객체지향프로그래밍응용 과제06 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

[예제 9-3]

[소스코드]

// 예제 9-3

// FlowLayout 배치관리자 활용

// 수평 간격이 30, 수직 간격을 40픽셀로 하고 LEFT로 정렬 배치하는 FlowLayout 배치관리자를 가진 컨텐트팬에

// 5개의 버튼 컴포넌트를 부착한 스윙 응용프로그램을 작성하라.

import javax.swing.\*;

import java.awt.\*;

public class FlowLayoutEx extends JFrame {

public FlowLayoutEx() {

setTitle("FlowLayout Sample");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

// 컨텐트팬에 FolwLayout 배치관리자 설정

c.setLayout(new FlowLayout(FlowLayout.LEFT, 30, 40));

c.add(new JButton("add"));

c.add(new JButton("sub"));

c.add(new JButton("mul"));

c.add(new JButton("div"));

c.add(new JButton("Calculate"));

setSize(300, 200); // 프레임 크기 300x200 설정

setVisible(true); // 화면에 프레임 출력

}

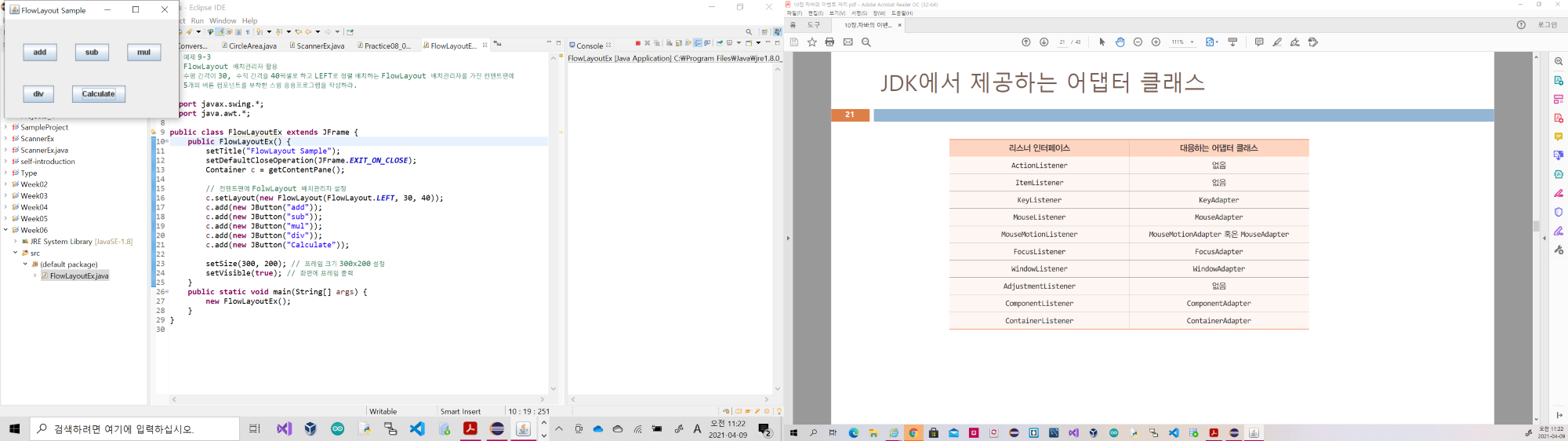
public static void main(String[] args) {

new FlowLayoutEx();

}

}

[실행결과]



[예제9-4]

[소스코드]

// 예제 9-4

// BorderLayout 배치관리자를 사용하는 예

// BorderLayout 배치관리자를 사용하여 컨텐트팬에 다음과 같이 5개의 버튼 컴포넌트를 삽입하라.

import javax.swing.\*;

import java.awt.\*;

public class BorderLayoutEx extends JFrame {

public BorderLayoutEx() {

setTitle("BorderLayout Sample");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

// 컨텐트팬에 BorderLayout 배치관리자 설정

c.setLayout(new BorderLayout(30, 20));

c.add(new JButton("Calculate"), BorderLayout.CENTER);

c.add(new JButton("add"), BorderLayout.NORTH);

c.add(new JButton("sub"), BorderLayout.SOUTH);

c.add(new JButton("mul"), BorderLayout.EAST);

c.add(new JButton("div"), BorderLayout.WEST);

setSize(300, 200); // 프레임 크기 300x200 설정

setVisible(true); // 프레임 화면에 출력

}

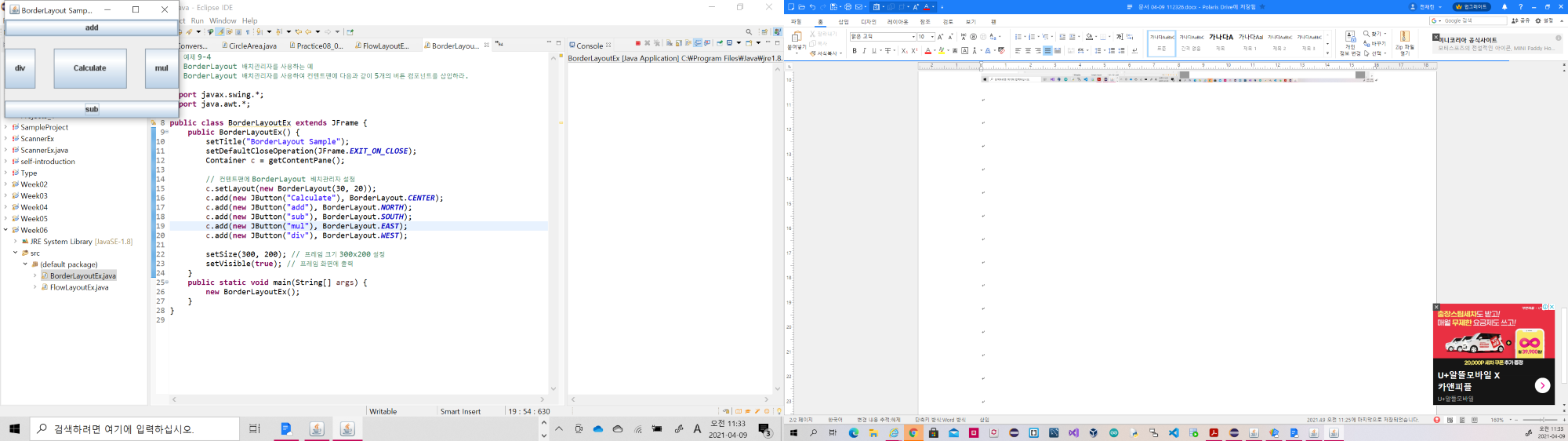
public static void main(String[] args) {

new BorderLayoutEx();

}

}

[실행결과]



[예제 9-5]

[소스코드]

// 예제 9-5

// GridLayout으로 입력 폼 만들기

// 아래 화면과 같이 사용자로부터 입력받는 폼인 스윙 응용프로그램을 작성하라.

//GridLayout을 사용하여 콘텐트팬을 4x2 그리드로 분할하고 JLable 컴포넌트와 JTextField 컴포넌트를 부착하면 된다.

// 두 행 사이의 수직 간격은 5픽셀로 하라.

import javax.swing.\*;

import java.awt.\*;

public class GridLayoutEx extends JFrame {

public GridLayoutEx() {

setTitle("GridLayout Sample");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// 4x2 격자의 GridLayout 배치관리자 생성

GridLayout grid = new GridLayout(4, 2);

grid.setVgap(5); // 격자 사이의 수직 간격을 5 픽셀로 설정

Container c = getContentPane();

c.setLayout(grid); // grid를 컨텐트팬의 배치관리자로 지정

c.add(new JLabel(" 이름"));

c.add(new JTextField(""));

c.add(new JLabel(" 학번"));

c.add(new JTextField(""));

c.add(new JLabel(" 학과"));

c.add(new JTextField(""));

c.add(new JLabel(" 과목"));

c.add(new JTextField(""));

setSize(300, 200);

setVisible(true);

}

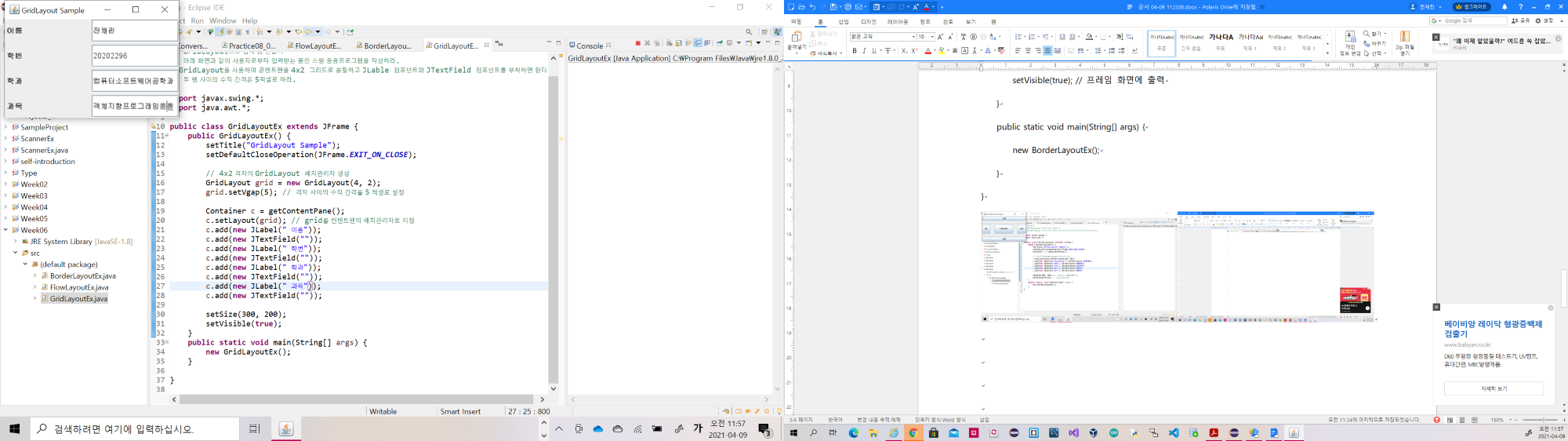
public static void main(String[] args) {

new GridLayoutEx();

}

}

[실행결과]



[예제 9-6]

[소스코드]

// 예제 9-6

// 배치관리자 없는 컨테이너에 컴포넌트를 절대 위치와 절대 크기로 지정

// 다음 그림과 같이 컨텐트팬에 배치관리자를 삭제하고 9개의 버튼과 하나의 문자열을 출력하는 프로그램을 작성하라.

// "Hello, Press Buttons!" 문자열은 JLabel 컴포넌트를 이용하고, 그 위치를 (130, 500)에, 크기는 200x20으로 한다.

// 버튼의 크기는 50x20으로 한다.

import javax.swing.\*;

import java.awt.\*;

public class NullContainerEx extends JFrame {

public NullContainerEx() {

setTitle("NullContainer Sample"); // 프레임 타이틀 달기

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); // 프레임 윈도우를 닫으면 프로그램 종료

Container c = getContentPane();

c.setLayout(null);

// JLabel 컴포넌트를 생성하고 위치와 크기를 직접 지정한다.

JLabel la = new JLabel("Hello, Prsess Button!");

la.setLocation(130, 50); // la를 (130, 50) 위치로 지정

la.setSize(200, 20); // la를 200x20 크기로 지정

c.add(la); // la를 컨텐트팬에 부착

// 9개의 버튼 컴포넌트를 생성하고 모두 동일한 크기로 설정한다.

// 위치는 서로 다르게 설정

for(int i=1; i<=9; i++) {

JButton b = new JButton(Integer.toString(i)); // 버튼 생성

b.setLocation(i\*15, i\*15); // 버튼의 위치 설정

b.setSize(50, 20); // 9 개의 버튼 크기는 동일하게 50x20

c.add(b); // 버튼을 컨텐트팬에 부착

}

setSize(300, 200);

setVisible(true);

}

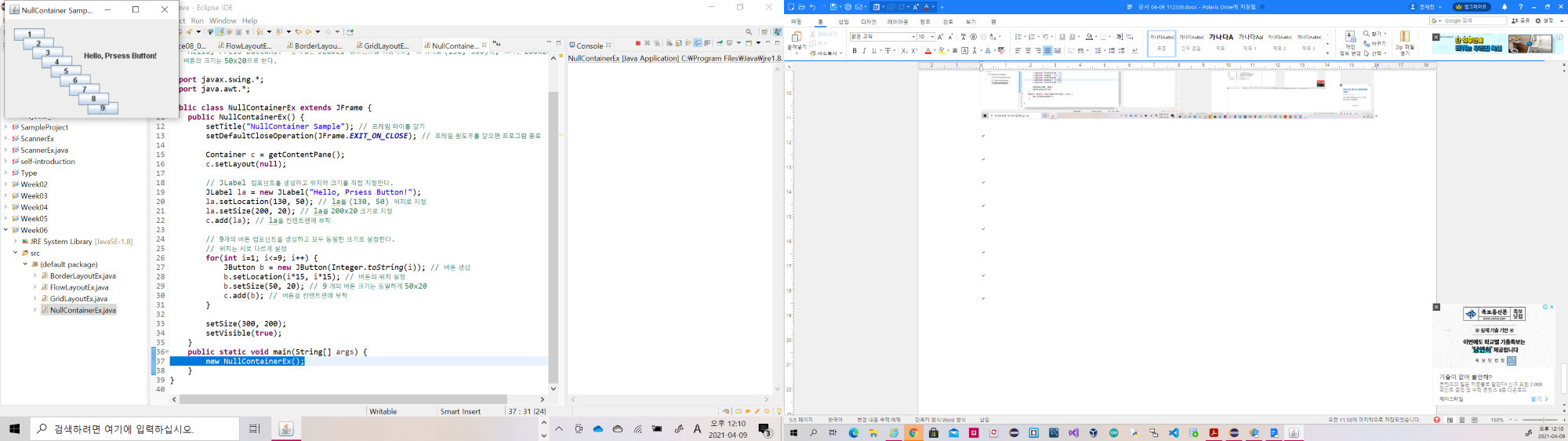
public static void main(String[] args) {

new NullContainerEx();

}

}

[실행결과]



[예제 10-1]

[소스코드]

// 예제 10-1

// 독립된 클래스로 Action 이벤트 리스너 작성

// 버튼을 클릭할 때 발생하는 Action 이벤트를 처리할 MyActionListener 클래스를 독립 클래스로 작성하라.

// 클릭할 때마다 버튼 문자열이 "Action"과 "액션"으로 번갈아 변하도록 하라.

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class IndepClassListener extends JFrame {

public IndepClassListener() {

setTitle("Action 이벤트 리스너 예제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.setLayout(new FlowLayout());

JButton btn = new JButton("Action");

btn.addActionListener(new MyActionListener()); // Action 이벤트 리스너 달기

c.add(btn);

setSize(350, 150);

setVisible(true);

}

public static void main(String[] args) {

new IndepClassListener();

}

}

// 독립된 클래스로 이벤트 리스너를 작성한다.

class MyActionListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

JButton b = (JButton)e.getSource(); // 이벤트 소스 버튼 알아내기

if(b.getText().equals("Action")) // 버튼의 문자열이 "Action"인지 비교

b.setText("액션"); // 버튼의 문자열이 "액션"으로 변경

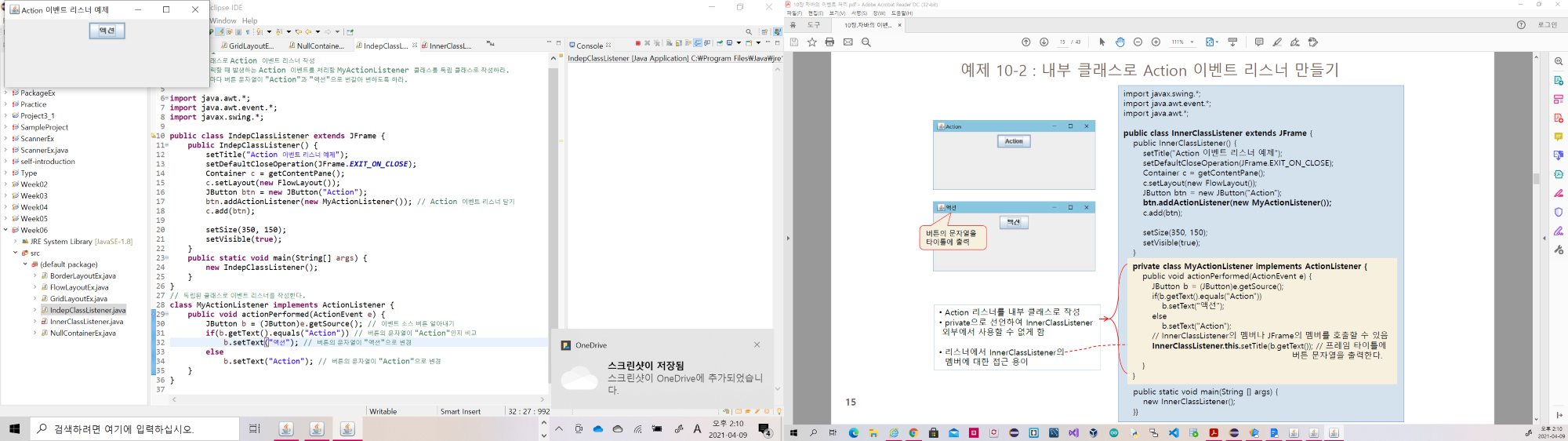
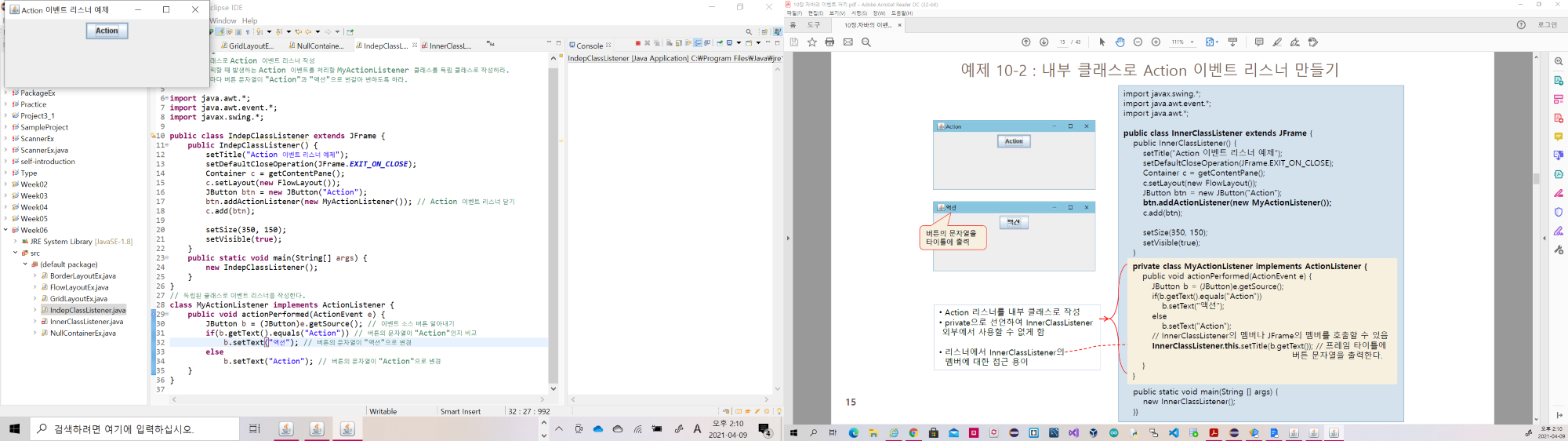
else

b.setText("Action"); // 버튼의 문자열이 "Action"으로 변경

}

}

[실행결과]



[예제10-2]

[소스코드]

// 예제 10-2

// 내부 클래스로 Action 이벤트 리스너 만들기

// 버튼을 클릭할 때 발생하는 Action 이벤트를 처리할 MyActionListener 클래스를 내부 클래스로 작성하라.

// 클릭할 때마다 버튼 문자열이 "Action"과 "액션"으로 번갈아 변하도록 하라.

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class InnerClassListener extends JFrame {

public InnerClassListener() {

setTitle("Action 이벤트 리스너 예제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.setLayout(new FlowLayout());

JButton btn = new JButton("Action");

btn.addActionListener(new MyActionListener()); // Action 이벤트 리스너 달기

c.add(btn);

setSize(350, 150);

setVisible(true);

}

// 내부 클래스로 Action 리스너를 작성한다.

class MyActionListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

JButton b = (JButton)e.getSource(); // 이벤트 소스 버튼 알아내기

if(b.getText().equals("Action")) // 버튼의 문자열이 "Action"인지 비교

b.setText("액션"); // 버튼의 문자열이 "액션"으로 변경

else

b.setText("Action"); // 버튼의 문자열이 "Action"으로 변경

// InnerClassListener의 멤버나 JFrame의 멤버를 호출할 수 있음

InnerClassListener.this.setTitle(b.getText()); // 프레임의 타이틀레 버튼 문자열을 출력한다.

}

}

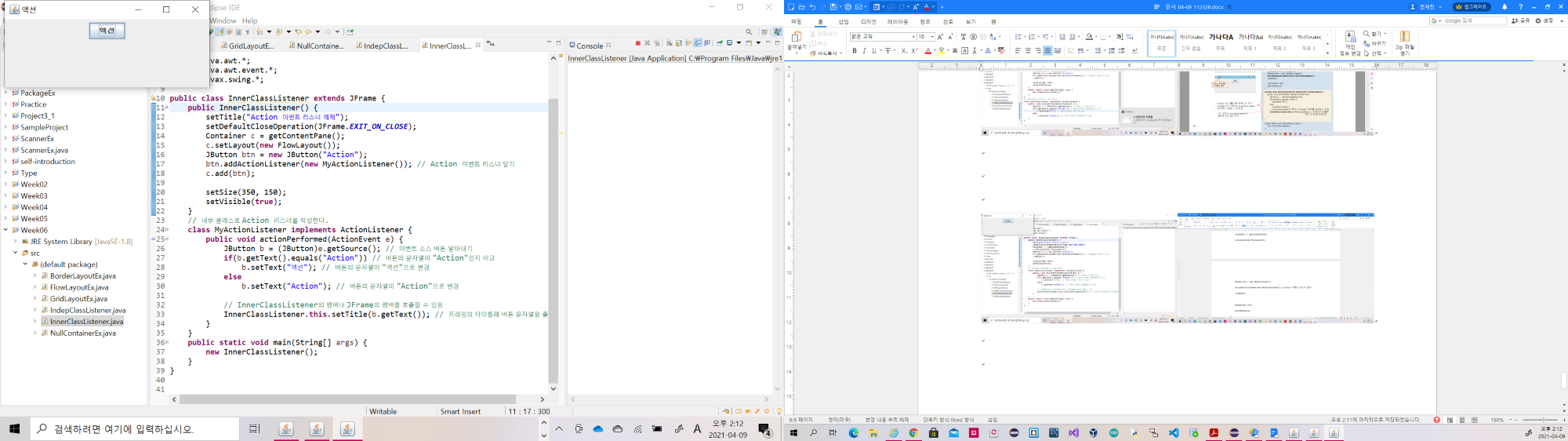
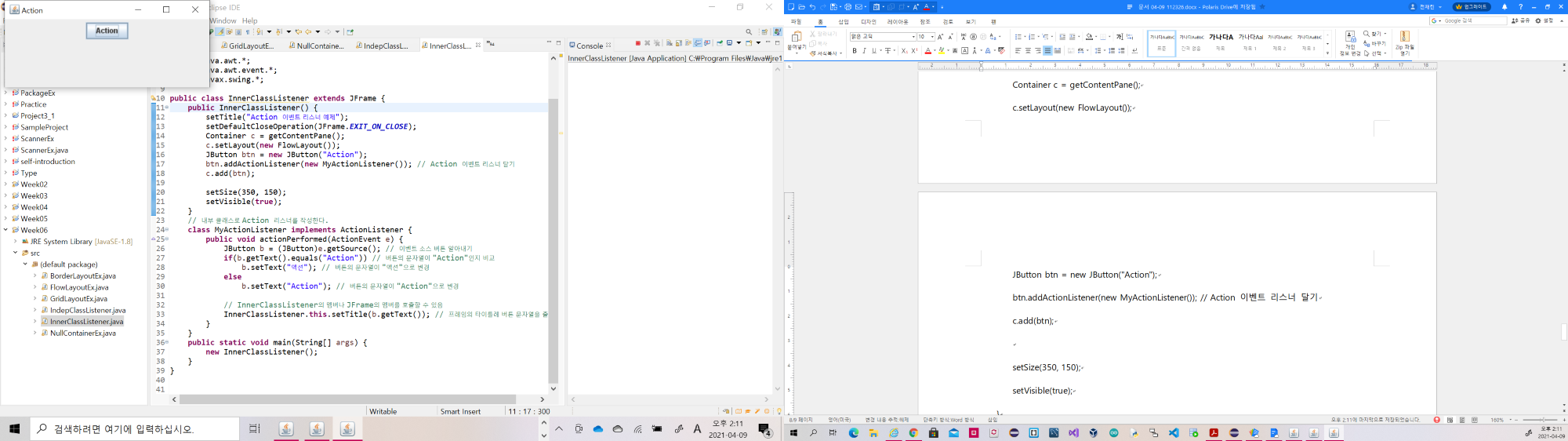
public static void main(String[] args) {

new InnerClassListener();

}

}

[실행결과]



[예제 10-3]

[소스코드]

// 예제 10-3

// 익명 클래스로 Action 이벤트 리스너 만들기

// 버튼을 클릭할 때 발생하는 Action 이벤트를 처리할 MyActionListener 클래스를 익명 클래스로 작성하라.

// 클릭할 때마다 버튼 문자열이 "Action"과 "액션"으로 번갈아 변하도록 하라.

// 예제 10-2를 익명 클래스를 이용하여 재작성하라.

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class AnonymousClassListener extends JFrame {

public AnonymousClassListener() {

setTitle("Action 이벤트 리스너 예제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.setLayout(new FlowLayout());

JButton btn = new JButton("Action");

c.add(btn);

btn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

JButton b = (JButton)e.getSource(); // 이벤트 소스 버튼 알아내기

if(b.getText().equals("Action")) // 버튼의 문자열이 "Action"인지 비교

b.setText("액션"); // 버튼의 문자열이 "액션"으로 변경

else

b.setText("Action"); // 버튼의 문자열이 "Action"으로 변경

// AnonymousClassListener의 멤버나 JFrame의 멤버를 호출할 수 있음

setTitle(b.getText()); // 프레임의 타이틀레 버튼 문자열을 출력한다.

}

});

setSize(350, 150);

setVisible(true);

}

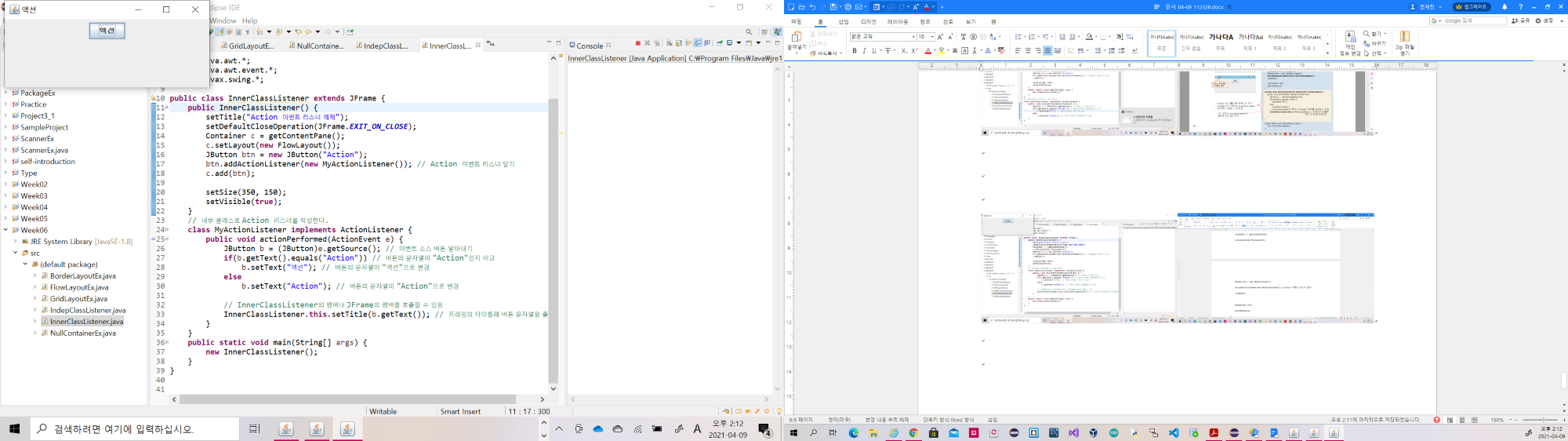
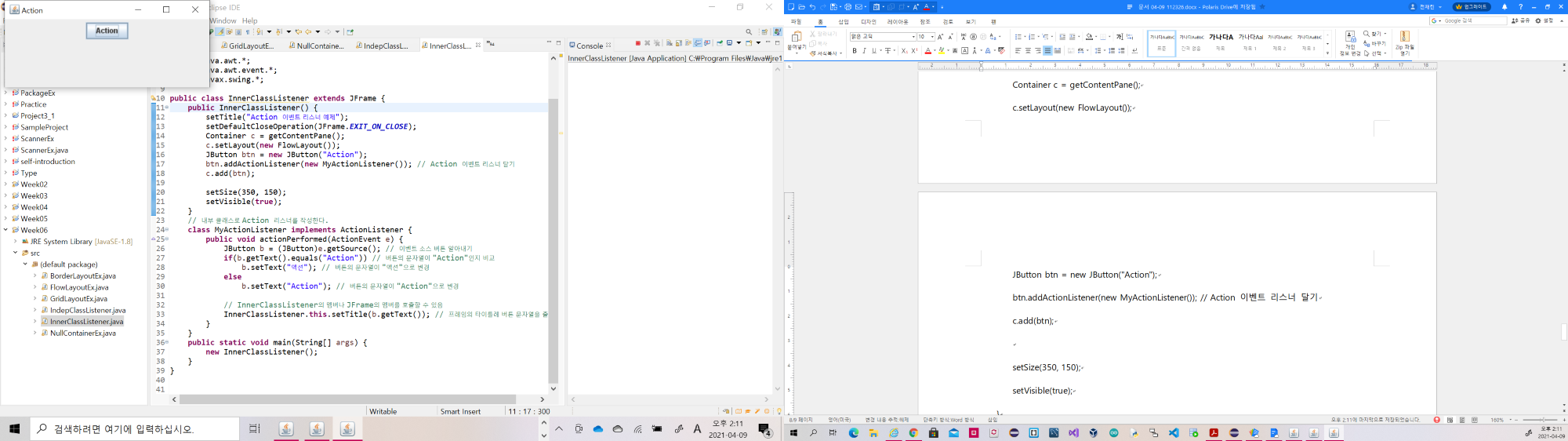
public static void main(String[] args) {

new AnonymousClassListener();

}

}

[실행결과]



[09장 연습문제(이론문제)]

1. 자바에서는 AWT 컴포넌트와 스윙 컴포넌트를 제공한다. 이들 중 어떤 것이 경량 컴포넌트(light weight component)이고, 어떤 것이 중량 컴포넌트(heavy weight component)인가? 그리거 이들은 어떤 점에서 서로 다른가?

AWT는 중량 컴포넌트이고, 스윙은 경량 컴포넌트이다. 중량 컴포넌트는 운영체제의 도움을 받아 화면에 출력되기 때문에 운영체제의 자원을 많이 소모하여 운영체제에 많은 부담을 준다. 경량 컴포넌트는 AWT와 달리 순수 자바 언어로 작성되었다. 운영체제의 도움을 받지 않기에 운영체제와 관계없이 항상 동일하게 작동하며 동일한 모양으로 그려진다.

3. 다음 중 스윙 컴포넌트가 아닌 것은?

JPanel

JTextField

JLabel

Button

-> , 스윙 컴포넌트의 이름은 AWT 컴포넌트와 구분하기 위해 모두 대문자 J로 시작한다.

5. 컴포넌트와 컨테이너에 대해 잘못 말한 것은?

컨테이너는 컴포넌트를 담을 수 있는 컴포넌트이다.

JPanel은 컨테이너로서 여러 개의 JButton 컴포넌트를 가질 수 있다.

JFrame은 최상위 컨테이너이다.

컴포넌트들은 컨테이너 없이도 출력된다.

-> , 컴포넌트는 컨테이너에 포함되어야 비로소 화면에 출력될 수 있는 GUI 객체이다.

7. 다음은 200x300 크기의 스윙 프레임을 만든 코드이다. 빈칸에 필요한 코드를 채워라.

import javax.swing.\*; // 이곳에 필요한 import 문을 삽입하라.

public class MyFrame extends JFrame {

public MyFrame() {

setTitle("hello"); // 프레임 타이틀로 "hello" 문자열 출력

setSize(200, 300); // 프레임 크기를 200x300으로 설정

setVisible(true); // 프레임 화면 출력

}

public static void main(String[] args) {

MyFrame frame = new MyFrame();

}

}

8. 다음은 스윙 프레임을 작성한 코드이다. 빈칸을 채워라

import javax.swing.\*; // 이곳에 필요한 import 문을 삽입하라.

import java.awt.\*; // 이곳에 필요한 import 문을 삽입하라.

public class MyFrame extends JFrame {

public MyFrame() {

Container c = getContentPane(); // 컨텐트팬 알아내기

c.setLayout(new FlowLayout()); // 컨텐트팬에 FlowLayout 배치관리자 설정

c.setBackground(Color.YELLOW); // 컨텐트팬 배경색을 노란색으로 설정

c.add(new JButton("click")); // 컨텐트팬에 "click" 버튼 달기

setSize(300, 300); // 프레임 크기를 200x300으로 설정

setVisible(true); // 프레임 화면 출력

}

}

9. 다음 지시에 따라 컨테이너 c에 배치관리자를 설정하는 코드를 작성하라.

Container c;

c.

(1) 컴포넌트 사이의 수평 수직 간격이 각각 3, 4픽셀인 BorderLayout

c.setLayout(new BorderLayout(3, 4));

(2) 컴포넌트 사이의 수평 수직 간격이 각각 5, 6픽셀이고 오른쪽 정렬하는 FlowLayout

c.setLayout(new FlowLayout(FlowLayout.RIGHT, 5, 6));

(3) 컴포넌트 사이의 수평 수직 간격이 각각 7, 8픽셀이고 행수 5, 열수 2인 GridLayout

new GridLayout(5, 2, 7, 8);

c.setLayout(grid);

(4) 현재 등록된 배치관리자 삭제

c.setLayout(null);

[09장 연습문제(실습문제)]

2. BorderLayout을 사용하여 컴포넌트 사이의 수평 수직 간격이 각각 5픽셀, 7픽셀이 되도록 스윙 응용프로그램을 작성하라.

[소스코드]

// Practice09\_02

// BorderLayout을 사용하여 컴포넌트 사이의 수평 수직 간격이 각각 5픽셀, 7픽셀이 되도록 스윙 응용프로그램을 작성하라.

// 컨텐트팬과 BorderLayout 활용

import javax.swing.\*;

import java.awt.\*;

public class Practice09\_02 extends JFrame {

public Practice09\_02() {

setTitle("Practice09\_02");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.setLayout(new BorderLayout(5, 7));

// 컨텐트팬에 BorderLayout 배치관리자 설정

c.add(new JButton("Center"), BorderLayout.CENTER);

c.add(new JButton("North"), BorderLayout.NORTH);

c.add(new JButton("West"), BorderLayout.WEST);

c.add(new JButton("East"), BorderLayout.EAST);

c.add(new JButton("South"), BorderLayout.SOUTH);

setSize(300, 200); // 프레임 크기 300x200 설정

setVisible(true); // 프레임 화면에 출력

}

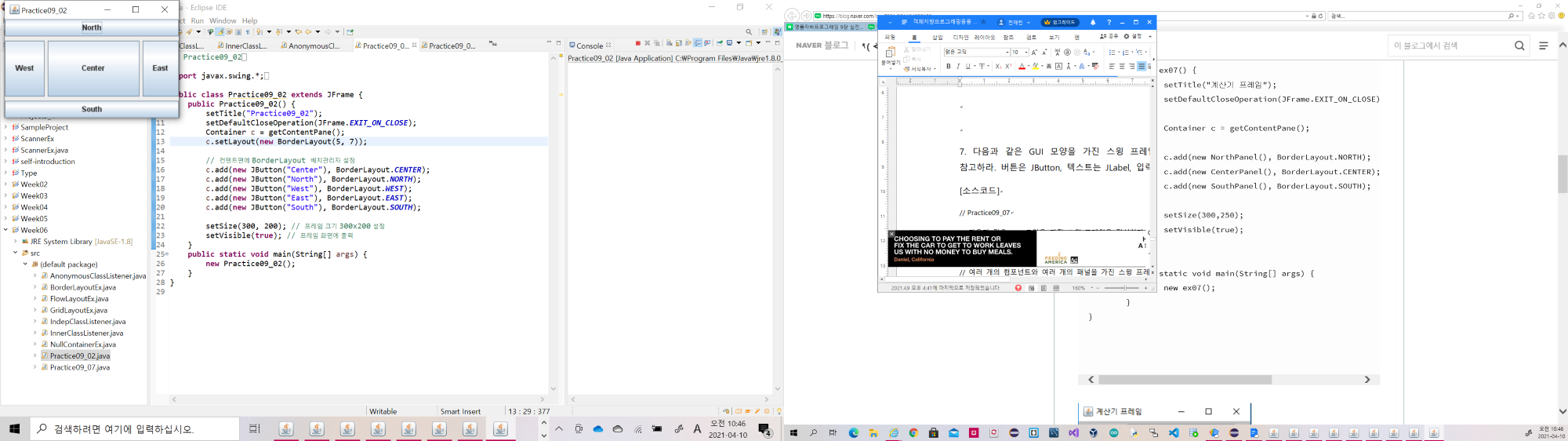
public static void main(String[] args) {

new Practice09\_02();

}

}

[실행결과]



7. 다음과 같은 GUI 모양을 가진 스윙 프레임을 작성하라. Open Challenge의 힌트나 정답을 참고하라. 버튼은 JButton, 텍스트는 JLabel, 입력창은 JTextField를 사용하면 된다.

[소스코드]

// Practice09\_07

// 다음과 같은 GUI 모양을 가진 스윙 프레임을 작성하라. Open Challenge의 힌트나 정답을 참고하라.

// 버튼은 JButton, 텍스트는 JLabel, 입력창은 JTextField를 사용하면 된다.

// 여러 개의 컴포넌트와 여러 개의 패널을 가진 스윙 프레임 만들기

import javax.swing.\*;

import java.awt.\*;

class North extends JPanel {

public North() {

setBackground(Color.LIGHT\_GRAY);

setLayout(new FlowLayout());

add(new JLabel("수식입력"));

add(new JTextField(18));

}

}

class Center extends JPanel {

public Center() {

setLayout(new GridLayout(4, 4, 5, 5));

for(int i=0; i<10; i++) {

add(new JButton(Integer.toString(i)));

}

add(new JButton("CE"));

add(new JButton("계산"));

JButton b[] = {new JButton("+"), new JButton("-"), new JButton("x"), new JButton("/")};

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

b[i].setBackground(Color.BLUE);

add(b[i]);

}

}

}

class South extends JPanel {

public South() {

setBackground(Color.YELLOW);

setLayout(new FlowLayout());

add(new JLabel("계산 결과"));

add(new JTextField(18));

}

}

public class Practice09\_07 extends JFrame {

public Practice09\_07() {

setTitle("계산기 프레임");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.add(new North(), BorderLayout.NORTH);

c.add(new Center(), BorderLayout.CENTER);

c.add(new South(), BorderLayout.SOUTH);

setSize(300, 300);

setVisible(true);

}

public static void main(String[] args) {

new Practice09\_07();

}

}

[실행결과]

