        int n = newtext.length();
287        if(label.getText().contains("/") == true && n < 10) {
288            label.setText(newtext);
289            info.setText("");
290            result = 0;
291            num = 0;
292        }
293    }
294
295    private void setSquare() {
296        String labeltext = label.getText();
297        String text = labeltext;
298        String newtext = labeltext + text;
299        int n = newtext.length();
300        if(label.getText().contains("/") == true && n < 10) {
301            label.setText(newtext);
302            info.setText("");
303            result = 0;
304            num = 0;
305        }
306    }
307
308    private void setRoot() {
309        String labeltext = label.getText();
310        String text = labeltext;
311        String newtext = labeltext + text;
312        int n = newtext.length();
313        if(label.getText().contains("/") == true && n < 10) {
314            label.setText(newtext);
315            info.setText("");
316            result = 0;
317            num = 0;
318        }
319    }
320
321    private void setMod() {
322        String labeltext = label.getText();
323        String text = labeltext;
324        String newtext = labeltext + text;
325        int n = newtext.length();
326        if(label.getText().contains("/") == true && n < 10) {
327            label.setText(newtext);
328            info.setText("");
329            result = 0;
330            num = 0;
331        }
332    }
333
334    private void setDiv() {
335        String labeltext = label.getText();
336        String text = labeltext;
337        String newtext = labeltext + text;
338        int n = newtext.length();
339        if(label.getText().contains("/") == true && n < 10) {
340            label.setText(newtext);
341            info.setText("");
342            result = 0;
343            num = 0;
344        }
345    }
346
347    private void setMul() {
348        String labeltext = label.getText();
349        String text = labeltext;
350        String newtext = labeltext + text;
351        int n = newtext.length();
352        if(label.getText().contains("/") == true && n < 10) {
353            label.setText(newtext);
354            info.setText("");
355            result = 0;
356            num = 0;
357        }
358    }
359
360    private void setAdd() {
361        String labeltext = label.getText();
362        String text = labeltext;
363        String newtext = labeltext + text;
364        int n = newtext.length();
365        if(label.getText().contains("/") == true && n < 10) {
366            label.setText(newtext);
367            info.setText("");
368            result = 0;
369            num = 0;
370        }
371    }
372
373    private void setSub() {
374        String labeltext = label.getText();
375        String text = labeltext;
376        String newtext = labeltext + text;
377        int n = newtext.length();
378        if(label.getText().contains("/") == true && n < 10) {
379            label.setText(newtext);
380            info.setText("");
381            result = 0;
382            num = 0;
383        }
384    }
385
386    private void setExp() {
387        String labeltext = label.getText();
388        String text = labeltext;
389        String newtext = labeltext + text;
390        int n = newtext.length();
391        if(label.getText().contains("/") == true && n < 10) {
392            label.setText(newtext);
393            info.setText("");
394            result = 0;
395            num = 0;
396        }
397    }
398
399    private void setLog() {
400        String labeltext = label.getText();
401        String text = labeltext;
402        String newtext = labeltext + text;
403        int n = newtext.length();
404        if(label.getText().contains("/") == true && n < 10) {
405            label.setText(newtext);
406            info.setText("");
407            result = 0;
408            num = 0;
409        }
410    }
411
412    private void setLog10() {
413        String labeltext = label.getText();
414        String text = labeltext;
415        String newtext = labeltext + text;
416        int n = newtext.length();
417        if(label.getText().contains("/") == true && n < 10) {
418            label.setText(newtext);
419            info.setText("");
420            result = 0;
421            num = 0;
422        }
423    }
424
425    private void setSqrt() {
426        String labeltext = label.getText();
427        String text = labeltext;
428        String newtext = labeltext + text;
429        int n = newtext.length();
430        if(label.getText().contains("/") == true && n < 10) {
431            label.setText(newtext);
432            info.setText("");
433            result = 0;
434            num = 0;
435        }
436    }
437
438    private void setSin() {
439        String labeltext = label.getText();
440        String text = labeltext;
441        String newtext = labeltext + text;
442        int n = newtext.length();
443        if(label.getText().contains("/") == true && n < 10) {
444            label.setText(newtext);
445            info.setText("");
446            result = 0;
447            num = 0;
448        }
449    }
450
451    private void setCos() {
452        String labeltext = label.getText();
453        String text = labeltext;
454        String newtext = labeltext + text;
455        int n = newtext.length();
456        if(label.getText().contains("/") == true && n < 10) {
457            label.setText(newtext);
458            info.setText("");
459            result = 0;
460            num = 0;
461        }
462    }
463
464    private void setTan() {
465        String labeltext = label.getText();
466        String text = labeltext;
467        String newtext = labeltext + text;
468        int n = newtext.length();
469        if(label.getText().contains("/") == true && n < 10) {
470            label.setText(newtext);
471            info.setText("");
472            result = 0;
473            num = 0;
474        }
475    }
476
477    private void setCot() {
478        String labeltext = label.getText();
479        String text = labeltext;
480        String newtext = labeltext + text;
481        int n = newtext.length();
482        if(label.getText().contains("/") == true && n < 10) {
483            label.setText(newtext);
484            info.setText("");
485            result = 0;
486            num = 0;
487        }
488    }
489
490    private void setSec() {
491        String labeltext = label.getText();
492        String text = labeltext;
493        String newtext = labeltext + text;
494        int n = newtext.length();
495        if(label.getText().contains("/") == true && n < 10) {
496            label.setText(newtext);
497            info.setText("");
498            result = 0;
499            num = 0;
500        }
501    }
502
503    private void setCsc() {
504        String labeltext = label.getText();
505        String text = labeltext;
506        String newtext = labeltext + text;
507        int n = newtext.length();
508        if(label.getText().contains("/") == true && n < 10) {
509            label.setText(newtext);
510            info.setText("");
511            result = 0;
512            num = 0;
513        }
514    }
515
516    private void setArcSin() {
517        String labeltext = label.getText();
518        String text = labeltext;
519        String newtext = labeltext + text;
520        int n = newtext.length();
521        if(label.getText().contains("/") == true && n < 10) {
522            label.setText(newtext);
523            info.setText("");
524            result = 0;
525            num = 0;
526        }
527    }
528
529    private void setArcCos() {
530        String labeltext = label.getText();
531        String text = labeltext;
532        String newtext = labeltext + text;
533        int n = newtext.length();
534        if(label.getText().contains("/") == true && n < 10) {
535            label.setText(newtext);
536            info.setText("");
537            result = 0;
538            num = 0;
539        }
540    }
541
542    private void setArcTan() {
543        String labeltext = label.getText();
544        String text = labeltext;
545        String newtext = labeltext + text;
546        int n = newtext.length();
547        if(label.getText().contains("/") == true && n < 10) {
548            label.setText(newtext);
549            info.setText("");
550            result = 0;
551            num = 0;
552        }
553    }
554
555    private void setArcCot() {
556        String labeltext = label.getText();
557        String text = labeltext;
558        String newtext = labeltext + text;
559        int n = newtext.length();
560        if(label.getText().contains("/") == true && n < 10) {
561            label.setText(newtext);
562            info.setText("");
563            result = 0;
564            num = 0;
565        }
566    }
567
568    private void setArcSec() {
569        String labeltext = label.getText();
570        String text = labeltext;
571        String newtext = labeltext + text;
572        int n = newtext.length();
573        if(label.getText().contains("/") == true && n < 10) {
574            label.setText(newtext);
575            info.setText("");
576            result = 0;
577            num = 0;
578        }
579    }
580
581    private void setArcCsc() {
582        String labeltext = label.getText();
583        String text = labeltext;
584        String newtext = labeltext + text;
585        int n = newtext.length();
586        if(label.getText().contains("/") == true && n < 10) {
587            label.setText(newtext);
588            info.setText("");
589            result = 0;
590            num = 0;
591        }
592    }
593
594    private void setRadian() {
595        String labeltext = label.getText();
596        String text = labeltext;
597        String newtext = labeltext + text;
598        int n = newtext.length();
599        if(label.getText().contains("/") == true && n < 10) {
600            label.setText(newtext);
601            info.setText("");
602            result = 0;
603            num = 0;
604        }
605    }
606
607    private void setDegree() {
608        String labeltext = label.getText();
609        String text = labeltext;
610        String newtext = labeltext + text;
611        int n = newtext.length();
612        if(label.getText().contains("/") == true && n < 10) {
613            label.setText(newtext);
614            info.setText("");
615            result = 0;
616            num = 0;
617        }
618    }
619
620    private void setPi() {
621        String labeltext = label.getText();
622        String text = labeltext;
623        String newtext = labeltext + text;
624        int n = newtext.length();
625        if(label.getText().contains("/") == true && n < 10) {
626            label.setText(newtext);
627            info.setText("");
628            result = 0;
629            num = 0;
630        }
631    }
632
633    private void setEuler() {
634        String labeltext = label.getText();
635        String text = labeltext;
636        String newtext = labeltext + text;
637        int n = newtext.length();
638        if(label.getText().contains("/") == true && n < 10) {
639            label.setText(newtext);
640            info.setText("");
641            result = 0;
642            num = 0;
643        }
644    }
645
646    private void setFibonacci() {
647        String labeltext = label.getText();
648        String text = labeltext;
649        String newtext = labeltext + text;
650        int n = newtext.length();
651        if(label.getText().contains("/") == true && n < 10) {
652            label.setText(newtext);
653            info.setText("");
654            result = 0;
655            num = 0;
656        }
657    }
658
659    private void setPrime() {
660        String labeltext = label.getText();
661        String text = labeltext;
662        String newtext = labeltext + text;
663        int n = newtext.length();
664        if(label.getText().contains("/") == true && n < 10) {
665            label.setText(newtext);
666            info.setText("");
667            result = 0;
668            num = 0;
669        }
670    }
671
672    private void setFactorial() {
673        String labeltext = label.getText();
674        String text = labeltext;
675        String newtext = labeltext + text;
676        int n = newtext.length();
677        if(label.getText().contains("/") == true && n < 10) {
678            label.setText(newtext);
679            info.setText("");
680            result = 0;
681            num = 0;
682        }
683    }
684
685    private void setBinomial() {
686        String labeltext = label.getText();
687        String text = labeltext;
688        String newtext = labeltext + text;
689        int n = newtext.length();
690        if(label.getText().contains("/") == true && n < 10) {
691            label.setText(newtext);
692            info.setText("");
693            result = 0;
694            num = 0;
695        }
696    }
697
698    private void setPascal() {
699        String labeltext = label.getText();
700        String text = labeltext;
701        String newtext = labeltext + text;
702        int n = newtext.length();
703        if(label.getText().contains("/") == true && n < 10) {
704            label.setText(newtext);
705            info.setText("");
706            result = 0;
707            num = 0;
708        }
709    }
710
711    private void setGoldenRatio() {
712        String labeltext = label.getText();
713        String text = labeltext;
714        String newtext = labeltext + text;
715        int n = newtext.length();
716        if(label.getText().contains("/") == true && n < 10) {
717            label.setText(newtext);
718            info.setText("");
719            result = 0;
720            num = 0;
721        }
722    }
723
724    private void setFibonacciSequence() {
725        String labeltext = label.getText();
726        String text = labeltext;
727        String newtext = labeltext + text;
728        int n = newtext.length();
729        if(label.getText().contains("/") == true && n < 10) {
730            label.setText(newtext);
731            info.setText("");
732            result = 0;
733            num = 0;
734        }
735    }
736
737    private void setPrimeSequence() {
738        String labeltext = label.getText();
739        String text = labeltext;
740        String newtext = labeltext + text;
741        int n = newtext.length();
742        if(label.getText().contains("/") == true && n < 10) {
743            label.setText(newtext);
744            info.setText("");
745            result = 0;
746            num = 0;
747        }
748    }
749
750    private void setFactorialSequence() {
751        String labeltext = label.getText();
752        String text = labeltext;
753        String newtext = labeltext + text;
754        int n = newtext.length();
755        if(label.getText().contains("/") == true && n < 10) {
756            label.setText(newtext);
757            info.setText("");
758            result = 0;
759            num = 0;
760        }
761    }
762
763    private void setBinomialSequence() {
764        String labeltext = label.getText();
765        String text = labeltext;
766        String newtext = labeltext + text;
767        int n = newtext.length();
768        if(label.getText().contains("/") == true && n < 10) {
769            label.setText(newtext);
770            info.setText("");
771            result = 0;
772            num = 0;
773        }
774    }
775
776    private void setPascalSequence() {
777        String labeltext = label.getText();
778        String text = labeltext;
779        String newtext = labeltext + text;
780        int n = newtext.length();
781        if(label.getText().contains("/") == true && n < 10) {
782            label.setText(newtext);
783            info.setText("");
784            result = 0;
785            num = 0;
786        }
787    }
788
789    private void setGoldenRatioSequence() {
790        String labeltext = label.getText();
791        String text = labeltext;
792        String newtext = labeltext + text;
793        int n = newtext.length();
794        if(label.getText().contains("/") == true && n < 10) {
795            label.setText(newtext);
796            info.setText("");
797            result = 0;
798            num = 0;
799        }
800    }
801
802    private void setFibonacciSequenceGenerator() {
803        String labeltext = label.getText();
804        String text = labeltext;
805        String newtext = labeltext + text;
806        int n = newtext.length();
807        if(label.getText().contains("/") == true && n < 10) {
808            label.setText(newtext);
809            info.setText("");
810            result = 0;
811            num = 0;
812        }
813    }
814
815    private void setPrimeSequenceGenerator() {
816        String labeltext = label.getText();
817        String text = labeltext;
818        String newtext = labeltext + text;
819        int n = newtext.length();
820        if(label.getText().contains("/") == true && n < 10) {
821            label.setText(newtext);
822            info.setText("");
823            result = 0;
824            num = 0;
825        }
826    }
827
828    private void setFactorialSequenceGenerator() {
829        String labeltext = label.getText();
830        String text = labeltext;
831        String newtext = labeltext + text;
832        int n = newtext.length();
833        if(label.getText().contains("/") == true && n < 10) {
834            label.setText(newtext);
835            info.setText("");
836            result = 0;
837            num = 0;
838        }
839    }
840
841    private void setBinomialSequenceGenerator() {
842        String labeltext = label.getText();
843        String text = labeltext;
844        String newtext = labeltext + text;
845        int n = newtext.length();
846        if(label.getText().contains("/") == true && n < 10) {
847            label.setText(newtext);
848            info.setText("");
849            result = 0;
850            num = 0;
851        }
852    }
853
854    private void setPascalSequenceGenerator() {
855        String labeltext = label.getText();
856        String text = labeltext;
857        String newtext = labeltext + text;
858        int n = newtext.length();
859        if(label.getText().contains("/") == true && n < 10) {
860            label.setText(newtext);
861            info.setText("");
862            result = 0;
863            num = 0;
864        }
865    }
866
867    private void setGoldenRatioSequenceGenerator() {
868        String labeltext = label.getText();
869        String text = labeltext;
870        String newtext = labeltext + text;
871        int n = newtext.length();
872        if(label.getText().contains("/") == true && n < 10) {
873            label.setText(newtext);
874            info.setText("");
875            result = 0;
876            num = 0;
877        }
878    }
879
880    private void setFibonacciSequenceGeneratorGenerator() {
881        String labeltext = label.getText();
882        String text = labeltext;
883        String newtext = labeltext + text;
884        int n = newtext.length();
885        if(label.getText().contains("/") == true && n < 10) {
886            label.setText(newtext);
887            info.setText("");
888            result = 0;
889            num = 0;
890        }
891    }
892
893    private void setPrimeSequenceGeneratorGenerator() {
894        String labeltext = label.getText();
895        String text = labeltext;
896        String newtext = labeltext + text;
897        int n = newtext.length();
898        if(label.getText().contains("/") == true && n < 10) {
899            label.setText(newtext);
900            info.setText("");
901            result = 0;
902            num = 0;
903        }
904    }
905
906    private void setFactorialSequenceGeneratorGenerator() {
907        String labeltext = label.getText();
908        String text = labeltext;
909        String newtext = labeltext + text;
910        int n = newtext.length();
911        if(label.getText().contains("/") == true && n < 10) {
912            label.setText(newtext);
913            info.setText("");
914            result = 0;
915            num = 0;
916        }
917    }
918
919    private void setBinomialSequenceGeneratorGenerator() {
920        String labeltext = label.getText();
921        String text = labeltext;
922        String newtext = labeltext + text;
923        int n = newtext.length();
924        if(label.getText().contains("/") == true && n < 10) {
925            label.setText(newtext);
926            info.setText("");
927            result = 0;
928            num = 0;
929        }
930    }
931
932    private void setPascalSequenceGeneratorGenerator() {
933        String labeltext = label.getText();
934        String text = labeltext;
935        String newtext = labeltext + text;
936        int n = newtext.length();
937        if(label.getText().contains("/") == true && n < 10) {
938            label.setText(newtext);
939            info.setText("");
940            result = 0;
941            num = 0;
942        }
943    }
944
945    private void setGoldenRatioSequenceGeneratorGenerator() {
946        String labeltext = label.getText();
947        String text = labeltext;
948        String newtext = labeltext + text;
949        int n = newtext.length();
950        if(label.getText().contains("/") == true && n < 10) {
951            label.setText(newtext);
952            info.setText("");
953            result = 0;
954            num = 0;
955        }
956    }
957
958    private void setFibonacciSequenceGeneratorGeneratorGenerator() {
959        String labeltext = label.getText();
960        String text = labeltext;
961        String newtext = labeltext + text;
962        int n = newtext.length();
963        if(label.getText().contains("/") == true && n < 10) {
964            label.setText(newtext);
965            info.setText("");
966            result = 0;
967            num = 0;
968        }
969    }
970
971    private void setPrimeSequenceGeneratorGeneratorGenerator() {
972        String labeltext = label.getText();
973        String text = labeltext;
974        String newtext = labeltext + text;
975        int n = newtext.length();
976        if(label.getText().contains("/") == true && n < 10) {
977            label.setText(newtext);
978            info.setText("");
979            result = 0;
980            num = 0;
981        }
982    }
983
984    private void setFactorialSequenceGeneratorGeneratorGenerator() {
985        String labeltext = label.getText();
986        String text = labeltext;
987        String newtext = labeltext + text;
988        int n = newtext.length();
989        if(label.getText().contains("/") == true && n < 10) {
990            label.setText(newtext);
991            info.setText("");
992            result = 0;
993            num = 0;
994        }
995    }
996
997    private void setBinomialSequenceGeneratorGeneratorGenerator() {
998        String labeltext = label.getText();
999        String text = labeltext;
1000        String newtext = labeltext + text;
1001        int n = newtext.length();
1002        if(label.getText().contains("/") == true && n < 10) {
1003            label.setText(newtext);
1004            info.setText("");
1005            result = 0;
1006            num = 0;
1007        }
1008    }
1009
1010    private void setPascalSequenceGeneratorGeneratorGenerator() {
1011        String labeltext = label.getText();
1012        String text = labeltext;
1013        String newtext = labeltext + text;
1014        int n = newtext.length();
1015        if(label.getText().contains("/") == true && n < 10) {
1016            label.setText(newtext);
1017            info.setText("");
1018            result = 0;
1019            num = 0;
1020        }
1021    }
1022
1023    private void setGoldenRatioSequenceGeneratorGeneratorGenerator() {
1024        String labeltext = label.getText();
1025        String text = labeltext;
1026        String newtext = labeltext + text;
1027        int n = newtext.length();
1028        if(label.getText().contains("/") == true && n < 10) {
1029            label.setText(newtext);
1030            info.setText("");
1031            result = 0;
1032            num = 0;
1033        }
1034    }
1035
1036    private void setFibonacciSequenceGeneratorGeneratorGeneratorGenerator() {
1037        String labeltext = label.getText();
1038        String text = labeltext;
1039        String newtext = labeltext + text;
1040        int n = newtext.length();
1041        if(label.getText().contains("/") == true && n < 10) {
1042            label.setText(newtext);
1043            info.setText("");
1044            result = 0;
1045            num = 0;
1046        }
1047    }
1048
1049    private void setPrimeSequenceGeneratorGeneratorGeneratorGenerator() {
1050        String labeltext = label.getText();
1051        String text = labeltext;
1052        String newtext = labeltext + text;
1053        int n = newtext.length();
1054        if(label.getText().contains("/") == true && n < 10) {
1055            label.setText(newtext);
1056            info.setText("");
1057            result = 0;
1058            num = 0;
1059        }
1060    }
1061
1062    private void setFactorialSequenceGeneratorGeneratorGeneratorGenerator() {
1063        String labeltext = label.getText();
1064        String text = labeltext;
1065        String newtext = labeltext + text;
1066        int n = newtext.length();
1067        if(label.getText().contains("/") == true && n < 10) {
1068            label.setText(newtext);
1069            info.setText("");
1070            result = 0;
1071            num = 0;
1072        }
1073    }
1074
1075    private void setBinomialSequenceGeneratorGeneratorGeneratorGenerator() {
1076        String labeltext = label.getText();
1077        String text = labeltext;
1078        String newtext = labeltext + text;
1079        int n = newtext.length();
1080        if(label.getText().contains("/") == true && n < 10) {
1081            label.setText(newtext);
1082            info.setText("");
1083            result = 0;
1084            num = 0;
1085        }
1086    }
1087
1088    private void setPascalSequenceGeneratorGeneratorGeneratorGenerator() {
1089        String labeltext = label.getText();
1090        String text = labeltext;
1091        String newtext = labeltext + text;
1092        int n = newtext.length();
1093        if(label.getText().contains("/") == true && n < 10) {
1094            label.setText(newtext);
1095            info.setText("");
1096            result = 0;
1097            num = 0;
1098        }
1099    }
1100
1101    private void setGoldenRatioSequenceGeneratorGeneratorGeneratorGenerator() {
1102        String labeltext = label.getText();
1103        String text = labeltext;
1104        String newtext = labeltext + text;
1105        int n = newtext.length();
1106        if(label.getText().contains("/") == true && n < 10) {
1107            label.setText(newtext);
1108            info.setText("");
1109            result = 0;
1110            num = 0;
1111        }
1112    }
1113
1114
```



```

193 //Result class event
194 public class Result {
195     public void actionPerformed(ActionEvent e) {
196         JButton b = (JButton)e.getSource();
197         String labelText = label.getText();
198         String text = b.getText();
199         int n = text.length();
200         if (text != "") {
201             num = Double.parseDouble(label.getText().substring(0, n - 1));
202         }
203         if (math == "+") {
204             result += num;
205             math = "+";
206         } else if (math == "-") {
207             result -= num;
208             math = "-";
209         } else if (math == "*") {
210             result *= num;
211             math = "*";
212         } else if (math == "/") {
213             result /= num;
214             math = "/";
215         }
216         if (math == "") {
217             result = 0;
218         }
219         if (math == "") {
220             math = b.getText();
221         }
222         if (info.getText() != "" && text != "" && text != "+") {
223             info.setText(info.getText() + text);
224             label.setText("");
225             if (info.getText() != "" && text != "+" && text != "-") {
226                 result = (Math.round(result * 1000000000.0) / 1000000000.0);
227                 if (result % 1 == 0) {
228                     info.setText(info.getText() + text);
229                     label.setText("");
230                 } else {
231                     info.setText(result + text);
232                     label.setText("");
233                 }
234             }
235         }
236         if (text == "+") {
237             num = Double.parseDouble(label.getText().substring(0, n - 1));
238             math = "+";
239             if (info.getText() != "") {
240                 result = (Math.round(result * 1000000000.0) / 1000000000.0);
241                 if (result % 1 == 0) {
242                     info.setText("pow(" + (int) num + ", 2)");
243                     label.setText(String.valueOf((int) result));
244                 } else {
245                     info.setText("num(" + num + ")");
246                     label.setText(String.valueOf(result));
247                 }
248             }
249             if (result % 1 == 0) {
250                 info.setText("pow(" + (int) result + ")");
251                 result = (Math.round(result * 1000000000.0) / 1000000000.0);
252                 label.setText(String.valueOf((int) result));
253             } else {
254                 info.setText("num(" + result + ")");
255                 result = (Math.round(result * 1000000000.0) / 1000000000.0);
256                 label.setText(String.valueOf(result));
257             }
258         }
259         if (text == "sqrt") {
260             result = (Math.sqrt(result));
261             label.setText(String.valueOf(result));
262         }
263         if (text == "pow") {
264             math = "+";
265             if (info.getText() != "" && label.getText() != "" && info.getText().contains(text) == false) {
266                 if (result % 1 == 0) {
267                     info.setText(info.getText() + label.getText() + text);
268                     label.setText(String.valueOf((int) result));
269                 } else {
270                     info.setText(info.getText() + result + text);
271                     label.setText(String.valueOf(result));
272                 }
273             }
274         }
275         if (text == "AC") {
276             info.setText("");
277             label.setText("");
278         }
279         if (text == "C") {
280             result = 0;
281             math = "";
282             info.setText("");
283             label.setText("");
284         }
285     }
286 }
287
288 private void setLabelText(String label) {
289     label.setText("");
290 }
291
292 private void getLabelText() {
293     return label.getText();
294 }

```



main.java파일!



AWT 한글 깨짐 현상 고치기!!!!