

화면을 만들고 디자인 하기 위한 awt 패키지의 클래스들을 살펴보자.

예제 1

```
public class FrameTest {

    public static void main(String[] args) {

        Frame frame = new Frame("첫 프레임");//프레임 제목
        frame.setBounds(800, 100, 400, 300);//( x좌표, y좌표, 너비, 높이 )

        System.out.println(frame.getBounds().getWidth());//double형식으로 반환
        System.out.println(frame.getBounds().height());//int형 속성

        frame.setBackground(Color.BLUE);//배경색 지정

        frame.setVisible(true);//true가 없으면 화면에 보이지 않음
    }
}
```

예제 2

```
public class FrameTest2 {

    public static void main(String[] args) {
        JFrame fr = new JFrame("두번째");
        fr.setSize(400, 400);//사이즈
        fr.setLocation(800, 100);//좌표

        //x버튼으로 프레임 닫기
        //프레임이 닫힐때 아예 메모리에서 없어져라!
        //이 기능은 JFrame에서만 사용 가능하며 일반 Frame에서는 사용할 수 없다.
        fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        fr.setVisible(true);

        //여러개의 프레임중 원하는 하나의 프레임만 종료하고자 한다면 dispose()
        //열려있는 프레임의 수와 상관없이 강제로 VM을 종료하려면 System.exit(0)
        //fr.dispose();
    }
}
```

예제3

```
public class FrameTest3 {

    public static void main(String[] args) {
        /*
        화면 중앙에 프레임 위치 시키기
        1. 실행 중인 모니터의 해상도를 알아야한다.
        2. 모니터의 중앙 좌표를 알아놓은 후,,
        3. 프레임 좌우 크기를 중앙 좌표에서 빼줘야한다.
            awt.Toolkit <-- 싱글톤
            Toolkit    beep()
                        get모니터사이즈() <=!!!  width  height  값!@

        x : 모니터 중앙 좌표값 - 내가 만들려는 프레임의width/2
        y : 모니터 중앙 좌표값 - 내가 만들려는 프레임의height/2
        */
        JFrame fr = new JFrame("세번째");
        fr.setSize(400, 400);
        //Toolkit을 통해 모니터 해상도 얻어오기
        Toolkit tk = Toolkit.getDefaultToolkit();
        /*for(int i=0; i<3; i++){
            try { Thread.sleep(1000); } catch (Exception e) {}
            tk.beep(); //1초 간격으로 비프음 출력
        }*/
        Dimension di = tk.getScreenSize();
        int monitorW=di.width;
        int monitorH=di.height;

        int x = monitorW/2-fr.getWidth()/2;
        int y = monitorH/2-fr.getHeight()/2;

        fr.setLocation(x, y);
        fr.setVisible(true);

        fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        fr.setVisible(true);

    }
}
```

예제4

프레임을 상속받는 클래스를 만들고, 이 클래스를 객체화하여 화면에 뿌려보자.

MyFrame클래스 생성

```
public class MyFrame extends Frame{
    public MyFrame() { //생성자
        setBounds(500, 200, 400, 400);
        setVisible(true);
    }
}
```

FrameTest4클래스를 만들고 MyFrame을 객체화해 사용해보자.

```
public class FrameTest4 {
    public static void main(String[] args) {
        MyFrame fr = new MyFrame();
        fr.setBackground(Color.YELLOW);
        fr.setTitle("네번째 사용자 프레임");
    }
}
```

예제5 - 우상단 메뉴 이벤트 감지하기

WindowListener를 구현하는 MyEventListener클래스 생성

```
public class MyEventListener implements WindowListener{

    @Override
    public void windowActivated(WindowEvent e) { }

    @Override
    public void windowClosed(WindowEvent e) { }

    @Override
    public void windowClosing(WindowEvent e) {
        System.out.println("x버튼을 클릭함");
        System.exit(0);
    }

    @Override
    public void windowDeactivated(WindowEvent e) { }
```

```

@Override
public void windowDeiconified(WindowEvent e) {
    System.out.println("원래대로 돌아옴");
}

@Override
public void windowIconified(WindowEvent e) {
    System.out.println("최소화 됨");
}

@Override
public void windowOpened(WindowEvent e) { }
}

```

FrameTest5클래스를 생성하고 이벤트 감지자를 붙여보자.

```

public class FrameTest5 {
    public static void main(String[] args) {

        MyFrame fr = new MyFrame();
        fr.setBackground(Color.YELLOW);
        fr.setTitle("다섯번째 사용자 프레임");

        //프레임에서 _ X 에 해당하는 이벤트를 감지자를 등록한다
        //이 감지자는 WindowListener를 구현하고 있는 MyEventListener클래스 이다.
        fr.addWindowListener( new MyEventListener() );
    }
}

```

실행후에 최소화, X버튼에 대한 이벤트가 감지되는지 확인.

위의 감지자(MyEventListener())에서 필요한 기능만 재정의 해서 사용할 수 있도록 다른 형태로 클래스를 한번 정의해 보자.

ListenerClass를 생성하고 WindowListener 구현

```

public class ListenerClass implements WindowListener{

    @Override
    public void windowActivated(WindowEvent e) { }

    @Override

```

```

public void windowClosed(WindowEvent e) { }

@Override
public void windowClosing(WindowEvent e) { }

@Override
public void windowDeactivated(WindowEvent e) { }

@Override
public void windowDeiconified(WindowEvent e) { }

@Override
public void windowIconified(WindowEvent e) { }

@Override
public void windowOpened(WindowEvent e) { }
}

```

WinClosingListener를 생성해서 ListenerClass를 상속받고, windowClosing()메서드를 재정의하자

```

public class WinClosingListener extends ListenerClass{

```

```

    @Override
    public void windowClosing(WindowEvent e) {
        System.out.println("종료버튼은 눌렀지만 종료하진 않을거다!!");
    }
}

```

FrameTest5 클래스의 내용 수정!!

```

public class FrameTest5 {
    public static void main(String[] args) {
        ....

        //프레임에서 _ X 에 해당하는 이벤트를 감지자를 등록한다
        //이 감지자는 WindowListener를 구현하고 있는 MyEventListener클래스 이다.
        //fr.addWindowListener( new MyEventListener() );
        fr.addWindowListener( new WinClosingListener() );
    }
}

```

똑같이 windowClosing()메서드를 사용하는 것이긴 하지만, 상황에 따라 재정의가 가능한 형태라고 보면 된다.

하지만 closing은 워낙 많이 사용하는 감지자이기도 해서, 익명 내부 클래스로 바로 어댑터를 생성하여 처리하는 것이 간단하다.

FrameTest5의 내용 수정

```
public class FrameTest5 {
    public static void main(String[] args) {
        .....

        //프레임에서 _ X 에 해당하는 이벤트를 감지자를 등록한다
        //이 감지자는 WindowListener를 구현하고 있는 MyEventListener클래스 이다.
        //fr.addWindowListener( new MyEventListener() );

        //fr.addWindowListener( new WinClosingListener() );

        //익명으로 감지자를 바로 붙여서 사용하는 방법
        fr.addWindowListener( new WindowAdapter(){
            @Override
            public void windowClosing(WindowEvent e) {
                System.out.println("어댑터를 생성하여 감지!!");
            }
        } );
    }
}
```

예제6 - 버튼 붙여보기

```
public class ButtonTest {
    public static void main(String[] args) {

        Frame f = new Frame("버튼 테스트");
        f.setBounds(200, 200, 400, 400);

        f.setLayout(null); //자동배치를 끈다.
        //그 순간 frame에 추가되는 각각의 컴포넌트는 고유의 size()와
        //location()을 갖고 있어야 한다.
        Button btnOk = new Button("확인");
        btnOk.setBounds(70, 90, 100, 50);

        Button btnClose = new Button("닫기");
        btnClose.setBounds(btnOk.getBounds());//btnOK와 같은 Bounds로 만든다
    }
}
```

//btnOK와 같은 Bounds라면 버튼 두개가 겹쳐져서 표시되므로
//닫기버튼은 OK버튼의 오른쪽에 붙여 준다.

```
btnClose.setLocation(btnOk.getWidth()+btnOk.getX() + 60, btnOk.getY());
```

//버튼은 먼저 add할수록 화면 위쪽에 보여진다

```
f.add(btnOk);
```

```
f.add(btnClose);
```

```
f.setVisible(true);
```

//종료버튼 감지

```
f.addWindowListener( new WindowAdapter(){  
    @Override  
    public void windowClosing(WindowEvent e) {  
        System.exit(0);  
    }  
});
```

```
}
```

```
}
```

예제7 - 레이아웃 종류(FlowLayout)

```
public class MyFlowLayout{
```

```
    Frame f = new Frame(); // 패널 초기화
```

```
    Button jb1 = new Button("버튼1"); // 버튼 초기화
```

```
    Button jb2 = new Button("버튼2"); // 버튼 초기화
```

```
    Button jb3 = new Button("버튼3"); // 버튼 초기화
```

```
    Button jb4 = new Button("버튼4"); // 버튼 초기화
```

```
    Button jb5 = new Button("버튼5"); // 버튼 초기화
```

```
    Button jb6 = new Button("버튼6"); // 버튼 초기화
```

```
    public MyFlowLayout(){
```

```
        f.setLayout( new FlowLayout() ); //Frame을 FlowLayout으로 세팅
```

```
        f.add(jb1); // jp라는 패널에 jb1라는 버튼 추가
```

```
        f.add(jb2); // jp라는 패널에 jb2라는 버튼 추가
```

```
        f.add(jb3); // jp라는 패널에 jb3라는 버튼 추가
```

```
        f.add(jb4); // jp라는 패널에 jb4라는 버튼 추가
```

```
        f.add(jb5); // jp라는 패널에 jb5라는 버튼 추가
```

```

        f.add(jb6); // jp라는 패널에 jb6라는 버튼 추가

        f.setSize(400, 300); // 윈도우의 크기 가로x세로
        f.setVisible(true);

        f.addWindowListener( new WindowAdapter(){
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    } //생성자

    public static void main(String[] args){
        new MyFlowLayout();
    }
}

```

예제8 - 레이아웃 종류(BorderLayout)

```

public class MyBorderLayout {

    Frame f = new Frame(); // 패널 초기화
    Button jb1 = new Button("북");
    Button jb2 = new Button("서");
    Button jb3 = new Button("중앙");
    Button jb4 = new Button("동");
    Button jb5 = new Button("남");
    Button jb6 = new Button("버튼6");

    public MyBorderLayout(){
        //프레임을 보더레이아웃으로 세팅
        f.setLayout( new BorderLayout() );

        //프레임에 버튼 추가
        //BorderLayout은 버튼 추가시에 North, West, Center, East, South의
        //다섯위치 중 어디에 배치시킬 것인지를 반드시 명시해줘야 한다.
        f.add(jb1, BorderLayout.NORTH ); //“North” 라고 명시해도 되긴 함
        f.add(jb2, BorderLayout.WEST );
        f.add(jb3, BorderLayout.CENTER );
        f.add(jb4, BorderLayout.EAST );
        f.add(jb5, BorderLayout.SOUTH );
    }
}

```



```

f.add(jb6, "Center");

f.setSize(400, 300); // 윈도우의 크기 가로x세로
f.setVisible(true); // 창을 보여줄때 true, 숨길때 false

f.addWindowListener(new WindowAdapter() {

    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
}

public static void main(String[] args){
    new MyBorderLayout();
}
}

```

예제9 - 버튼 클릭

```

public class ButtonTest {
    public static void main(String[] args) {

        Frame f = new Frame();

        f.setBounds(800, 100, 1000, 500);
        f.setLayout( new FlowLayout() );

        Button btn1 = new Button("1");
        Button btn2 = new Button("2");
        Button btn3 = new Button("3");
        Button btn4 = new Button("4");

        //FlowLayout에서의 버튼 크기 변경은 이렇게 해줘야 한다.
        //그냥 btn1.setSize(200, 100) 로는 변경 안됨
        btn1.setPreferredSize(new Dimension(200, 100));
        btn2.setPreferredSize(new Dimension(200, 100));
        btn3.setPreferredSize(new Dimension(200, 100));
        btn4.setPreferredSize(new Dimension(200, 100));

        f.add(btn1);
    }
}

```

```
f.add(btn2);
```

```
f.add(btn3);
```

```
f.add(btn4);
```

```
f.setVisible(true);
```

```
//이벤트 감지자 등록
```

```
btn1.addActionListener(al);
```

```
btn2.addActionListener(al);
```

```
btn3.addActionListener(al);
```

```
btn4.addActionListener(al);
```

```
//종료버튼 감지
```

```
f.addWindowListener(new WindowAdapter() {
```

```
    @Override
```

```
    public void windowClosing(WindowEvent e) {
```

```
        System.exit(0);
```

```
    }
```

```
});
```

```
}//main
```

```
static ActionListener al = new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        System.out.println( e.getActionCommand() ); //버튼에 부착되어 있는 제목
```

```
        //System.out.println(e);
```

```
        //System.out.println(e.getID());
```

```
        //System.out.println( ((Button)e.getSource()).getWidth());
```

```
        //버튼에 부착되어 있는 제목을 통해 어떤 버튼이 클릭되었는지 구별
```

```
        if(e.getActionCommand().equals("1")){
```

```
            System.out.println("1번 눌렀음");
```

```
        }else if(e.getActionCommand().equals("2")){
```

```
            System.out.println("2번 눌렀음");
```

```
        }else if(e.getActionCommand().equals("3")){
```

```
            System.out.println("3번 눌렀음");
```

```

        }else if(e.getActionCommand().equals("4")){
            System.out.println("4번 눌렀음");

        }
        System.out.println("-----");
    }
};
}

```

예제10 - CheckBox와 RadioButton

```

public class CheckBoxTest {

    public static void main(String[] args) {
        Frame f = new Frame("질문");

        f.setBounds(500, 100, 800, 250);

        f.setLayout(new FlowLayout());

        Label q1 = new Label("1. 관심 분야는 무엇입니까?");
        Checkbox news = new Checkbox("news", true); //처음부터 체크하고 시작한다.
        Checkbox sports = new Checkbox("sports");
        Checkbox movie = new Checkbox("영화");
        Checkbox music = new Checkbox("음악");

        //체크박스 선택 여부 판단
        news.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {

                String str = e.getStateChange() == 1 ? "뉴스 선택됨" : "뉴스 선택해제";
                System.out.println( str );
            }
        });

        Font font = new Font("맑은고딕", Font.BOLD, 30);

        q1.setFont(font);
        f.add(q1);
        f.add(news);
        f.add(sports);
    }
}

```

```
f.add(movie);
f.add(music);
```

```
//배타적 선택 : 여러개중에 한개만 선택(라디오 버튼(둥근모양))
Label q2 = new Label("2. 영화는 한달에 얼마나 자주 봅니다? ");
CheckboxGroup group = new CheckboxGroup();
Checkbox one = new Checkbox("한번", group, true);
Checkbox two = new Checkbox("두번", group, false);
Checkbox three = new Checkbox("세번",group, false);
```

```
//라디오 버튼 선택여부 판단
one.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {

        System.out.println( "한 번 봅니다" );
    }
});
```

```
two.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {

        System.out.println( "두 번 봅니다" );
    }
});
```

```
q2.setFont(font);
f.add(q2);
f.add(one);
f.add(two);
f.add(three);
```

```
f.setVisible(true);
```

```
f.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
```

```
}//main
```

```
}
```

예제 11 - Choice

```
public class ChoiceTest {

    public static void main(String[] args) {
        Frame f = new Frame("질문");

        f.setSize(500, 250);
        f.setLocation(400, 100);

        //레이아웃 없음. 자동배치 끄기
        f.setLayout(null);

        Choice day = new Choice();
        day.add("SUN");
        day.add("MON");
        day.add("TUE");
        day.add("WED");
        day.add("THU");
        day.add("FRI");
        day.add("SAT");

        //사이즈를 설정하려면 자동배치를 꺼야한다. 현재 크기가 반영된다.
        //높이는 안에있는 요소의 사이즈로 결정되므로 크게 의미가 없다.
        day.setSize(150, 0);
        day.setLocation(50, 100); //초이스 위치

        day.addItemListener(new ItemListener() {

            @Override
            public void itemStateChanged(ItemEvent e) {

                System.out.println(day.getSelectedItem());

            }

        });

        f.add(day);
        f.setVisible(true);
    }
}
```

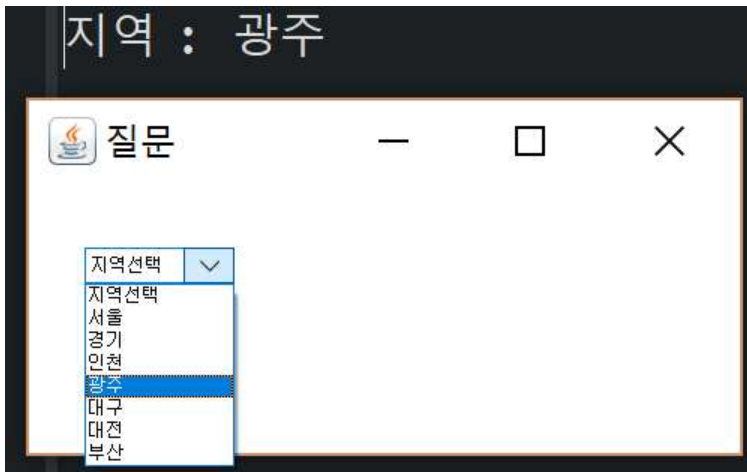
```

        f.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    } //main
}

```

예제12 - Choice감지자

문제 : 다음과 같은 결과를 완성하시오(지역을 선택하고 선택한 지역이 콘솔에 표시)



이벤트를 분리시키기 위해 ItemListener를 구현하는 ChoiceHandler 생성

```

public class ChoiceHandler implements ItemListener{
    @Override
    public void itemStateChanged(ItemEvent e) {

        String str = (String)e.getItem();
        System.out.println( "지역 : " + str );

    }
}

```

ChoiceTest2 클래스 생성 및 내용 추가

```

public class ChoiceTest2 {

    public static void main(String[] args) {
        Frame f = new Frame("질문");

        f.setBounds(400, 100, 500, 250);
    }
}

```

```

f.setLayout(null);

Choice day = new Choice();
day.add("지역선택");
day.add("서울");
day.add("경기");
day.add("인천");
day.add("광주");
day.add("대구");
day.add("대전");
day.add("부산");

//사이즈를 설정하려면 자동배치를 꺼야한다. 현재 크기가 반영된다.
//높이는 안에있는 요소의 사이즈로 결정되므로 크게 의미가 없다.
day.setSize(100, 0);
day.setLocation(50, 100);

day.addItemListener( new ChoiceHandler() );

f.add(day);
f.setVisible(true);

f.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
}

}

```

예제13 - Frame 레이아웃 예제

```

public class FrameTest {
    public static void main(String[] args) {

        Frame frame = new Frame("제목");

        //버튼객체 배열로 준비
        Button [] arbtn = new Button[5];

        String [] btnTitles={ "닫기", "열기", "취소", "확인", "안녕" };
    }
}

```

```

for(int i=0; i<arbtn.length; i++){
    arbtn[i] = new Button( btnTitles[i] );
}

//Frame도 BorderLayout과 같이 위치 선정이 가능하다.
frame.add(arbtn[0], BorderLayout.EAST);
frame.add(arbtn[1], BorderLayout.WEST);
frame.add(arbtn[2], BorderLayout.SOUTH);
frame.add(arbtn[3], BorderLayout.NORTH);
frame.add(arbtn[4], BorderLayout.CENTER);

//프레임에 내용물이 있을 때 그 내용물의 내용 만큼만 영역을 확보해 준다.
frame.pack();
frame.setVisible(true);

//동쪽에 위치한 닫기버튼 클릭
arbtn[0].addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});

//x버튼 누를때
frame.addWindowListener(new WindowAdapter() {

    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
}

}

```

예제14 - 모양 만들어서 이벤트 처리 해보기

```

public class FrameTest2 {

    public static void main(String[] args) {

```


//#프레임 설정

```
Frame frame = new Frame("문장 입력기");  
frame.setBounds(800, 100, 250, 400);  
frame.setBackground(Color.CYAN);
```

//폰트

```
Font font = new Font(Font.SANS_SERIF, Font.PLAIN, 18);
```

//북쪽단

//컴포넌트를 하나로 묶는다. Panel은 기본이 FlowLayout이다.

```
Panel pNorth = new Panel();  
pNorth.setBackground(Color.GREEN);
```

```
//TextField tf = new TextField("하고싶은 말", 10);
```

```
TextField tf = new TextField(10); //입력상자의 길이
```

```
Button btn = new Button("입력");
```

```
btn.setEnabled(false);
```

```
pNorth.add(tf);
```

```
pNorth.add(btn);
```

```
pNorth.setFont(font);
```

//중앙단

```
TextArea ta =
```

```
    new TextArea("", 0, 0, TextArea.SCROLLBARS_VERTICAL_ONLY);
```

```
ta.setBackground(Color.LIGHT_GRAY);
```

```
ta.setFont(font);
```

```
ta.setEditable(false);
```

//남쪽단

```
Panel pSouth = new Panel();
```

```
pSouth.setFont(font);
```

```
pSouth.setBackground(Color.MAGENTA);
```

```
Button btnSave = new Button("저장");
```

```
Button btnClose = new Button("닫기");
```

```
pSouth.add(btnSave);
```

```
pSouth.add(btnClose);
```

//컴포넌트 배치

```
frame.add(pNorth, BorderLayout.NORTH);
```

```
frame.add(ta, BorderLayout.CENTER);
```

```

frame.add(pSouth, BorderLayout.SOUTH);

frame.setVisible(true);
frame.setResizable(false);
//-----여기까지 모양 잘 나오는지 확인-----

//TextField에 값이 들어간 경우에만 입력버튼 활성화
tf.addTextListener(new TextListener() {

    @Override
    public void textValueChanged(TextEvent e) {
        if(tf.getText().equals("")){
            btn.setEnabled(false);

        }else{
            btn.setEnabled(true);

        }
    }
});
//-----여기까지 처리 잘 되는지 확인-----

tf.addKeyListener(new KeyAdapter() {

    @Override
    public void keyTyped(KeyEvent e) {
        //TextField에 엔터값이 감지되면 아래 TextArea에 내용 추가
        if(e.getKeyChar() == KeyEvent.VK_ENTER){
            ta.append(tf.getText()+"\n");

            //텍스트필드 비워주고 포커스 이동
            tf.setText("");
            tf.requestFocus();

        }
    }
});
//-----여기까지 처리 잘 되는지 확인-----

```

//입력버튼 클릭시에도 TextArea에 내용 추가

```
btn.addActionListener(new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        //ta.setText(tf.getText());
```

```
        ta.append(tf.getText()+"\n");// \r\n
```

```
        //텍스트필드 비워주고 포커스 이동
```

```
        tf.setText("");
```

```
        tf.requestFocus();
```

```
    }
```

```
});
```

```
//-----여기까지 처리 잘 되는지 확인-----
```

//닫기버튼 클릭

```
btnClose.addActionListener(new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        System.exit(0);
```

```
    }
```

```
});
```

```
//-----여기까지 처리 잘 되는지 확인-----
```

//저장버튼 클릭

```
btnSave.addActionListener(new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        String message = ta.getText();
```

```
        try {
```

```
//경로 잡기 : FileDialog (save, load할 때 사용하는 대화상자, 경로를 구해온다)
```

```
        FileDialog fd =
```

```
            new FileDialog(frame, "저장", FileDialog.SAVE);
```

```
        fd.setVisible(true);
```

```
        //System.out.println(fd.getDirectory()+fd.getFile());
```

```
        String path = fd.getDirectory() + fd.getFile();
```

```
        if(!message.equals("")){
```

```

        //스트림으로 path경로에 텍스트를 저장
        FileWriter fw = new FileWriter(path);
        BufferedWriter bw = new BufferedWriter(fw);

        bw.write(message);

        //FileDialog의 취소버튼을 누르지 않고 정상적으로
        //저장한 경우
        if( fd.getFile() != null ){
            JOptionPane.showMessageDialog(
                frame, path+"\n경로에 저장했습니다");
        }

        bw.close();
    }else{
        //TextArea에 기록할 내용이 없다면...
        JOptionPane.showMessageDialog(
            frame, "저장할 내용이 없습니다");
    }

    } catch (IOException e1) {
        e1.printStackTrace();
    }
}

});
//-----여기까지 처리 잘 되는지 확인-----

//종료버튼
frame.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent e) {
        // JOptionPane.showMessageDialog(null,"프로그램을 종료합니다");
        System.exit(0);
    }
});
//-----여기까지 처리 잘 되는지 확인-----

}

}

}

```

예제15 - Menu

MenuTest클래스 생성 및 내용 추가.

```
public class MenuTest {  
  
    public static void main(String[] args) {  
        //프레임에 메뉴를 붙이자  
        Frame f = new Frame("메뉴");  
        f.setSize(300,200);  
  
        MenuBar mb = new MenuBar();  
  
        Menu mFile = new Menu("파일");  
        MenuItem miNew = new MenuItem("New");  
        MenuItem miOpen = new MenuItem("Open");  
        MenuItem miSave = new MenuItem("Save");  
  
        //하위메뉴를 가지고 있는 메뉴  
        Menu print = new Menu("Print");  
        MenuItem printSetup = new MenuItem("프린트 셋업");  
        MenuItem printPre = new MenuItem("미리보기");  
  
        print.add(printSetup);  
        print.add(printPre);  
  
        MenuItem miClose = new MenuItem("Close");  
  
        mFile.add(miNew);  
        mFile.add(miOpen);  
        mFile.add(miSave);  
        mFile.add(print);  
        mFile.add(miClose);  
  
        //메뉴바에 메뉴 추가  
        mb.add(mFile);  
  
        //메뉴를 설정할 때는 add가 아닌 setMenuBar를 사용하여  
        //만들어 놓은 메뉴를 세팅(교체)한다!!!  
        f.setMenuBar(mb);  
        f.setVisible(true);  
    }  
}
```

```

        f.addWindowListener( new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

    } //main
}

```

여기까지 확인 해보고 메뉴 클릭을 감지해보자

MyMenu 클래스 생성 및 내용 추가

```

public class MyMenu implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        //클릭한 메뉴에 쓰여져 있는 텍스트를 구별하여 이벤트 처리
        if(e.getActionCommand().equals("프린트 셋업")){
            System.out.println("프린트 셋업");

        }else if(e.getActionCommand().equalsIgnoreCase("Close")){
            System.out.println("종료");
            System.exit(0);
        }
    } //actionPerformed()
}

```

MenuTest 클래스에 내용 추가

```

public class MenuTest {
    public static void main(String[] args) {
        .....

        //메뉴바에 메뉴 추가
        mb.add(mFile);

        //메뉴를 설정할 때는 add가 아닌 setMenuBar를 사용하여
        //만들어 놓은 메뉴를 세팅(교체)한다!!!
        f.setMenuBar(mb);
        f.setVisible(true);
    }
}

```

```

printSetup.addActionListener(new MyMenu());
miClose.addActionListener(new MyMenu());

f.addWindowListener( new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
} //main
}

```

Frame에서 JLabel을 사용해 이미지 띄우기

```

public class ImageTest {

    public static void main(String[] args) {

        Frame f = new Frame();
        f.setLayout(null);

        f.setBounds(500, 100, 500, 500);

        //이미지
        ImageIcon img = new ImageIcon("개구리1.jpg");

        JLabel jl = new JLabel(img); //JLabel에만 ImageIcon을 추가할 수 있다.
        jl.setBounds(10, 30, 100, 100); //원본이미지의 width, height와 같아야 함
        f.add(jl);

        f.setVisible(true);

        //x버튼 이벤트 감지
        f.addWindowListener( new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        } );

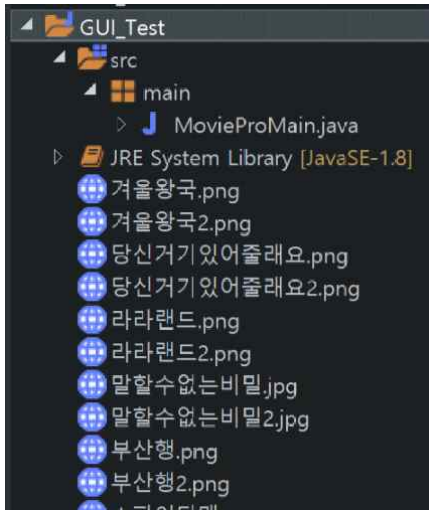
    } //main
}

```

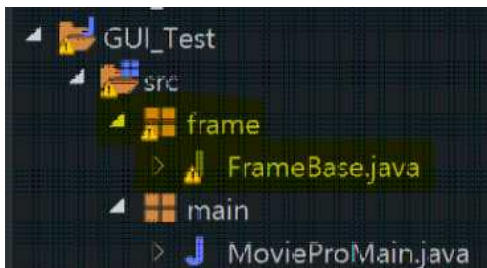
실행해서 프린트 셋업과 닫기버튼에 이벤트가 잘 적용됐는지 확인해보자.

예제16 - 영화예매 프로젝트 만들어보기

GUI_Test 프로젝트 -> main패키지에 MovieProMain클래스 생성
준비해놓은 awt리소스들을 GUI_Test프로젝트에 추가시켜놓는다.



frame패키지 -> FrameBase클래스 생성 및 내용 추가.

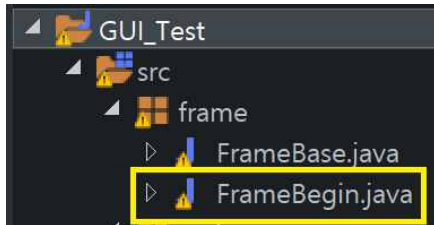


```
public class FrameBase extends JFrame {  
    private static FrameBase instance;  
  
    private FrameBase(JPanel e) {  
        //시스템 정보(화면 크기를 얻기 위한 Toolkit)  
        Toolkit tk = Toolkit.getDefaultToolkit();  
  
        //기본 JFrame 구조  
        setTitle("영화 예매 프로그램");  
        setLayout(null);  
        setBounds((((int) tk.getScreenSize().getWidth()) / 2 - 300,  
                    ((int) tk.getScreenSize().getHeight()) / 2 - 400,  
                    600, 800);  
  
        add(e);  
        setVisible(true);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    } //생성자
```


//싱글톤 기법을 사용하려고 한다

```
public static void getInstance(JPanel e) {  
    //static으로 선언했으므로 해당 메서드가 생성자보다도 먼저 호출된다  
    instance = new FrameBase(e); //생성자를 통해 기본 프레임 정의  
  
    instance.getContentPane().removeAll();  
    instance.getContentPane().add(e);  
  
    instance.revalidate(); //레이아웃 관리자에게 레이아웃정보를 다시 계산하도록 지시  
    instance.repaint(); //레이아웃을 새로 그린다  
} //getInstance()  
}
```

frame패키지 -> FrameBegin클래스 생성 및 내용 추가.



```
public class FrameBegin extends JPanel {  
    public FrameBegin() {
```

//JPanel 구조

```
setBackground(new Color(255, 241, 54));  
setLayout(null);  
setSize(600, 800);
```

//포스터 이미지

```
ImageIcon cinema = new ImageIcon("MSG.jpg");  
JLabel lblCinema = new JLabel(cinema);  
lblCinema.setSize(600, 620);  
lblCinema.setLocation(-15, 0);  
add(lblCinema);
```

//하단의 버튼 설정

```
JButton btnInfo = new JButton("영화 예매");  
JButton btnPost = new JButton("영화 평가하기");
```

```
btnInfo.setBackground(new Color(183, 240, 117));  
btnPost.setBackground(new Color(183, 240, 117));
```

```

        btnInfo.setSize(300, 150);
        btnInfo.setLocation(
            ((int) getSize().getWidth() / 2) - 310,
            (int) getLocation().getY() / 2 + 610);

        btnInfo.setFont(new Font("굴림", Font.BOLD, 26));
        btnPost.setSize(300, 150);

        btnPost.setLocation(
            (int) btnInfo.getLocation().getX() + 300,
            (int) btnInfo.getLocation().getY());

        btnPost.setFont(new Font("굴림", Font.BOLD, 26));

        add(btnInfo);
        add(btnPost);
    } //생성자
}

```

MovieProMain 에 내용 추가.

```

public class MovieProMain {
    public static void main(String[] args) {

        FrameBase.getInstance(new FrameBegin());

    }
}

```

실행 해 보면 아래와 같은 결과가 나온다. 영화 예매 기능부터 추가해보자!



FrameBegin에 내용 추가

```

public class FrameBegin extends JPanel {

    public FrameBegin() {
        .....
        add(btnInfo);
        add(btnPost);

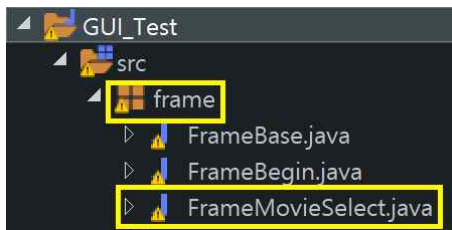
        //버튼 이벤트(예매하기 / 후기 남기기)
        btnInfo.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                //FrameMovieSelect 클래스는 아직 없으므로 오류가 난다.
                FrameBase.getInstance( new FrameMovieSelect() );
            }

        });
    } //생성자
}

```

frame패키지 -> FrameMovieSelect 클래스 생성만 해두자.(오류 방지)



영화 한편의 정보를 저장하기 위한 movieVO패키지 -> Movie클래스 생성 및 내용 추가



```

public class Movie {
    String mName; //영화 이름
    int price; //영화 가격
    double score; //영화 평점
}

```

```
String story; //영화 줄거리
String review; //영화 후기(평점+이름+한줄평)

int total; //평점 누적합
int personScore; //방금 등록한 평점
String rev; //한줄 평 등록
int personNum; //평점 등록한 사람 수
String reviewName = ""; //평점 등록한 사람 이름
```

```
//생성자
```

```
public Movie(String mName, int price, double score, String story,String review) {
    //super();
    this.mName = mName;
    this.price = price;
    this.total = (int)score;
    this.story = story;
    this.review = review;
    this.personNum = 1;
    setScore();
}
```

```
public String getmName() {
    return mName;
}
```

```
public int getPersonScore() {
    return personScore;
}
```

```
public void setPersonScore(int personScore) {
    this.personScore = personScore;
}
```

```
public int getPersonNum() {
    return personNum;
}
```

```
public void setPersonNum() {
    this.personNum++;
}
```

```
public int getTotal() {  
    return total;  
}
```

```
public void setTotal(int personN) {  
    this.total += personN;  
}
```

```
public String getRev() {  
    return rev;  
}
```

```
public void setRev(String rev) {  
    this.rev = rev;  
}
```

```
public String getReviewName() {  
    return reviewName;  
}
```

```
public void setReviewName(String reviewName) {  
    this.reviewName = reviewName;  
}
```

```
public void setStory(String story) {  
    this.story = story;  
}
```

```
public double getScore() {  
    return score;  
}
```

```
public void setScore() {  
    this.score = (double)this.total / this.personNum;  
}
```

```
public String getReview() {  
    return review;  
}
```

```
public void setReview() {
```

```

        this.review += this.personScore + "점\t<" + this.reviewName + ">\t" +
this.rev +"\n";
    }

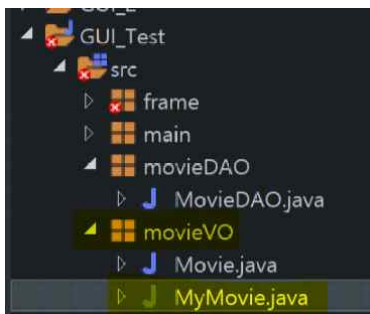
    public int getPrice() {
        return price;
    }

    public String getStory() {
        return story;
    }

    @Override
    public String toString() {
        return "★영화 제목 : " + mName +
"\n★가격 : " + price + "\n★평점 : " + score +
"\n★최신 한줄 평 : " + review;
    }
}

```

개인별 예매정보를 저장하기 위한 MyMovie클래스 생성 및 내용 추가.



```

public class MyMovie {
    // 예매정보(영화 이름, 시간대, 좌석)
    String mName;
    String time;
    String seat;

    public MyMovie() {

    }

    public MyMovie(String mName, String time, String seat) {
        this.mName = mName;
    }
}

```

```

        this.time = time;
        this.seat = seat;
    }

    public String getmName() {
        return mName;
    }

    public void setmName(String mName) {
        this.mName = mName;
    }

    public String getTime() {
        return time;
    }

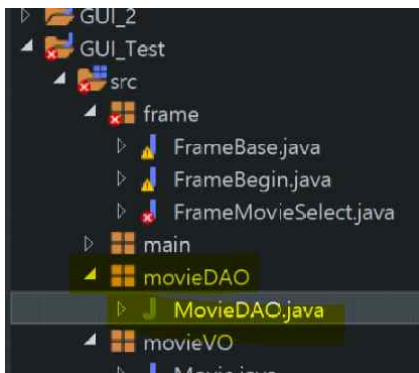
    public void setTime(String time) {
        this.time = time;
    }

    public String getSeat() {
        return seat;
    }

    public void setSeat(String seat) {
        this.seat = seat;
    }
}

```

영화 정보와 예매 현황 관리를 위해 movieDAO패키지 -> MovieDAO클래스 생성 및 내용 추가.



```

public class MovieDAO {
    //영화정보 리스트, 예매 정보 리스트
    private static ArrayList<Movie> list;

```

```
private static ArrayList<MyMovie> reserveList;
```

```
//MovieDAO 기본생성자
```

```
public MovieDAO() {  
    if (list == null) {  
        init();  
    }  
  
    if(reserveList == null){  
        reserveList = new ArrayList<MyMovie>();  
    }  
}
```

```
// 처음에 한번만 실행하는 메서드
```

```
private void init() {  
    list = new ArrayList<Movie>();
```

```
//영화1
```

```
list.add(new Movie( "라라랜드", 10000, 5,  
                    "황홀한 사랑, 순수한 희망, 격렬한 열정... "  
                    +"이 곳에서 모든 감정이 폭발한다! 꿈을 꾸는 사람들을 위  
한 별들의 도시 ‘라라랜드’. 재즈 피아니스트 "  
                    +"‘세바스찬’(라이언 고슬링)과 배우 지망생 ‘미아’(엠마 스  
톤) 인생에서 가장 빛나는 순간 만난 "  
                    +"두 사람은 미완성인 서로의 무대를 만들어가기 시작한  
다.",  
                    "5점\t<이정용>\t감동적 입니다. 쥬륵..\n"));
```

```
//영화2
```

```
list.add(new Movie("말할수없는비밀", 10000, 5, "나는 너를 사랑해!너도 나를 사랑  
하니? "  
                    +"예술학교로 전학 온 상륜(주걸륜)은 아버지의 영향을 받아 피아노에  
천부적인 소질을 보인다. "  
                    +"학교를 둘러보던 중, 신비스러운 피아노 연주가 흘러나오는 옛 음악  
실을 발견하게 되고, "  
                    +"그곳에서 샤오위(계륜미)를 만난다. 그들은 아름다운 피아노 선율처  
럼 즐거운 시간을 보내고, "  
                    +"둘 사이에는 애뜻한 마음이 싹튼다. 그러나 상륜이 샤오위를 더 알  
고 싶어할 때마다 그녀는 비밀이라고 "  
                    +"일관하며 의미심장한 미소만 짓는다. 그러던 어느 날, 샤오위는 상  
륜이 같은 반 여학생 칭요와 뽀뽀하는 "
```


+ "모습을 보게 되고, 그의 곁에서 사라지는데…",

"5점\t<김현진>\t피아노 배틀장면이 인상적이었습니다...☆\n");

//영화3

list.add(new Movie("인사이드아웃", 10000, 4,

 "“괜찮아, 다 잘 될 거야! 우리가 행복하게 만들어 줄게” 모든 사람의 머릿속에 존재하는 감정 컨트롤 본부 "

 +"그곳에서 불철주야 열심히 일하는 기쁨, 슬픔, 버럭, 까칠, 소심 다섯 감정들. 이사 후 새로운 환경에 "

 +"적응해야 하는 ‘라일리’를 위해 그 어느 때 보다 바쁘게 감정의 신호를 보내지만 우연한 실수로 ‘기쁨’과 "

 +"“슬픔’이 본부를 이탈하게 되자 ‘라일리’의 마음 속에 큰 변화가 찾아온다. ‘라일리’가 예전의 모습을 "

 +"되찾기 위해서는 ‘기쁨’과 ‘슬픔’이 본부로 돌아가야만 한다! 그러나 엄청난 기억들이 저장되어 있는 "

 +"머릿속 세계에서 본부까지 가는 길은 험난하기만 한데… 과연, ‘라일리’는 다시 행복해질 수 있을까? "

 +"지금 당신의 머릿속에서 벌어지는 놀라운 일! 하루에도 몇번씩 변하는 감정의 비밀이 밝혀진다!",

"4점\t<박지은>\t애니메이션을 보고 운건 처음이라능...농물\n");

//영화4

list.add(new Movie("겨울왕국", 10000, 5,

 "얼어붙은 세상을 녹일 자매가 온다! 서로가 최고의 친구였던 자매 ‘엘사’와 ‘안나’. 하지만 언니 "

 +"“엘사’에게는 하나뿐인 동생에게조차 말 못할 비밀이 있다. 모든 것을 얼려버리는 신비로운 힘이 바로 그것. "

 +"“엘사’는 통제할 수 없는 자신의 힘이 두려워 왕국을 떠나고, 얼어버린 왕국의 저주를 풀기 위해 ‘안나’는 "

 +"언니를 찾아 환상적인 여정을 떠나는데……",

"5점\t<배정수>\t똑똑똑, 나랑 눈사람 만들래?\n");

//영화5

list.add(new Movie("당신거기있어줄래요", 10000, 2,

 "인생을 뒤바꾼 기적 같은 10번의 기회 “넌 30년 전의 나고, 난 30년 후의 너야” 2015년 현재 “간절히 바라는 소원이 있습니까?” "

 +"현재의 수현(김윤석)은 의료 봉사 활동 중 한 소녀의 생명을 구하고 소녀의 할아버지로부터 신비로운 10개의 알약을 답례로 받는다. "

+ "호기심에 알약을 삼킨 수현은 순간 잠에 빠져들고 다시 눈을 떴을 때, 30년 전 과거의 자신과 마주하게 된다. 1985년 과거 "

+ " "분명 모르는 사람인데... 이상하게 낯이 익었어" 오래된 연인 연아(채서진)와 행복한 나날을 보내던 과거의 수현(변요한)은 "

+ "우연히 길에 쓰러진 남자를 돕게 된다. 남자는 본인이 30년 후의 수현이라 주장하고 황당해하던 과거의 수현은 그가 내미는 증거들을 "

+ "보고 점차 혼란에 빠진다. "과거는 되돌릴 수 없어. 지금 이 순간 역시, 되돌릴 수 없는 시간이고." "당신에겐 과거지만 나한테 "

+ "미래예요. 그 미련 내가 정하는 거고!" 사랑했던 연아를 꼭 한 번 보고 싶었다는 현재 수현의 말에 과거 수현은 알 수 없는 불안감을 "

+ "느끼고 이어 믿기 힘든 미래에 대해 알게 되는데... 그 때로 돌아간다면... 지금의 내 인생도 바뀔 수 있을까요?",

"2점\t<그녀석>\t로맨스 꺼져\n");

//영화6

list.add(new Movie("스파이더맨", 10000, 4,

"전세계를 사로잡은 영웅, 그러나 아무도 몰랐던 그의 이야기. 어릴적 사라진 부모 대신 삼촌 내외와 살고 있는 피터 파커(앤드류 가필드)는 "

+ "여느 고등학생처럼 평범한 학교 생활을 하며 일상을 보내고, 같은 학교 학생 그웬 스테이지(엠마 스톤)와 첫사랑에 빠져 우정과 사랑, "

+ "그리고 둘 만의 비밀을 키워나간다. 그러던 어느 날 아버지가 사용했던 비밀스러운 가방을 발견하고 부모님의 실종사건에 대한 의심을 "

+ "품게 된 그는 그 동안 숨겨져 왔던 과거의 비밀을 추적하게 된다. 아버지의 옛 동료 코너스 박사(리스 이판)의 실험실을 찾아가게 된 "

+ "피터는 우연한 사고로 특별한 능력을 갖게 되고, 뜻밖의 피터의 도움으로 연구를 완성한 코너스 박사는 자신의 숨겨진 자아인 악당 "

+ " "리자드"를 탄생시킨다. 세상을 위협하는 세력앞에 피터는 그의 인생을 통째로 바꾸어 버릴 일생일대의 선택, 바로 '스파이더맨'이라 "

+ "불리우는 영웅이 되기로 결심하는데... 2012년 6월 28일, 스파이더맨의 숨겨진 비밀이 마침내 밝혀진다!",

"4점\t<양현정>\t앤드류가필드 존잘존멋 9x9\n");

//영화7

list.add(new Movie("어벤져스", 10000, 4,

"어벤져스는 끝났다! 희망은 없다! 쉴드의 숙적 히드라는 연구를 통해 새로운 능력자 막시모프 남매를 탄생시키고, 히드라의 기지를 "

+ "공격하는 도중 토니 스타크는 완다 막시모프의 초능력으로 인해 자신이 가장 두려워하는 미래를 보게 된다. 이에 '뉴욕전쟁' 때와 "

+ "같은 사태가 벌어지지 않도록 스타크는 배너 박사와

함께 지구를 지킬 최강의 인공지능 울트론을 탄생시키게 되지만, 울트론은 예상과 "
+"다르게 지배를 벗어나 폭주하기 시작하는데...",

"4점\t<김민재>\t나의 영웅 내 장래희망\n");

//영화8

```
list.add(new Movie("부산행", 10000, 5,  
    "전대미문의 재난이 대한민국을 덮친다! 정체불명의 바이러스가 전국  
으로 확산되고 대한민국 긴급재난경보령이 선포된 가운데, "  
    +"열차에 몸을 실은 사람들은 단 하나의 안전한 도시  
부산까지 살아가기 위한 치열한 사투를 벌이게 된다. 서울에서 부산까지의 거리 "  
    +"442KM 지키고 싶은, 지켜야만 하는 사람들의 극한의  
사투!",
```

"5점\t<한의희>\t공유로 시작해서 공유로 끝난 영화\n");

}//init()

//이름으로 영화 정보를 찾는 메서드

```
public Movie searchMovie(String name) {  
    for (int i = 0; i < list.size(); i++) {  
  
        if (list.get(i).getmName().equals(name)) {  
  
            return list.get(i);  
  
        }  
    }  
    return null;  
}
```

//예매 정보 리스트에 방금 예매한 정보를 등록

```
public void addReserveInfo(MyMovie e){  
    reserveList.add( new MyMovie( e.getmName(),e.getTime(),e.getSeat() ) );  
}
```

//이미 예매된 정보인지 검색

```
public boolean dupSeat(String name,String time,String seat){  
  
    for (int i = 0; i < reserveList.size(); i++) {  
        if (((reserveList.get(i).getmName().equals(name)) &&  
            (reserveList.get(i).getTime().equals(time)))) &&
```



```

ImageIcon movie1 = new ImageIcon("라라랜드.png");
JButton btnMovie1 = new JButton(movie1);
btnMovie1.setName("라라랜드");

btnMovie1.setSize(185,250);
btnMovie1.setLocation(72,30);
add(btnMovie1);

btnMovie1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        Movie movie1 = new MovieDAO().searchMovie("라라랜드");
        //포스터를 클릭해도 아직 상세보기는 제공하지 않고 있다.
    }
});

//영화2의 정보
ImageIcon movie2 = new ImageIcon("말할수없는비밀.jpg");
JButton btnMovie2 = new JButton(movie2);
btnMovie2.setName("말할수없는비밀");

btnMovie2.setSize(185,250);
btnMovie2.setLocation(320,30);
add(btnMovie2);

btnMovie2.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        Movie movie2 = new MovieDAO().searchMovie("말할수없는비밀");
        //포스터를 클릭해도 아직 상세보기는 제공하지 않고 있다.
    }
});

//영화3의 정보
ImageIcon movie3 = new ImageIcon("인사이드아웃.png");
JButton btnMovie3 = new JButton(movie3);
btnMovie3.setName("인사이드아웃");

btnMovie3.setSize(185,250);
btnMovie3.setLocation(72,340);

```

```
add(btnMovie3);
```

```
btnMovie3.addActionListener(new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        Movie movie3 = new MovieDAO().searchMovie("인사이드아웃");
```

```
        //포스터를 클릭해도 아직 상세보기는 제공하지 않고 있다.
```

```
    }
```

```
});
```

```
//영화4의 정보
```

```
ImageIcon movie4 = new ImageIcon("겨울왕국.png");
```

```
JButton btnMovie4 = new JButton(movie4);
```

```
btnMovie4.setName("겨울왕국");
```

```
btnMovie4.setSize(185,250);
```

```
btnMovie4.setLocation(320,340);
```

```
add(btnMovie4);
```

```
btnMovie4.addActionListener(new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        Movie movie4 = new MovieDAO().searchMovie("겨울왕국");
```

```
        //포스터를 클릭해도 아직 상세보기는 제공하지 않고 있다.
```

```
    }
```

```
});
```

```
//하단의 버튼
```

```
JPanel bottomSet = new JPanel();
```

```
bottomSet.setBounds(0, 660, 600, 100);
```

```
bottomSet.setLayout(null);
```

```
bottomSet.setBackground(new Color(0xFFD700));
```

```
JButton btnBack = new JButton("뒤로가기");
```

```
btnBack.setBackground(new Color(0xA6A6A6));
```

```
btnBack.setSize(183, 87);
```

```
btnBack.setLocation(5, 0);
```

```

btnBack.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnBack);

btnBack.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        FrameBase.getInstance(new FrameBegin());
    }
});

JButton btnNext = new JButton("다음 페이지");

btnNext.setBackground(new Color(0xA6A6A6));
btnNext.setSize(183, 87);
btnNext.setLocation(391, 0);
btnNext.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnNext);

btnNext.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        //다음페이지로 넘어가기 기능 만들어줘야 한다.
        //FrameMovieSelect2 클래스를 만들어서 1번과 동일하게 내용 처리
        FrameBase.getInstance( new FrameMovieSelect2() );
    }
});

add(bottomSet, BorderLayout.SOUTH);

} //생성자
}

```

FrameMovieSelect2 클래스 생성하고 내용 추가



```

public class FrameMovieSelect2 extends JPanel {
    //생성자에 내용 추가
    public FrameMovieSelect2() {
        //JPanel 구조
        setBackground(new Color(0xFFD700));
        setLayout(null);
        setSize(600,2000);

        //영화5의 정보
        ImageIcon movie5 = new ImageIcon("당신거기있어줄래요.png");
        JButton btnMovie5 = new JButton(movie5);
        btnMovie5.setName("당신거기있어줄래요");

        btnMovie5.setSize(185,250);
        btnMovie5.setLocation(72,30);
        add(btnMovie5);

        btnMovie5.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                Movie movie5 = new MovieDAO().searchMovie("당신거기있어줄래요");
            }
        });

        //영화6의 정보
        ImageIcon movie6 = new ImageIcon("스파이더맨.png");
        JButton btnMovie6 = new JButton(movie6);
        btnMovie6.setName("스파이더맨");

        btnMovie6.setSize(185,250);
        btnMovie6.setLocation(320,30);
        add(btnMovie6);

        btnMovie6.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                Movie movie6 = new MovieDAO().searchMovie("스파이더맨");
            }
        });
    }
}

```


//영화7의 정보

```
ImageIcon movie7 = new ImageIcon("어벤져스.png");  
JButton btnMovie7 = new JButton(movie7);  
btnMovie7.setName("어벤져스");
```

```
btnMovie7.setSize(185,250);  
btnMovie7.setLocation(72,340);  
add(btnMovie7);
```

```
btnMovie7.addActionListener(new ActionListener() {  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        Movie movie7 = new MovieDAO().searchMovie("어벤져스");  
    }  
});
```

//영화8의 정보

```
ImageIcon movie8 = new ImageIcon("부산행.png");  
JButton btnMovie8 = new JButton(movie8);  
btnMovie8.setName("부산행");
```

```
btnMovie8.setSize(185,250);  
btnMovie8.setLocation(320,340);  
add(btnMovie8);
```

```
btnMovie8.addActionListener(new ActionListener() {  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        Movie movie8 = new MovieDAO().searchMovie("부산행");  
    }  
});
```

//하단의 버튼 설정

```
JPanel bottomSet = new JPanel();  
  
bottomSet.setBounds(0, 660, 600, 100);  
bottomSet.setLayout(null);  
bottomSet.setBackground(new Color(0xFFD700));
```

```

JButton btnBack = new JButton("이전 페이지");

btnBack.setBackground(new Color(0xA6A6A6));
btnBack.setSize(183, 87);
btnBack.setLocation(5, 0);
btnBack.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnBack);

btnBack.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        FrameBase.getInstance(new FrameMovieSelect());
    }
});

JButton btnHome = new JButton("처음 화면으로");

btnHome.setBackground(new Color(0xA6A6A6));
btnHome.setSize(183, 87);
btnHome.setLocation(198, 0);
btnHome.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnHome);

btnHome.addActionListener(new ActionListener() {

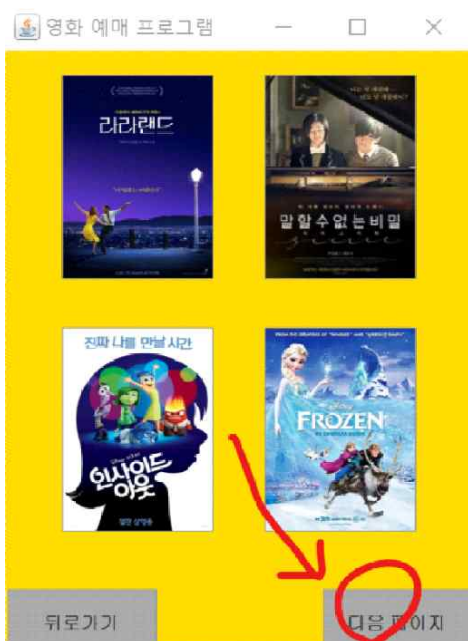
    @Override
    public void actionPerformed(ActionEvent e) {
        FrameBase.getInstance(new FrameBegin());
    }
});

add(bottomSet, BorderLayout.SOUTH);

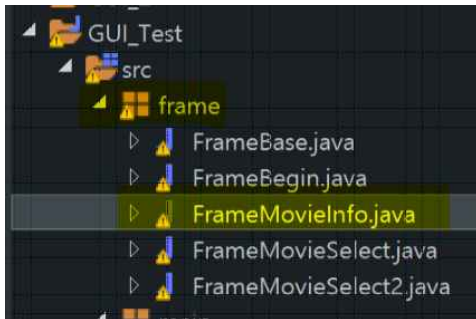
} //생성자
}

```

여기까지 실행 해서 다음페이지로 전환이 잘 되는지 확인해보기



이제 각 영화를 클릭하여 상세보기 페이지로 이동시켜 보자.
 frame패키지 -> FrameMovieInfo 클래스 생성 및 내용 추가.



```
public class FrameMovieInfo extends JPanel {

    public FrameMovieInfo( Movie m ) { //클릭한 이미지의 정보를 생성자 파라미터로 받는다.
        // JPanel 구조
        setBackground(new Color(250, 244, 192));
        setLayout(null);
        setSize(600, 800);

        // 포스터 부분
        final int movieNum = 8;
        int movieflag = 0;
        String[] movieurl = {
            "라라랜드2.png", "말할수없는비밀2.jpg", "인사이드아웃2.png",
            "겨울왕국2.png", "당신거기있어줄래요2.png", "스파이더맨2.png",
            "어벤져스2.png", "부산행2.png" };

        ImageIcon[] movieImg = new ImageIcon[movieNum];
```

```

for (int i = 0; i < movieNum; i++) {

    movieImg[i] = new ImageIcon( movieurl[i] );
    if (m.getName().equals("라라랜드")) {
        movieflag = 0;
    } else if (m.getName().equals("말할수없는비밀")) {
        movieflag = 1;
    } else if (m.getName().equals("인사이드아웃")) {
        movieflag = 2;
    } else if (m.getName().equals("겨울왕국")) {
        movieflag = 3;
    } else if (m.getName().equals("당신거기있어줄래요")) {
        movieflag = 4;
    } else if (m.getName().equals("스파이더맨")) {
        movieflag = 5;
    } else if (m.getName().equals("어벤져스")) {
        movieflag = 6;
    } else if (m.getName().equals("부산행")) {
        movieflag = 7;
    }
}

JLabel img = new JLabel( movieImg[movieflag] );
img.setBounds(50, 13, 285, 350);
add(img);

// 영화 관련 정보 텍스트
TextArea story = new TextArea(
    "\n◎영화제목" + "\n" + "    " + m.getName() +
    "\n\n" + "◎평    점" + "\n" + "    " +
    String.format("%.1f", m.getScore()) + " / 5 점\n\n" +
    "◎줄 거 리" + "\n" + "    " + m.getStory(),
    0, 0, TextArea.SCROLLBARS_VERTICAL_ONLY);
story.setSize(580, 280);
story.setLocation(0, 375);
story.setBackground(new Color(0xFFD700));
story.setFont(new Font(Font.DIALOG, Font.BOLD, 20));
add(story);

// 등록된 후기 보기

```

```

JButton btnReview = new JButton("리뷰 보기");

btnReview.setBackground(new Color(229, 216, 92));
btnReview.setSize(150, 40);
btnReview.setLocation(400, 325);
btnReview.setFont(new Font("나눔고딕코딩", Font.BOLD, 18));

btnReview.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        //리뷰보기 기능 추가 예정
    }
});
add(btnReview);

//하단의 버튼
JPanel bottomSet = new JPanel();

bottomSet.setBounds(0, 660, 600, 100);
bottomSet.setLayout(null);
bottomSet.setBackground(new Color(0xFFD700));

JButton btnBack = new JButton("뒤로가기");
btnBack.setBackground(new Color(0xA6A6A6));
btnBack.setSize(183, 87);
btnBack.setLocation(5, 0);
btnBack.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnBack);

int buttonNum = movieflag;
btnBack.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        if (0 <= buttonNum && buttonNum <= 3) {
            FrameBase.getInstance(new FrameMovieSelect());
        } else {
            FrameBase.getInstance(new FrameMovieSelect2());
        }
    }
});

```

```

});

JButton btnHome = new JButton("처음 화면으로");

btnHome.setBackground(new Color(0xA6A6A6));
btnHome.setSize(183, 87);
btnHome.setLocation(198, 0);
btnHome.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnHome);

btnHome.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        FrameBase.getInstance(new FrameBegin());
    }
});

JButton btnBuy = new JButton("예매하기");

btnBuy.setBackground(new Color(0xA6A6A6));
btnBuy.setSize(183, 87);
btnBuy.setLocation(391, 0);
btnBuy.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnBuy);

btnBuy.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        //예매하기 기능 추가 예정
    }
});

add(bottomSet, BorderLayout.SOUTH);

} //생성자
}

```

[포스터 클릭시 상세보기 페이지로 잘 전환 되는지 확인](#)



이제 리뷰 보기 페이지를 만들어보자.

frame패키지 -> FrameReviewRead클래스 생성하고 내용 추가



```
public class FrameReviewRead extends JFrame {
```

```
    public FrameReviewRead( Movie m ){ //리뷰를 보고싶은 영화 정보를 파라미터로 받아야한다

        //후기 출력 창
        TextArea review =
new TextArea("평점\t이름\t\t\t한줄평\n", 0, 0,TextArea.SCROLLBARS_VERTICAL_ONLY);
        review.setEditable(false);
        review.append(m.getReview());
        review.setSize(800,300);
        review.setLocation(700,200);
        review.setFont(new Font("맑은고딕",Font.PLAIN,18));
        add(review);

        //새로운 서브 프레임 구현
        Frame fr= new Frame();
        fr.setTitle(m.getmName()+"의 리뷰");
        fr.setSize(800, 300);
        fr.setLocation(700, 200);
        fr.add(review);
```

```
fr.setVisible(true);
```

//리뷰창의 x를 눌렀을 때 System.exit(0)를 하면 예매프로그램 자체가 종료되므로
//dispose()를 통해 서브 프레임만 종료해줘야 한다.

```
fr.addWindowListener(new WindowAdapter() {  
    @Override  
    public void windowClosing(WindowEvent e) {  
        fr.setVisible(false);  
        fr.dispose();  
    }  
});  
} //생성자  
}
```

FrameMovieInfo 클래스에 리뷰 호출 코드 추가

```
public class FrameMovieInfo extends JPanel {  
    public FrameMovieInfo(Movie m) {  
  
        // JPanel 구조  
        setBackground(new Color(250, 244, 192));  
        setLayout(null);  
        setSize(600, 800);  
  
        .....  
  
        // 등록된 후기 보기  
        JButton btnReview = new JButton("리뷰 보기");  
  
        btnReview.setBackground(new Color(229, 216, 92));  
        btnReview.setSize(150, 40);  
        btnReview.setLocation(400, 325);  
        btnReview.setFont(new Font("나눔고딕코딩", Font.BOLD, 18));  
  
        btnReview.addActionListener(new ActionListener() {  
  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                new FrameReviewRead(m);  
            }  
        });  
        add(btnReview);  
    }  
}
```



```
//하단의 버튼
```

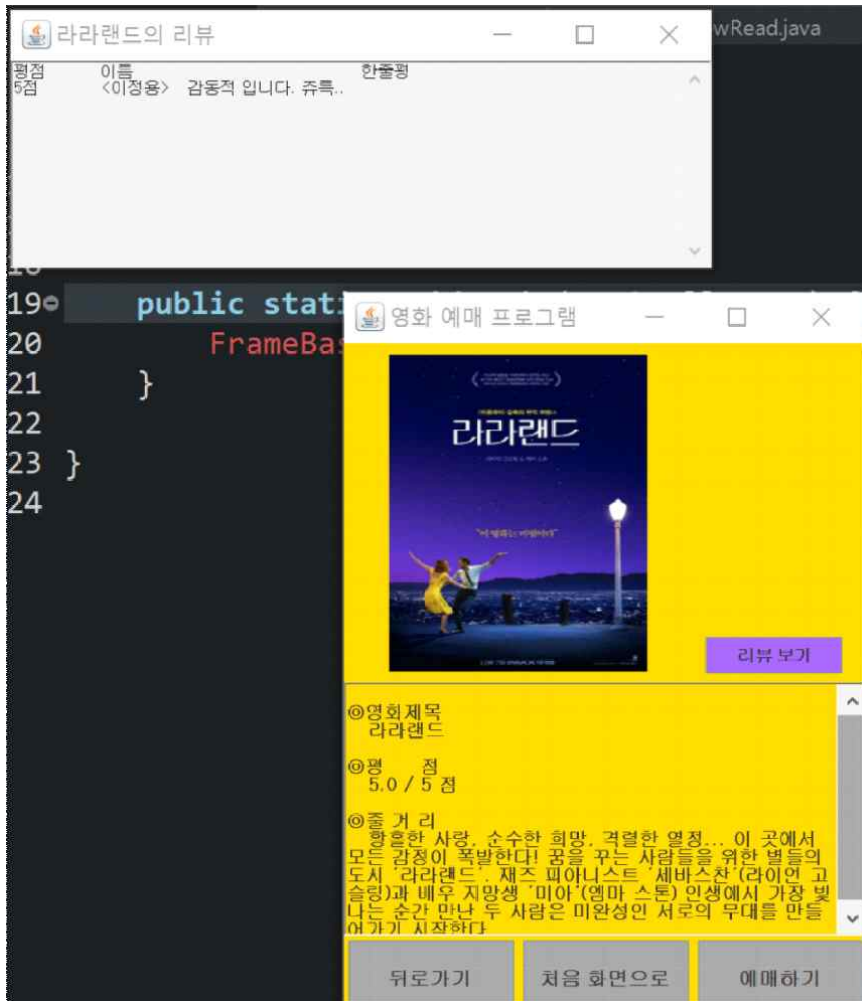
```
JPanel bottomSet = new JPanel();
```

```
.....
```

```
}//생성자
```

```
}
```

실행 해서 리뷰보기 가능한지 확인.



이제 예매하기 기능을 추가해보자!!!

FrameMovieInfo 에 예매하기 버튼 클릭 처리

```
public class FrameMovieInfo extends JPanel {
```

```
    public FrameMovieInfo(Movie m) {
```

```
        .....
```

```
        JButton btnBuy = new JButton("예매하기");
```

```
        .....
```

```
        btnBuy.addActionListener(new ActionListener() {
```

```

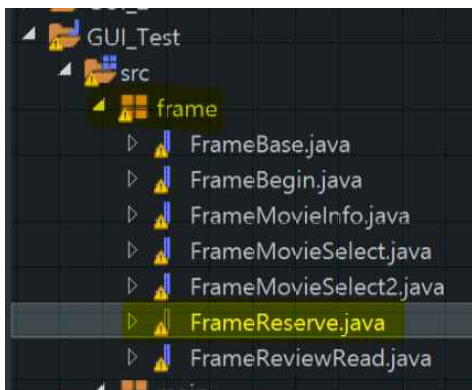
        @Override
        public void actionPerformed(ActionEvent e) {
            FrameBase.getInstance(new FrameReserve(m));
        }
    });

    add(bottomSet, BorderLayout.SOUTH);

    } //생성자
}

```

frame패키지 -> FrameReserve클래스 생성 및 내용 추가



```

public class FrameReserve extends JPanel {
    public FrameReserve( Movie m ) {
        //JPanel 구조
        setBackground(new Color(0xFFD700));
        setLayout(null);
        setSize(600, 800);

        JPanel top = new JPanel();
        top.setBackground(new Color(0xFFD700));
        top.setLayout(null);
        top.setSize(600, 300);
        top.setLocation(0, 0);

        JPanel center = new JPanel();
        center.setBackground(new Color(0xFFD700));
        center.setLayout(new GridLayout(5, 8));
        center.setSize(600, 350);
        center.setLocation(0, 305);

        JPanel bottom = new JPanel();
    }
}

```

```
bottom.setBackground(new Color(0xFFD700));
bottom.setLayout(null);
bottom.setSize(600, 200);
bottom.setLocation(0, 660);
```

```
//상단 JPanel(영화 포스터,영화 이름, 시간대 선택)
```

```
final int movieNum = 8;
int movieflag = 0;
String[] movieurl = {
    "라라랜드.png", "말할수없는비밀.jpg", "인사이드아웃.png",
    "겨울왕국.png", "당신거기있어줄래요.png", "스파이더맨.png",
    "어벤져스.png", "부산행.png" };

```

```
ImageIcon[] movieImg = new ImageIcon[movieNum];
```

```
for (int i = 0; i < movieNum; i++) {
    movieImg[i] = new ImageIcon(movieurl[i]);
    if (m.getName().equals("라라랜드")) {
        movieflag = 0;
    } else if (m.getName().equals("말할수없는비밀")) {
        movieflag = 1;
    } else if (m.getName().equals("인사이드아웃")) {
        movieflag = 2;
    } else if (m.getName().equals("겨울왕국")) {
        movieflag = 3;
    } else if (m.getName().equals("당신거기있어줄래요")) {
        movieflag = 4;
    } else if (m.getName().equals("스파이더맨")) {
        movieflag = 5;
    } else if (m.getName().equals("어벤져스")) {
        movieflag = 6;
    } else if (m.getName().equals("부산행")) {
        movieflag = 7;
    }
}
}

```

```
JLabel img = new JLabel(movieImg[movieflag]);
img.setBounds(50, 15, 175, 248);
```

```
JLabel name = new JLabel(m.getName());
name.setFont(new Font("나눔고딕코딩", Font.BOLD, 32));
```

```
name.setBounds(260, 35, 300, 175);
```

```
Choice movieTime = new Choice();  
long time = System.currentTimeMillis();
```

```
SimpleDateFormat dayTime =  
    new SimpleDateFormat("MM월 dd일(EE)");//(EE:요일)
```

```
String str = dayTime.format( new Date(time) );
```

```
String[] timeSlot = {  
    "시간대 선택",  
    str + " 06:30~08:30",  
    str + " 09:00~11:00",  
    str + " 11:30~13:30",  
    str + " 14:00~16:00",  
    str + " 16:30~18:30",  
    str + " 19:00~21:30",  
    str + " 22:00~00:00",  
    str + " 00:30~02:30" };
```

```
for (int i = 0; i < timeSlot.length; i++) {  
    movieTime.add(timeSlot[i]);  
}
```

```
movieTime.setFont(new Font("나눔고딕코딩", Font.PLAIN, 18));  
movieTime.setSize(270, 200);  
movieTime.setLocation(260, 200);
```

```
JLabel screen = new JLabel("[스 크 린]");  
screen.setFont(new Font("나눔고딕코딩", Font.BOLD, 32));  
screen.setBounds(180, 223, 240, 120);  
screen.setVisible(false);
```

```
//중반부 JPanel(좌석 선택)
```

```
ButtonGroup bg = new ButtonGroup();  
final int seatNum = 8 * 5;  
JRadioButton[] seatChoice = new JRadioButton[seatNum];  
String[] seatn = { "1", "2", "3", "4", "5", "6", "7", "8" };  
String seatAl = "A";
```

```

//a1 ~ a8
//....
//e1 ~ e8 까지 5행 8열의 좌석배치를 화면에 그린다.
for (int i = 0; i < 5; i++) {

    if (i == 1) {
        seatAl = "B";
    } else if (i == 2) {
        seatAl = "C";
    } else if (i == 3) {
        seatAl = "D";
    } else if (i == 4) {
        seatAl = "E";
    }

    for (int j = 0; j < 8; j++) {
        seatChoice[(i * 8) + j] = new JRadioButton( seatAl + seatn[j] );
        bg.add(seatChoice[(i * 8) + j]);
        center.add(seatChoice[(i * 8) + j]);
        seatChoice[(i * 8) + j].setVisible(false);
    }

}

}

//하단의 버튼
JButton btnBack = new JButton("뒤로가기");

btnBack.setBackground(new Color(0xA6A6A6));
btnBack.setSize(183, 87);
btnBack.setLocation(5, 0);
btnBack.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottom.add(btnBack);

btnBack.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        FrameBase.getInstance(new FrameMovieInfo(m));
    }

});

```

```

JButton btnHome = new JButton("처음 화면으로");

btnHome.setBackground(new Color(0xA6A6A6));
btnHome.setSize(183, 87);
btnHome.setLocation(198, 0);
btnHome.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottom.add(btnHome);

btnHome.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        FrameBase.getInstance(new FrameBegin());
    }
});

```

```

JButton btnBuy = new JButton("결제하기");

btnBuy.setBackground(new Color(0xA6A6A6));
btnBuy.setSize(183, 87);
btnBuy.setLocation(391, 0);
btnBuy.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottom.add(btnBuy);

```

//결제하기 버튼 유효성 체크

```

btnBuy.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {

        MyMovie myM = new MyMovie();
        myM.setmName(m.getmName());
        myM.setTime(movieTime.getSelectedItem());
        int seatIndex = -1;
        for (int i = 0; i < seatNum; i++) {
            if (seatChoice[i].isSelected()) {
                seatIndex = i;
            }
        }

        if (movieTime.getSelectedItem().equals("시간대 선택")) {

```

```

        JOptionPane.showMessageDialog(
            null, "시간을 선택해 주세요");

    } else if (seatIndex == -1) {
        JOptionPane.showMessageDialog(
            null, "좌석을 선택해 주세요");

    } else {
        myM.setSeat(seatChoice[seatIndex].getText());
        //결제 성공 코드 추가 예정
    }
}

}); //addActionListener

top.add(movieTime);
top.add(img);
top.add(name);
top.add(screen);

add(top);
add(center);
add(bottom);

//이미 등록된 예매 정보인지 확인 및 정확한 예매 정보인지 확인
movieTime.addItemListener(new ItemListener() {

    @Override
    public void itemStateChanged(ItemEvent e) {
        for (int i = 0; i < seatNum; i++) {
            seatChoice[i].setEnabled(true);
        }

        //choice가 “시간대 선택”으로 되어 있을때는 좌석메뉴 비활성화
        if (e.getItem().equals("시간대 선택")) {

            for (int i = 0; i < seatNum; i++) {
                seatChoice[i].setVisible(false);
                screen.setVisible(false);
            }
        }
    }
});

```

```

    } else {

        //그렇지 않다면 좌석메뉴 활성화
        for (int i = 0; i < seatNum; i++) {
            seatChoice[i].setVisible(true);
            screen.setVisible(true);
        }
    }

    MovieDAO setSeat = new MovieDAO();
    for (int j = 0; j < seatNum; j++) {

        if (setSeat.dupSeat(
            m.getmName(),
            e.getItem().toString(),
            seatChoice[j].getText())) {

            //이미 예약된 좌석은 비활성화
            seatChoice[j].setEnabled(false);
        }
    }
    } //for
    } //itemStateChanged
    }); //addItemListener
    } //생성자
}

```

실행 해서 예매하기 페이지 접근 되는지 확인



이제 결제하기 페이지를 만들어보자.

FrameReserve 클래스에 결제기능 추가

```
public class FrameReserve extends JPanel {

    public FrameReserve(Movie m) {

        .....

        //결제하기 버튼 유효성 체크
        btnBuy.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {

                .....

                if (movieTime.getSelectedItem().equals("시간대 선택")) {
                    JOptionPane.showMessageDialog(
                        null, "시간을 선택해 주세요");

                } else if (seatIndex == -1) {
                    JOptionPane.showMessageDialog(
                        null, "좌석을 선택해 주세요");

                } else {
                    myM.setSeat(seatChoice[seatIndex].getText());
                    FrameBase.getInstance(
                        new FrameMovieConfirm(m, myM));
                }
            }
        }); //addActionListener
    }
}
```

```

top.add(movieTime);
top.add(img);
top.add(name);
top.add(screen);

```

```

add(top);
add(center);
add(bottom);
.....

```

```

} //생성자

```

```

}

```

FrameMovieConfirm클래스가 없으니 이를 생성하고 내용을 추가하자



```

public class FrameMovieConfirm extends JPanel {

```

```

    public FrameMovieConfirm( Movie m, MyMovie e ) {

```

```

        //JPanel 구조

```

```

        setBackground(new Color(0xFFD700));

```

```

        setLayout(null);

```

```

        setSize(600, 800);

```

```

        //영화 포스터

```

```

        final int movieNum = 8;

```

```

        int movieflag = 0;

```

```

        String[] movieurl = {

```

```

            "라라랜드2.png", "말할수없는비밀2.jpg", "인사이드아웃2.png",

```

```

            "겨울왕국2.png", "당신거기있어줄래요2.png", "스파이더맨2.png",

```

```

            "어벤져스2.png", "부산행2.png" };

```

```

        ImageIcon[] movieImg = new ImageIcon[movieNum];

```

```

        for (int i = 0; i < movieNum; i++) {

```



```
bottomSet.setBounds(0, 660, 600, 100);
bottomSet.setLayout(null);
bottomSet.setBackground(new Color(0xFFD700));
```

```
JButton btnBack = new JButton("뒤로가기");
```

```
btnBack.setBackground(new Color(0xA6A6A6));
btnBack.setSize(183, 87);
btnBack.setLocation(5, 0);
btnBack.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnBack);
```

```
btnBack.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        FrameBase.getInstance(new FrameReserve(m));
    }
});
```

//최종결제 버튼

```
JButton btnBuy = new JButton("결제하기");
```

```
btnBuy.setBackground(new Color(0xA6A6A6));
btnBuy.setSize(183, 87);
btnBuy.setLocation(391, 0);
btnBuy.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnBuy);
```

//결제하기 버튼 클릭

```
btnBuy.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent a) {
        new MovieDAO().addReserveInfo(e);
        JOptionPane.showMessageDialog(
            null, "영화제목 : " + m.getmName() + "\n" +
                "영화시간 : " + e.getTime() + "\n" +
                "좌석 : " + e.getSeat() + "\n" +
                "금액 : " + m.getPrice() + "원 입니다.");
        //결제를 마치고 첫 페이지로 보낸다.
    }
});
```

```

FrameBase.getInstance(new FrameBegin());
    }
});

add(bottomSet, BorderLayout.SOUTH);

} //생성자
}

```

결제가 잘 되는지 확인 해 보고, 같은 시간대의 결제화면에서 방금 예매한 좌석이 비활성화 되어있는지도 확인.



예매관련 기능은 마무리가 되었고, 이제 평가하기 기능을 추가해보자.



FrameBegin에 평가하기 버튼 클릭 감지자 추가

```

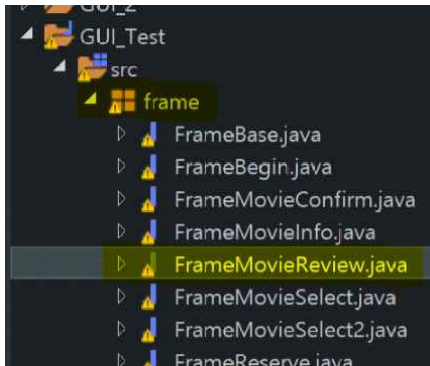
public class FrameBegin extends JPanel {
    public FrameBegin() {
        .....
        //버튼 이벤트(예매창으로 / 후기남기기로)
        btnInfo.addActionListener(new ActionListener() {
            .....
        });

        btnPost.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                FrameBase.getInstance( new FrameMovieReview() );
            }
        });
    }
}

```

FrameMovieReview 클래스 생성 및 내용 추가



```

public class FrameMovieReview extends JPanel {
    public FrameMovieReview() {
        // JPanel 기본 구조
        setBackground(new Color(0xFFD700));
        setLayout(null);
        setSize(600, 800);

        // 상단부 이미지
        ImageIcon popcon = new ImageIcon("popcon2.png");
        JLabel lblPopcon = new JLabel(popcon);
        lblPopcon.setSize(600, 280);
        lblPopcon.setLocation(0, 0);
        add(lblPopcon);
    }
}

```

```
// 후기 남길 영화 선택(라디오버튼)
```

```
JPanel dis2 = new JPanel(); // 평점 패널 생성
```

```
dis2.setLayout(new GridLayout(3, 3));
```

```
dis2.setBounds(0, 280, 600, 220);
```

```
dis2.setBorder(
```

```
    new TitledBorder( null, //테두리없음
```

```
    "평점 줄 영화 선택", //내용
```

```
    TitledBorder.LEADING, //텍스트 정렬방식:양쪽정렬
```

```
    TitledBorder.TOP, //텍스트 표시 위치 : 상단
```

```
    null, //폰트 지정안함
```

```
    new Color(255, 122, 144) )); //색상
```

```
JRadioButton movie1 = new JRadioButton("라라랜드");
```

```
JRadioButton movie2 = new JRadioButton("말할수없는비밀");
```

```
JRadioButton movie3 = new JRadioButton("인사이드아웃");
```

```
JRadioButton movie4 = new JRadioButton("겨울왕국");
```

```
JRadioButton movie5 = new JRadioButton("당신거기있어줄래요");
```

```
JRadioButton movie6 = new JRadioButton("스파이더맨");
```

```
JRadioButton movie7 = new JRadioButton("어벤져스");
```

```
JRadioButton movie8 = new JRadioButton("부산행");
```

```
add(dis2);
```

```
// 평점 점수 받기 (라디오 버튼)
```

```
JPanel dis1 = new JPanel(); // 라디오 버튼, 한줄평 받을 패널 생성
```

```
dis1.setBorder(new TitledBorder(
```

```
    null, //테두리없음
```

```
    "리뷰남기기", TitledBorder.LEADING, //텍스트 정렬방식:양쪽정렬
```

```
    TitledBorder.TOP, //텍스트 표시 위치 : 상단
```

```
    null, //폰트 지정안함
```

```
    null)); //색상 지정안함
```

```
dis1.add(new JLabel("평점 "));
```

```
JRadioButton point1 = new JRadioButton("1점");
```

```
JRadioButton point2 = new JRadioButton("2점");
```

```
JRadioButton point3 = new JRadioButton("3점");
```

```
JRadioButton point4 = new JRadioButton("4점");
```

```
JRadioButton point5 = new JRadioButton("5점");
```

```
dis1.add(point1);
```

```
dis1.add(point2);
```

```
dis1.add(point3);
dis1.add(point4);
dis1.add(point5);
```

```
ButtonGroup group2 = new ButtonGroup();
group2.add(point1);
group2.add(point2);
group2.add(point3);
group2.add(point4);
group2.add(point5);
```

```
dis1.setSize(600, 120);
dis1.setLocation(0, 500);
```

```
// 작성자 이름
```

```
TextField reviewName = new TextField("작성자 이름", 40);
reviewName.setBounds(0, 430, 50, 100);
dis1.add(reviewName);
```

```
// 한줄 평 입력받기
```

```
TextField review = new TextField("한줄평을 입력해주세요.(30 글자 미만)", 50);
review.setBounds(0, 430, 400, 100);
dis1.add(review);
```

```
JButton btn = new JButton("완료");
btn.setBounds(400, 430, 200, 100);
dis1.add(btn);
btn.setEnabled(false);
```

```
add(dis1);
dis1.setVisible(false);
```

```
// 영화 라디오버튼 정보
```

```
String mName[] = { "라라랜드", "말할수없는비밀", "인사이드아웃",
                  "겨울왕국", "당신거기있어줄래요", "스파이더맨",
                  "어벤져스", "부산행" };

```

```
ButtonGroup group1 = new ButtonGroup();
JRadioButton movie[] = new JRadioButton[8];
```

```
for (int i = 0; i < mName.length; i++) {
```



```
movie[i] = new JRadioButton(mName[i]);
movie[i].setFont(new Font("나눔고딕코딩", Font.BOLD, 16));
```

```
dis2.add(movie[i]);
group1.add(movie[i]);
```

```
//각 라디오버튼에 클릭 이벤트 감지
```

```
movie[i].addActionListener(new ActionListener() {
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
//어떤 라디오버튼이든 선택하는 순간 한줄평 작성폼이 화면에 보여진다.
```

```
    dis1.setVisible(true);
```

```
        String[] movieurl = {
```

```
        "라라랜드.png", "말할수없는비밀.jpg", "인사이드아웃.png",
```

```
        "겨울왕국.png", "당신거기있어줄래요.png", "스파이더맨.png",
```

```
        "어벤져스.png", "부산행.png" };
```

```
//평가를 위해 선택한 영화를 index로 기억한다
```

```
int movieIndex = 0;
```

```
if (e.getActionCommand().equals("라라랜드")) {
```

```
    movieIndex = 0;
```

```
} else if (e.getActionCommand().equals("말할수없는비밀")) {
```

```
    movieIndex = 1;
```

```
} else if (e.getActionCommand().equals("인사이드아웃")) {
```

```
    movieIndex = 2;
```

```
} else if (e.getActionCommand().equals("겨울왕국")) {
```

```
    movieIndex = 3;
```

```
} else if (e.getActionCommand().equals("당신거기있어줄래요")) {
```

```
    movieIndex = 4;
```

```
} else if (e.getActionCommand().equals("스파이더맨")) {
```

```
    movieIndex = 5;
```

```
} else if (e.getActionCommand().equals("어벤져스")) {
```

```
    movieIndex = 6;
```

```
} else if (e.getActionCommand().equals("부산행")) {
```

```
    movieIndex = 7;
```

```
}
```

```
}//actionPerformed
```

```
});
```

```
}//for
```

// 한줄 평 등록(30글자 이상, 빈 문자열은 등록불가)

```
review.addTextListener(new TextListener() {

    @Override
    public void textValueChanged(TextEvent e) {
        if (review.getText().equals("")) {
            btn.setEnabled(false);

        } else if (review.getText().length() > 30) {
            btn.setEnabled(false);

        } else if (review.getText().equals("한줄평을 입력해주세요.(30 글자 미만)") {
            btn.setEnabled(false);

        } else {
            //1 ~ 30자 범위의 평가가 있을때만 완료버튼 활성화
            btn.setEnabled(true);
        }
    }
});
```

// 작성자 이름 창 클릭시 텍스트 힌트 제거

```
reviewName.addFocusListener(new FocusListener() {

    @Override
    public void focusLost(FocusEvent e) {
    }

    @Override
    public void focusGained(FocusEvent e) {
        if (reviewName.getText().equals("작성자 이름")) {
            reviewName.setText("");
        }
    }
});
```

// 한줄 평 창 클릭시 텍스트 힌트 제거

```
review.addFocusListener(new FocusListener() {
```

```
    @Override
```

```

        public void focusLost(FocusEvent e) {
        }

        @Override
        public void focusGained(FocusEvent e) {
            if (review.getText().equals("한줄평을 입력해주세요.(30 글자 미만)")) {
                review.setText("");
            }
        }
    });

```

// 완료 버튼

```

btn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        if (point1.isSelected() || point2.isSelected() ||
            point3.isSelected() || point4.isSelected() ||
            point5.isSelected()) {

            if (reviewName.getText().equals("")) {
                reviewName.setText("익명");

            } else if (reviewName.getText().equals("작성자 이름")) {
                reviewName.setText("익명");
            }
        }
    }
}

```

//몇 점 줬는지 확인하기

```

int num = 0;
if (point1.isSelected()) {
    num = 1;
} else if (point2.isSelected()) {
    num = 2;
} else if (point3.isSelected()) {
    num = 3;
} else if (point4.isSelected()) {
    num = 4;
} else if (point5.isSelected()) {
    num = 5;
}

```

```

        for (int i = 0; i < movie.length; i++) {
            if (movie[i].isSelected()) {
                //작성한 후기를 영화 리스트에 추가
                new MovieDAO().inputReview(
                    movie[i].getText(), reviewName.getText(),
                    review.getText(), num);
            }
        }
    } //for

    JOptionPane.showMessageDialog(
        null, "리뷰 작성을 완료하였습니다.");
    FrameBase.getInstance(new FrameBegin());

    } else {
        //점수를 주지 않은경우
        JOptionPane.showMessageDialog(
            null, "평점을 입력해주세요.");
    }
} //actionPerformed
});

// 뒤로가기 버튼
JPanel bottomSet = new JPanel();
bottomSet.setBounds(0, 660, 600, 100);
bottomSet.setLayout(null);
bottomSet.setBackground(new Color(0xFFD700));
JButton btnBack = new JButton("뒤로가기");
btnBack.setBackground(new Color(0xA6A6A6));
btnBack.setSize(183, 87);
btnBack.setLocation(5, 0);
btnBack.setFont(new Font("나눔고딕코딩", Font.BOLD, 22));
bottomSet.add(btnBack);

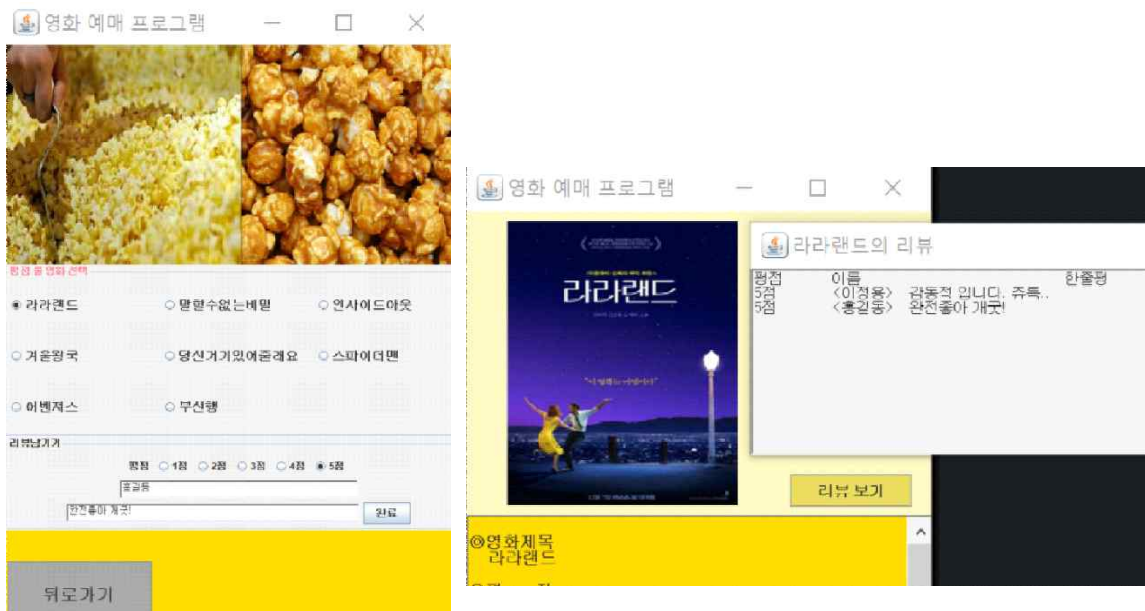
btnBack.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        FrameBase.getInstance(new FrameBegin());
    }
});

add(bottomSet, BorderLayout.SOUTH);

```

}//생성자

다음과 같이 리뷰를 남기고 해당 영화에 리뷰가 추가되어있는지 확인하고 마무리하자



끝~!!