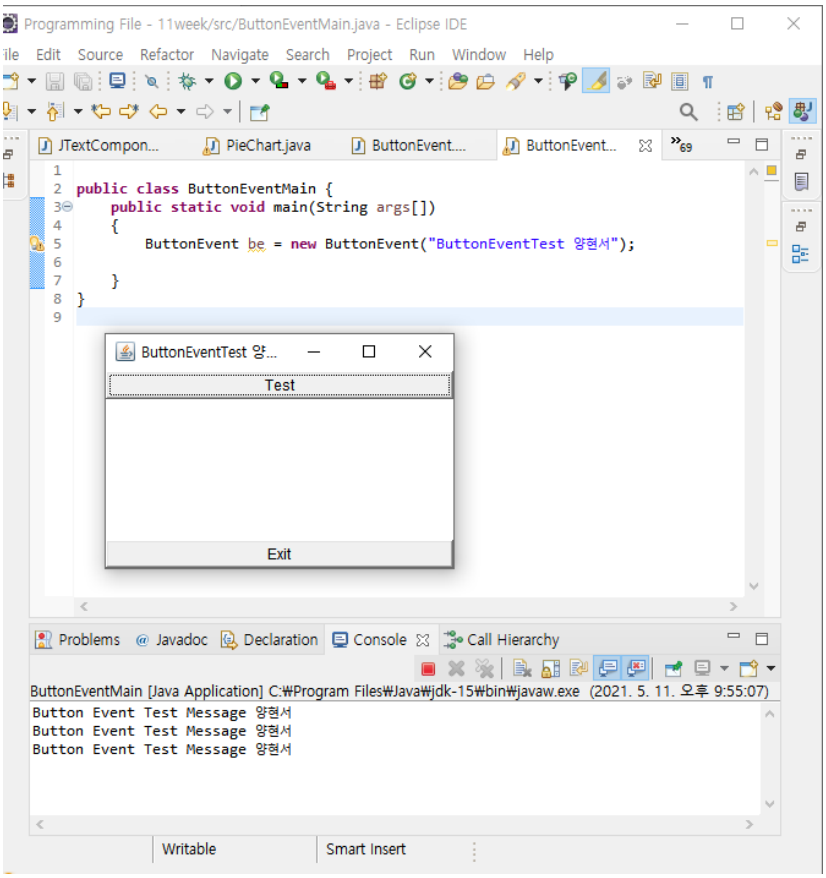
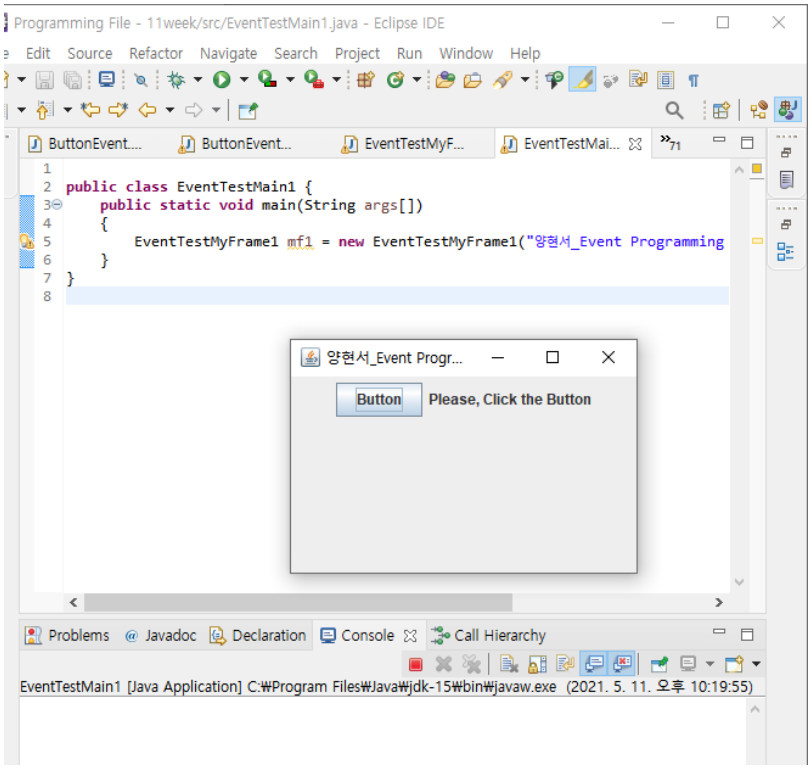
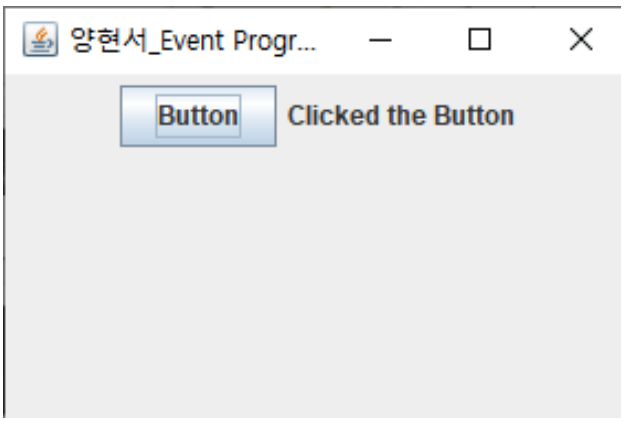
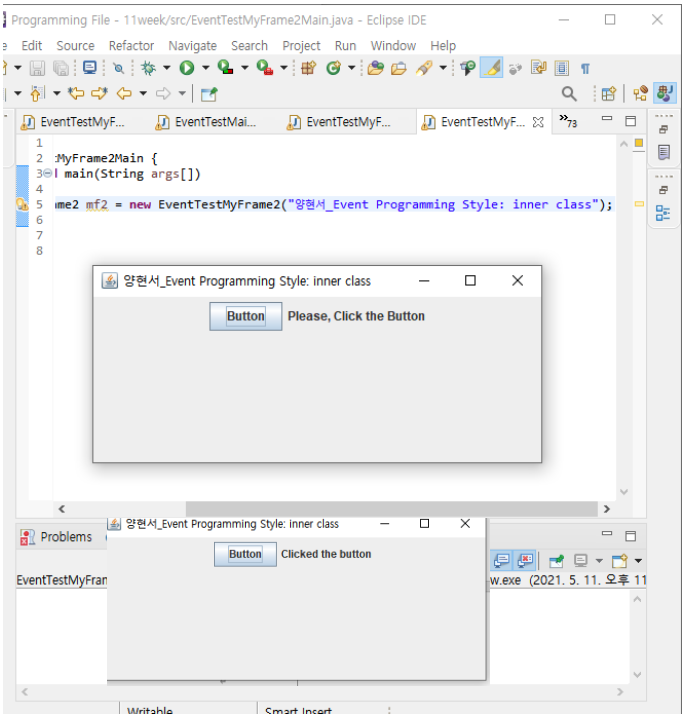
객체지향프로그래밍 11 주차 정리

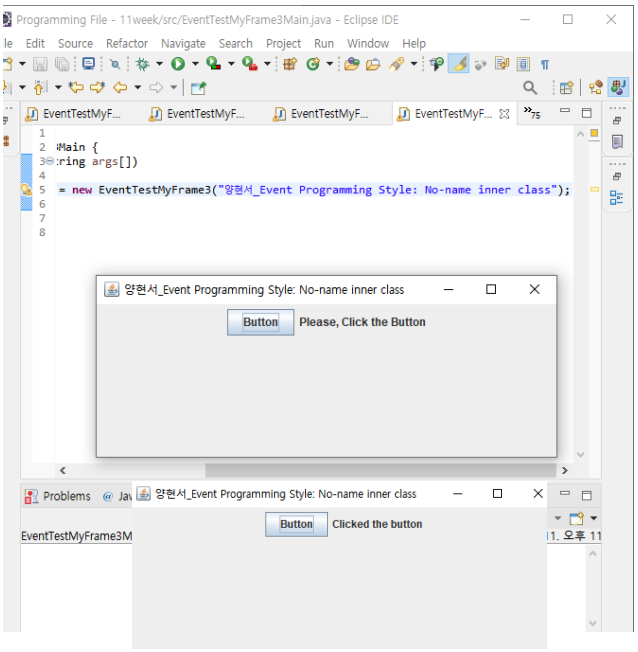
프로그래밍

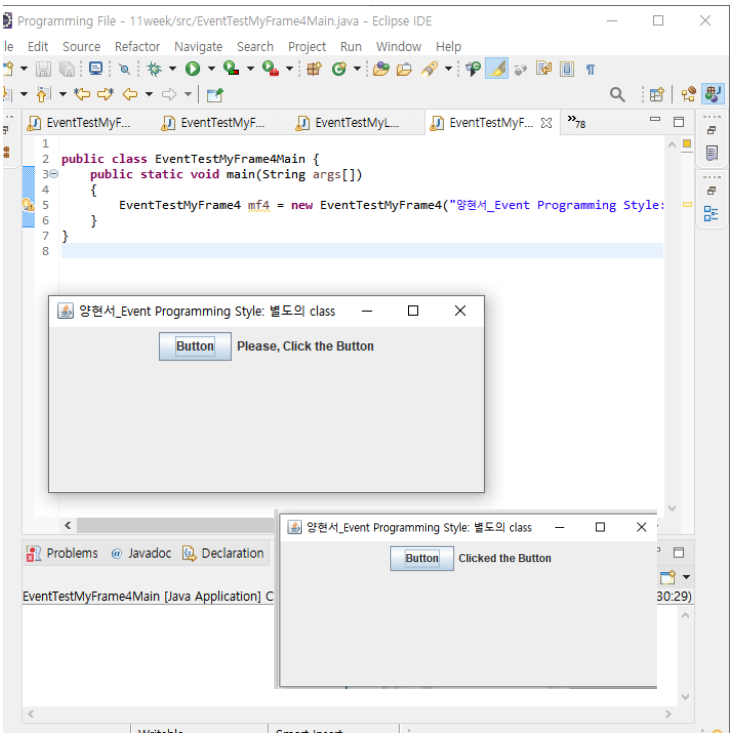


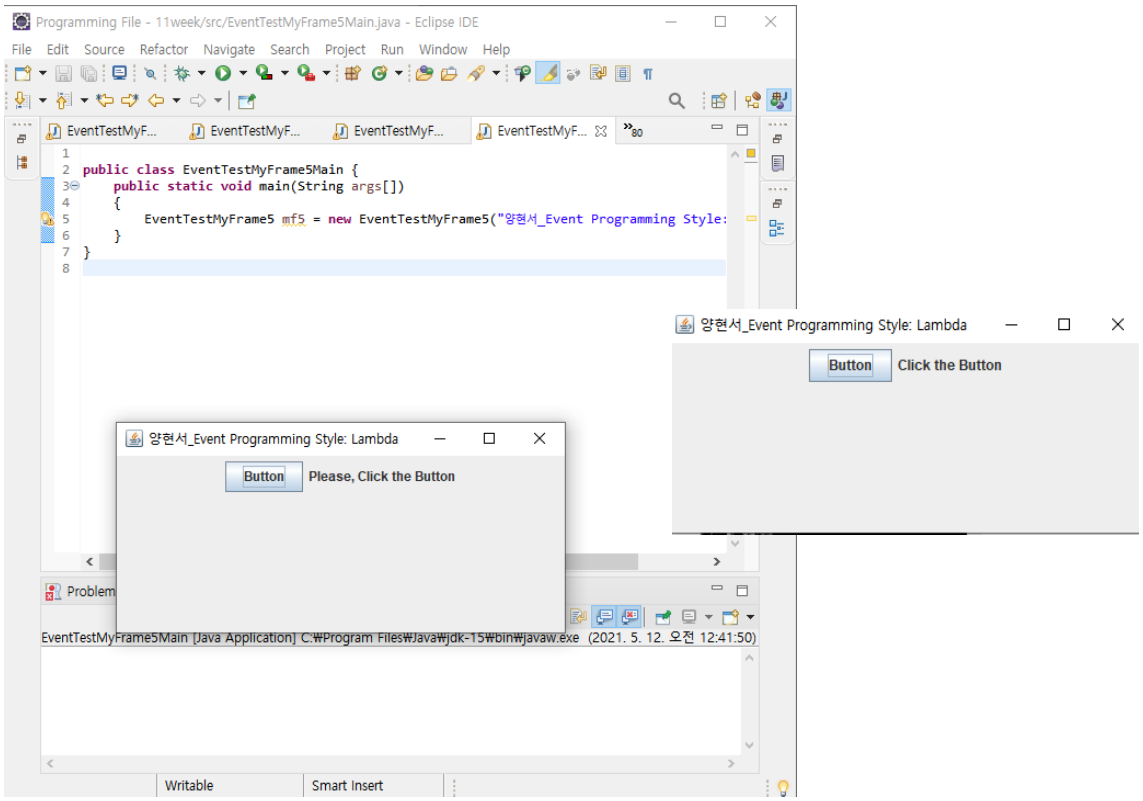


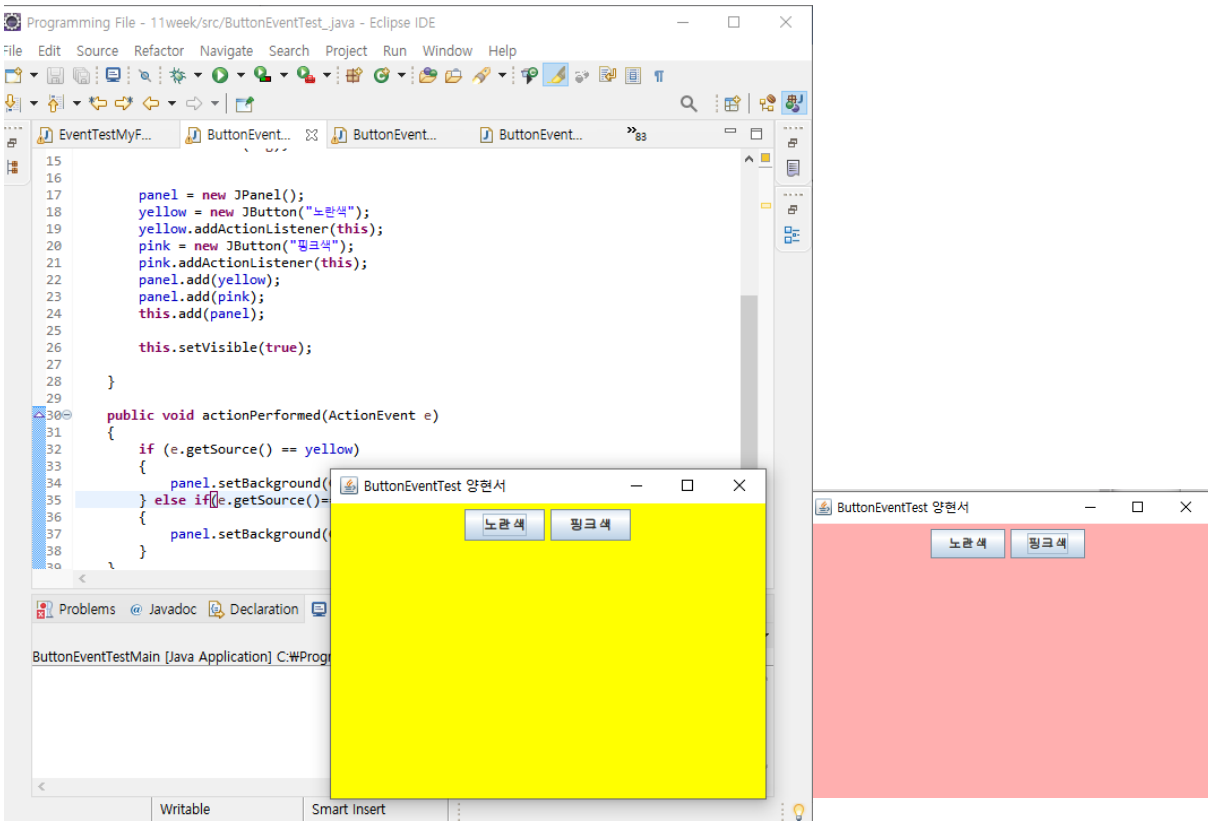


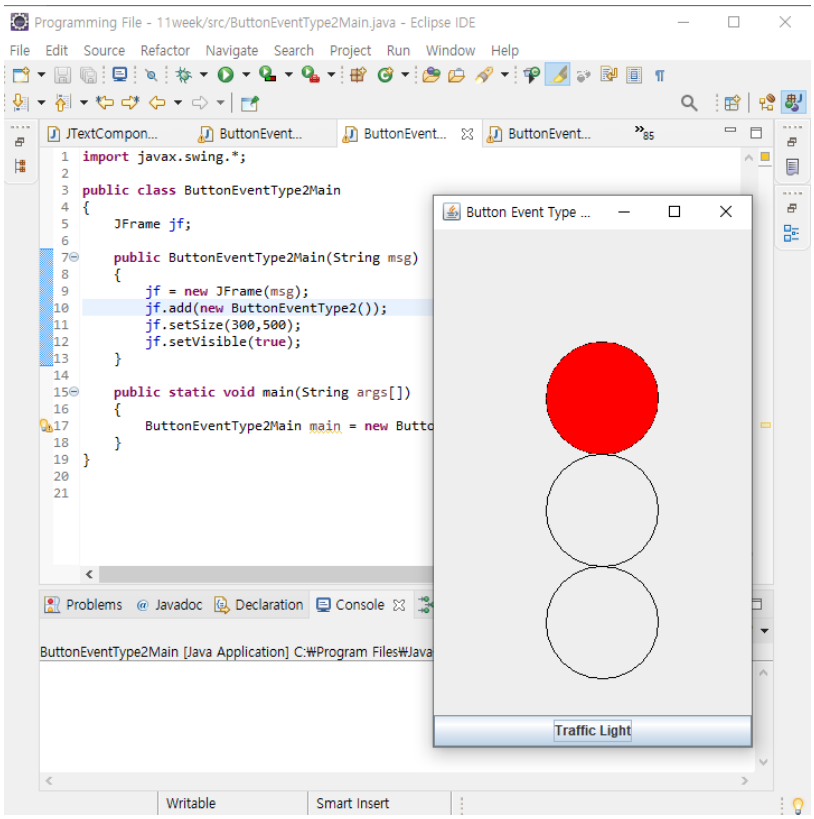

Programming File - 11week/src/EventTestMyFrame2Main.java - Eclipse IDE
양현서_Event Programming Style: inner class
Button
Please, Click the Button
양현서_Event Programming Style: inner class
Button
Clicked the button

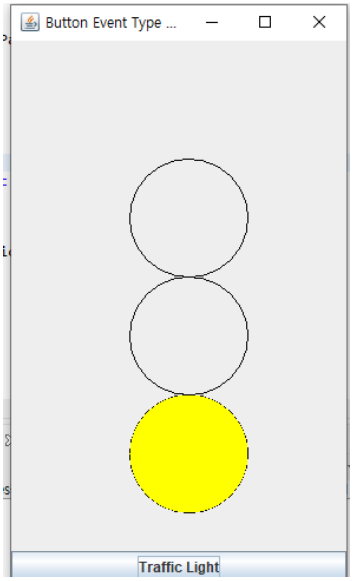

Programming File - 11week/src/EventTestMyFrame3Main.java - Eclipse IDE
양현서_Event Programming Style: No-name inner class
Button
Please, Click the Button
양현서_Event Programming Style: No-name inner class
Button
Clicked the button

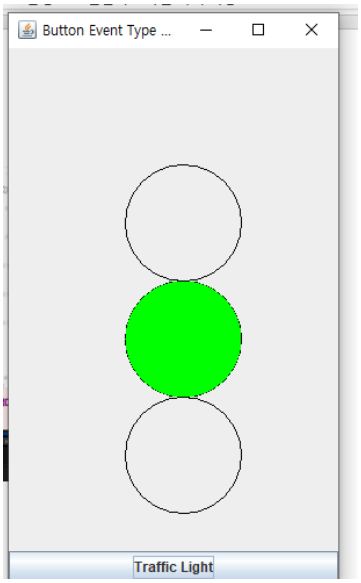

Programming File - 11week/src/EventTestMyFrame4Main.java - Eclipse IDE
양현서_Event Programming Style: 별도의 class
Button
Please, Click the Button
양현서_Event Programming Style: 별도의 class
Button
Clicked the Button

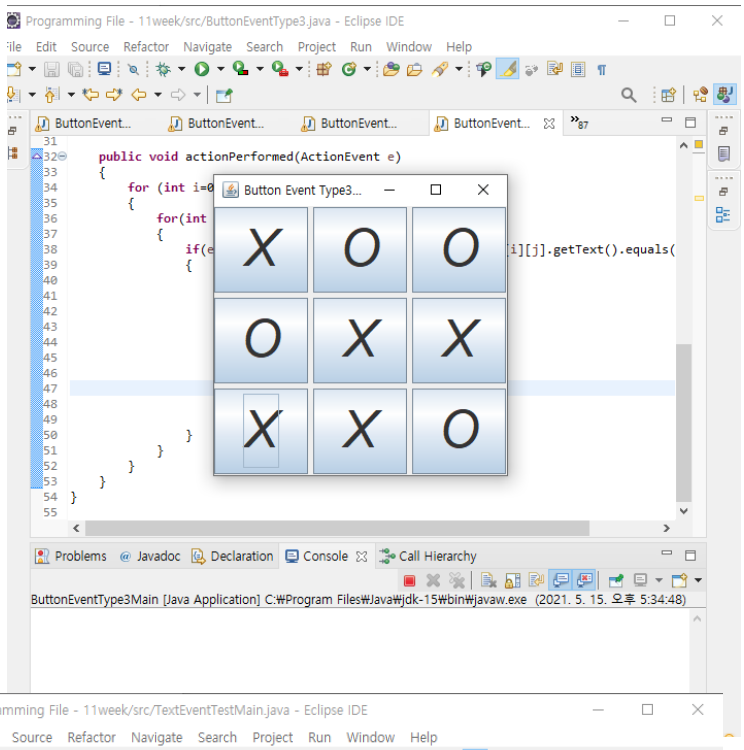

Programming File - 11week/src/EventTestMyFrame5Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
EventTestMyF...
EventTestMyF...
EventTestMyF...
EventTestMyF...
1
2 public class EventTestMyFrame5Main {
3 public static void main(String args[])
4 {
5 EventTestMyFrame5 mf5 = new EventTestMyFrame5("양현서_Event Programming Style:
6 }
7 }
8
양현서_Event Programming Style: Lambda
Button
Please, Click the Button
양현서_Event Programming Style: Lambda
Button
Click the Button
Problem
EventTestMyFrame5Main [Java Application] C:₩Program Files₩Java₩jdk-15₩bin₩javaw.exe (2021. 5. 12. 오전 12:41:50)
Writable
Smart Insert

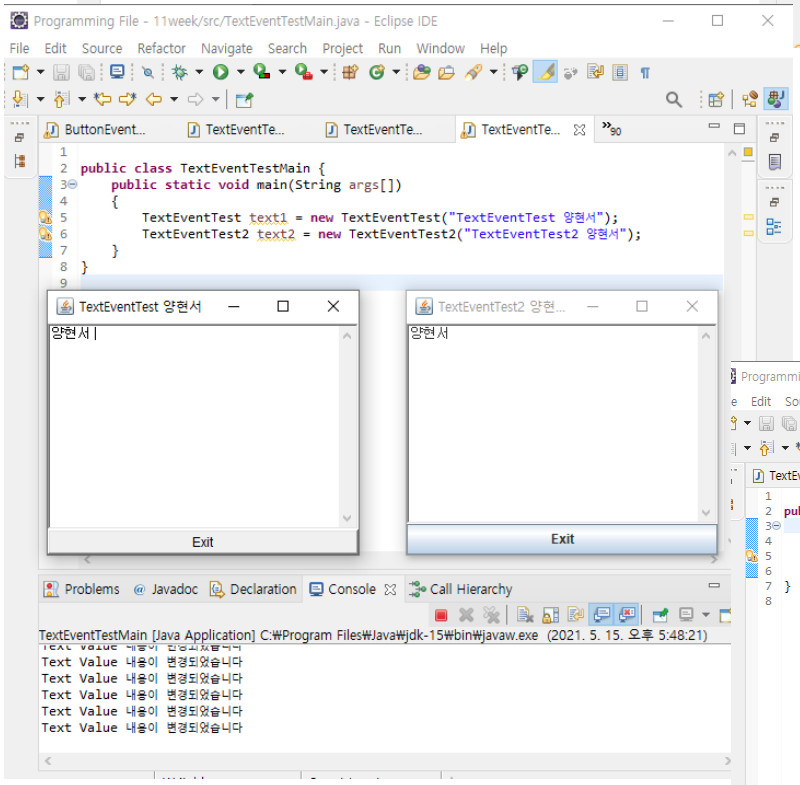

Programming File - 11week/src/ButtonEventTest_.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
EventTestMyF...
ButtonEvent...
ButtonEvent...
ButtonEvent...
15
16
17 panel = new JPanel();
18 yellow = new JButton("노란색");
19 yellow.addActionListener(this);
20 pink = new JButton("핑크색");
21 pink.addActionListener(this);
22 panel.add(yellow);
23 panel.add(pink);
24 this.add(panel);
25
26 this.setVisible(true);
27
28 }
29
30 public void actionPerformed(ActionEvent e)
31 {
32 if (e.getSource() == yellow)
33 {
34 panel.setBackground(
35 } else if(e.getSource()=
36 {
37 panel.setBackground(
38 }
ButtonEventTest 양현서
노란색
핑크색
ButtonEventTest 양현서
노란색
핑크색
Problems
Javadoc
Declaration
ButtonEventTestMain [Java Application] C:₩Progr
Writable
Smart Insert

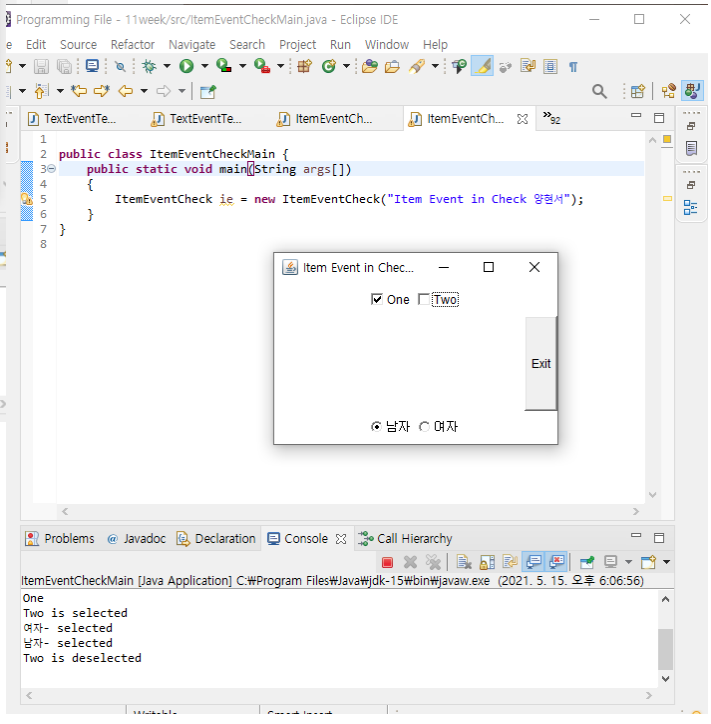

Programming File - 11week/src/ButtonEventType2Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
JTextCompon... ButtonEvent... ButtonEvent... ButtonEvent...
1 import javax.swing.*;
2
3 public class ButtonEventType2Main
4 {
5 JFrame jf;
6
7 public ButtonEventType2Main(String msg)
8 {
9 jf = new JFrame(msg);
10 jf.add(new ButtonEventType2());
11 jf.setSize(300,500);
12 jf.setVisible(true);
13 }
14
15 public static void main(String args[])
16 {
17 ButtonEventType2Main main = new Butto
18 }
19 }
20
21
Problems Javadoc Declaration Console
ButtonEventType2Main [Java Application] C:₩Program Files₩Java
Writable Smart Insert
Button Event Type ...
Traffic Light

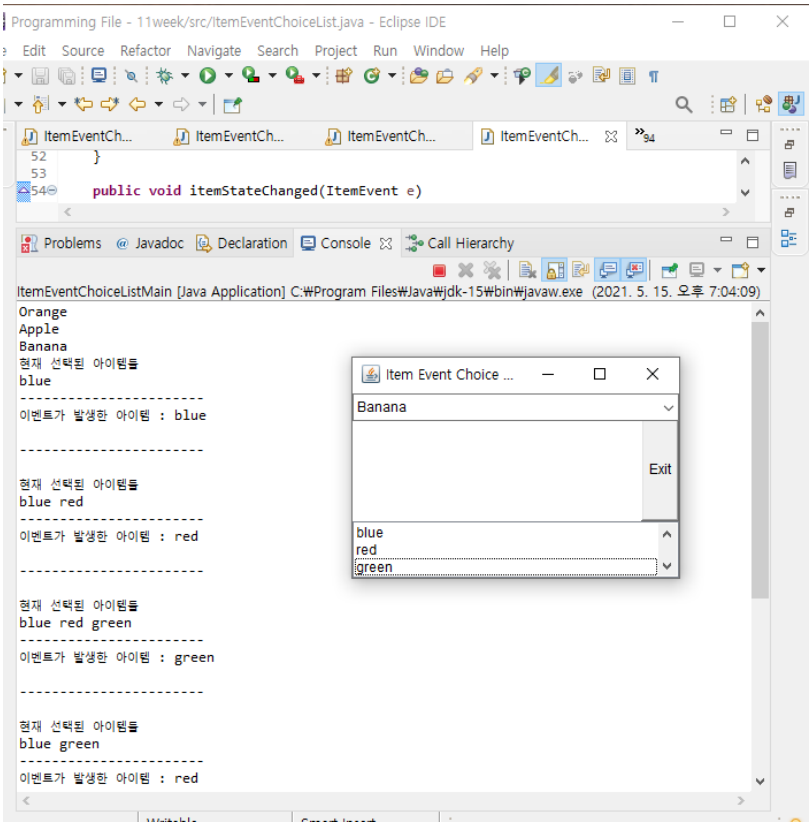

Button Event Type ...
Traffic Light


Button Event Type ...
Traffic Light


Programming File - 11week/src/ButtonEventType3.java - Eclipse IDE
Button Event Type3...
X O O
O X X
X X O
ButtonEventType3Main [Java Application] C:₩Program Files₩Java₩jdk-15₩bin₩javaw.exe (2021. 5. 15. 오후 5:34:48)


Programming File - 11week/src/TextEventTestMain.java - Eclipse IDE
public class TextEventTestMain {
public static void main(String args[])
{
TextEventTest text1 = new TextEventTest("TextEventTest 양현서");
TextEventTest2 text2 = new TextEventTest2("TextEventTest2 양현서");
}
}
TextEventTest 양현서
양현서
Exit
TextEventTest2 양현...
양현서
Exit
TextEventTestMain [Java Application] C:₩Program Files₩Java₩jdk-15₩bin₩javaw.exe (2021. 5. 15. 오후 5:48:21)
Text Value 내용이 변경되었습니다
Text Value 내용이 변경되었습니다
Text Value 내용이 변경되었습니다
Text Value 내용이 변경되었습니다
Text Value 내용이 변경되었습니다


Programming File - 11week/src/ItemEventCheckMain.java - Eclipse IDE
public class ItemEventCheckMain {
public static void main(String args[])
{
ItemEventCheck ie = new ItemEventCheck("Item Event in Check 양현서");
}
}
Item Event in Chec...
One Two
Exit
남자 여자
ItemEventCheckMain [Java Application] C:₩Program Files₩Java₩jdk-15₩bin₩javaw.exe (2021. 5. 15. 오후 6:06:56)
One
Two is selected
여자- selected
남자- selected
Two is deselected


Programming File - 11week/src/ItemEventChoiceList.java - Eclipse IDE
Edit Source Refactor Navigate Search Project Run Window Help
ItemEventCh...
ItemEventCh...
ItemEventCh...
ItemEventCh...
52 }
53
54 public void itemStateChanged(ItemEvent e)
Problems Javadoc Declaration Console Call Hierarchy
ItemEventChoiceListMain [Java Application] C:₩Program Files₩Java₩jdk-15₩bin₩javaw.exe (2021. 5. 15. 오후 7:04:09)
Orange
Apple
Banana
현재 선택된 아이템들
blue

이벤트가 발생한 아이템 : blue

현재 선택된 아이템들
blue red

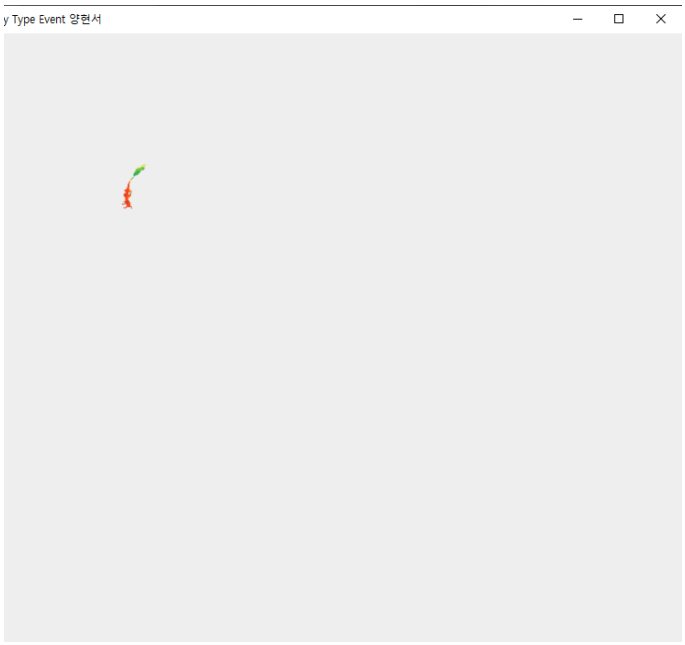
이벤트가 발생한 아이템 : red

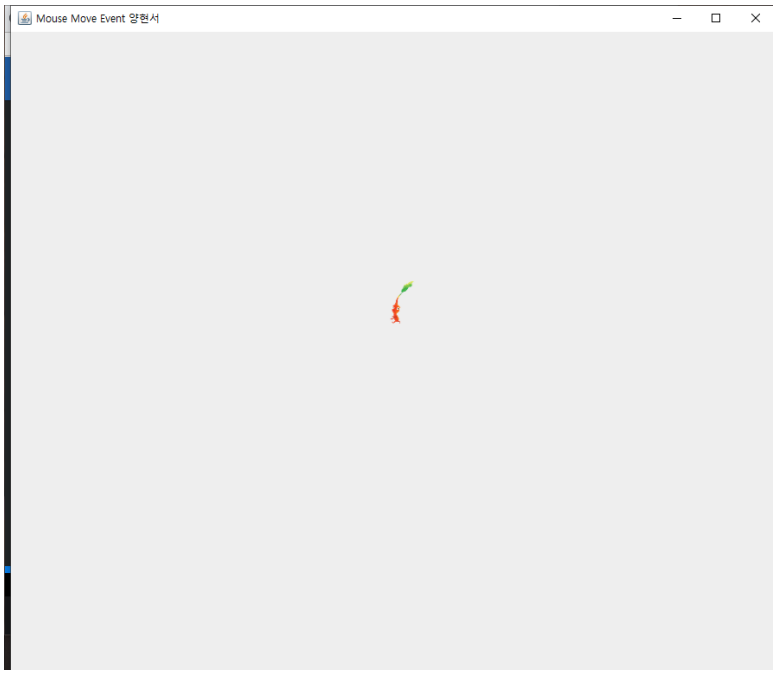
현재 선택된 아이템들
blue red green

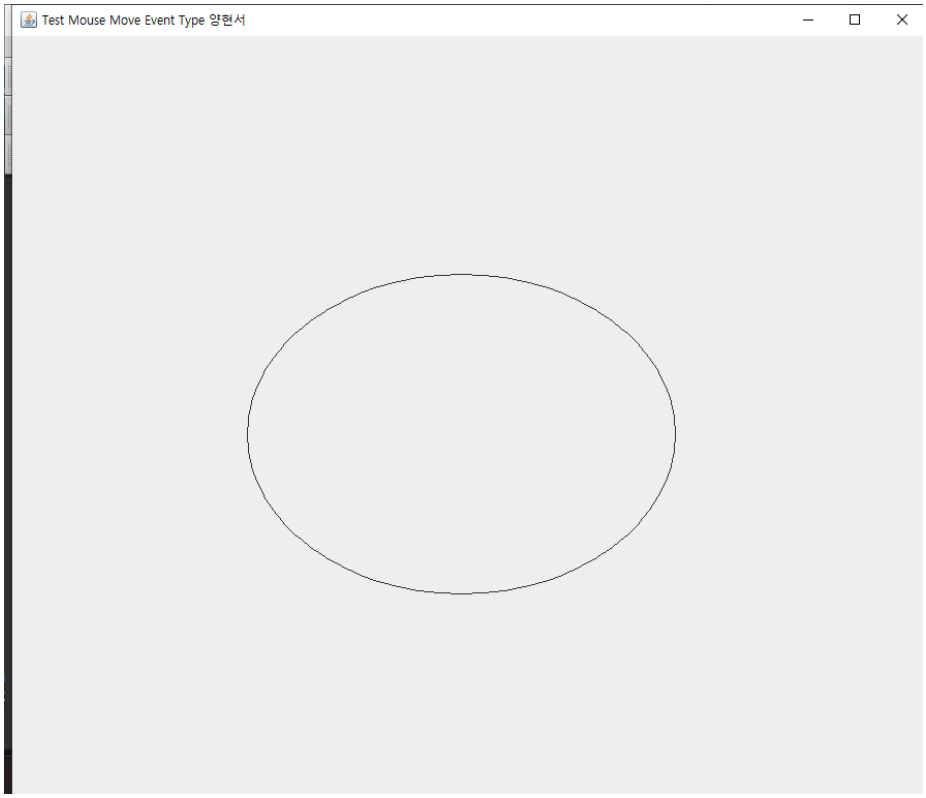
이벤트가 발생한 아이템 : green

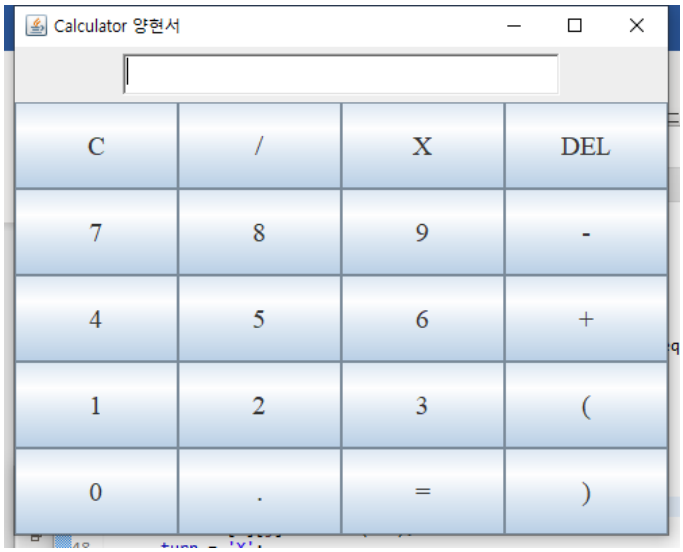
현재 선택된 아이템들
blue green

이벤트가 발생한 아이템 : red
Item Event Choice ...
Banana
Exit
blue
red
green

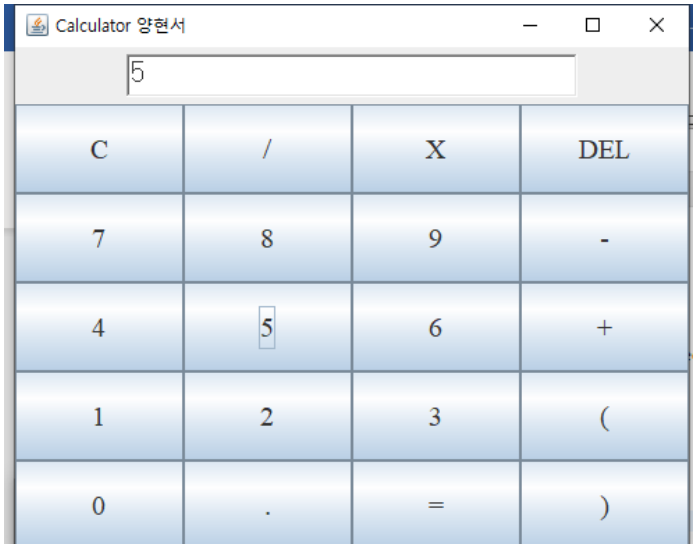

y Type Event 양현서


Mouse Move Event 양현서


Test Mouse Move Event Type 양현서



초기화면



계산 중간

계산 결과

설명

: 입력한 내용들을 하나의 String 으로 만들어서 String 내용으로 계산을 하고 싶었으나 String 이 하나로 묶이지 않아 계산 결과가 나오지 않습니다.

String 이 묶이는 것을 확인한 후에 계산 코드를 작성하려 했으나 String 이 묶이지 않아 계산 코드도 작성하지 못했습니다.

버튼을 누르면 텍스트 필드에 값이 작성되기는 합니다.

이벤트

GUI 에서 action 을 하고 싶을 때

Ex)쓰레드 > Timer Interrupt

클릭 -> 이벤트

마우스 -> 이벤트

정의한 기능이 수행된다

이벤트 드리븐 핸들링(Event Driven Handling)

⇨ ECA rule

클릭 -> 이벤트 발생, 시그니처, 메시지 전달 -> 컨디션 체크, 메시지 내용 분석, 실행할 코드 선택 -> 함수 수행, 액션

컨디션 체크 부분이 제일 어렵다

⇨ EventListener

컴포넌트에서 이벤트 발생 -> 이벤트 리스너 -> 수행(implementation: 개발자가 구현할 부분)

Design 단계 : GUI 설계

지금까지 한 것

리스너 등록

이벤트 처리를 쉽게 하기 위해 만들어졌다

앞으로 할 것

리스너는 자신이 수행할 함수를 가지고 있다 => 빠르게 동작할 수 있다

이벤트 핸들링을 위한 method

actionPerformed

이벤트를 지원하기 위한 메소드

getSource()　　: 이벤트의 소스 Object

getId()　　: 이벤트 타입 Type

getActionCommand()　　: 컴포넌트 이름 Component name

자바에서의 이벤트 프로세싱 스텝

1. 버튼 오브젝트 생성(설계)

   *클래스에 ActionListener 를 implements 해주어야 한다

   ActionListenr 는 interface

   이 안에 actionPerformed 가 정의되어 있다

   따라서 항상 ActionListenr 를 상속받아야 한다

2. 리스너 등록(객체.addActionListener())
3. 구현(actionPerformed()함수를 만든다)

⇨ 언어마다 함수 이름이 다르다

AWT Event

ActionEvent

AdjustmentEvent

ComponentEvent

ContainerEvent

FocusEvent

ItemEvent

KeyEvent

MouseEvent

MouseMotionEvent

TextEvent

WindowEvent

이벤트 핸들링 코드

어디에 어떻게 구현?

1. Listener 를 상속받아서 구현(가장 기본적인 방법)
2. Named Inner class(많이 사용할 때)

   여러 번 반복해서 사용할 땐 2 번이 가장 최적의 방법

   이너클래스(Inner Class)

   java 파일 컴파일 시 두개의 클래스파일 생성

   원래클래스이름$이너클래스이름.class
3. No name Inner class(가장 많이 사용하는 방법, 한번만 사용)

   딱 한번만 사용할 때에는 3 번으로 하는 것이 최적의 방법
4. Independent class(좋지 않은 방법, 독립된 클래스를 만들어서 사용하는 방법, 가장 안 쓰는 방법)

   다른 사람과 공유하고 싶을 때 사용한다
5. Lambda Expression(3 번과 같은 형태, 더 쉽게 만든 것)

   간단한 것은 5 번으로 하는 것이 가장 최적

   코드의 블록을 표현할 수 있다

   코드 내용을 알고 있을 때 Lambda 를 이용해서 코드 블록을 만드는 방법

   (argument) -> {body}

   *패턴을 알고 있어야 한다

   3 번을 간단히 한 것이 Lambda Expression

⇨ 정답은 없다. 프로그래밍 스타일에 따라 다를 것

**오버로딩: 똑같은 이름을 여러 개 만드는 것(ex) 객체 생성 시 받는 매개변수마다 실행될 생성자 들)

**오버라이딩: 재구현하는 것(ex) 인터페이스의 함수들의 구현 내용을 다시 구현하는 것)

이벤트 타입에 따른 Implements

Button Event ->Action Event: 사용자가 버튼을 클릭하는 경우, 메뉴를 클릭하는 경우

Mouse Event: 마우스를 사용하는 경우

Key Event : 특정 Key 를 사용하는 경우

Text Event: 텍스트 필드에서 엔터키를 누르는 경우

⇨ 등록을 하고 구현

Button Event Types

addActionListener()등록, actionPerformed()를 수행

actionPerformed 를 등록시켜주면 된다

Text Event

: 텍스트를 칠 때마다 텍스트 내용이 변경되었다고 뜨는 것

Item Event

아이템이 선택되거나 해제될 때마다 이벤트가 발생

CheckBox, Choice, RadioButton, List 등에게 아이템 이벤트를 사용할 수 있다

itemStateChange(itemEvent e)

SELECTED, DESELECTED, ITEM_STATE_CHANGED 의 필드값이 존재

getItem() : 선택된 아이템 객체 반환

getItemSelectable(), getStateChange() 함수 사용 가능

Key Event

keyTyped(KeyEvent e)

keyPressed(KeyEvent e)

KeyReleased(KeyEvent e)

Mouse Event