객체지향프로그래밍응용 과제07 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

[예제 11-1]

[소스코드]

// 예제 11-1

// 스윙 컴포넌트의 공통 기능, JComponent의 메소드

// 3개의 버튼을 활용하여 JComponent의 공통 메소드의 사용 사례를 보인다.

// 첫 번째 버튼은 폰트, 배경색, 글자색 활용에 대해, 두 번째 버튼은 활성화/비활성화, 세 번째 버튼은 위치를 다루는 코드로 활용하였다.

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class JComponentEx extends JFrame {

public JComponentEx() {

super("JComponent의 공통 메소드 예제");

Container c = getContentPane();

c.setLayout(new FlowLayout());

JButton b1 = new JButton("Magenta/Yellow Button");

JButton b2 = new JButton("Disabled Button");

JButton b3 = new JButton("getX(), getY()");

b1.setBackground(Color.YELLOW); // 배경색 설정

b1.setForeground(Color.MAGENTA); // 글자색 설정

b1.setFont(new Font("Arial", Font.ITALIC, 20)); // Arial, 20픽셀 폰트 설정

b2.setEnabled(false); // 버튼 비활성화

b3.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

JButton b = (JButton)e.getSource();

JComponentEx frame = (JComponentEx)b.getTopLevelAncestor();

frame.setTitle(b.getX() + ", " + b.getY()); // 타이틀에 버튼 좌표 출력

System.out.println("버튼3을 클릭했어요.." + b.getText() + "높이" + b.getHeight() + "넓이" + b.getWidth());

b.setText("버튼3");

}

});

c.add(b1); c.add(b2); c.add(b3); // 컨텐트팬에 버튼 부착

setSize(260, 200);

setVisible(true);

}

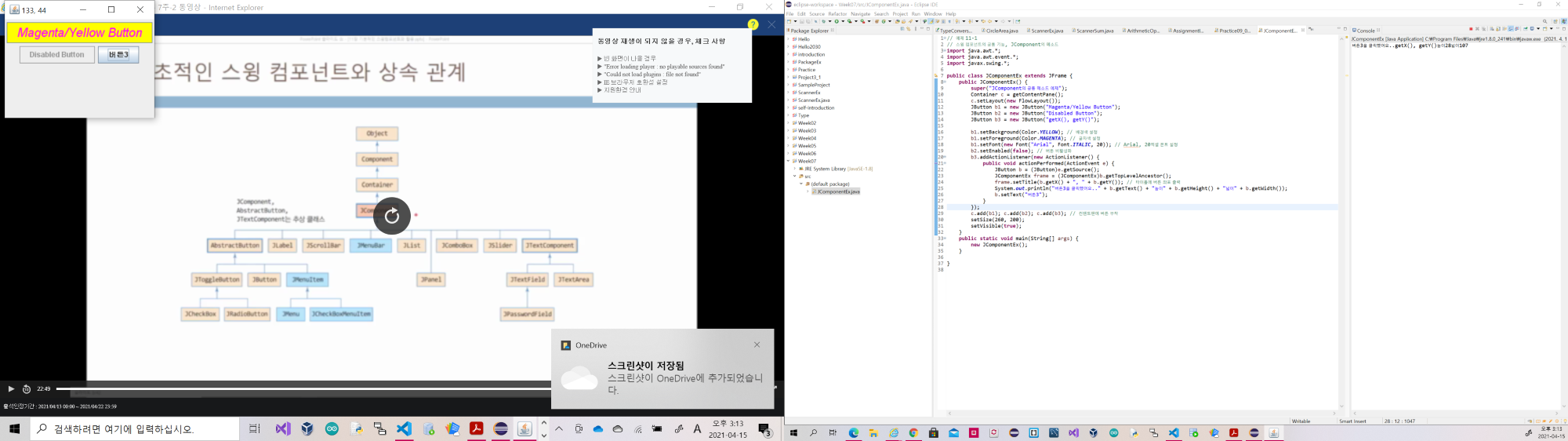
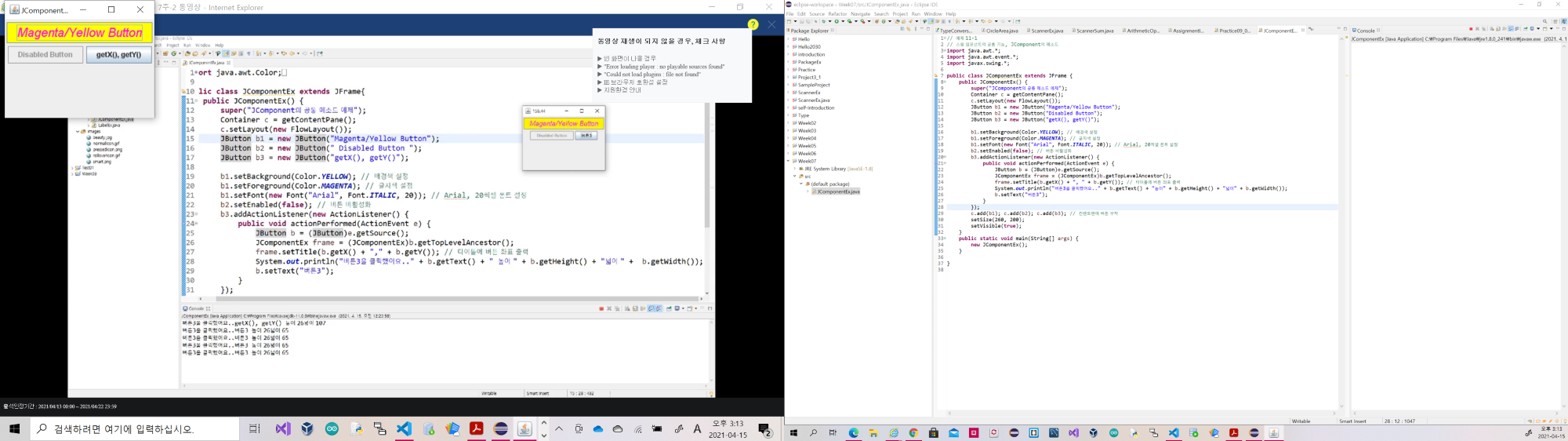
public static void main(String[] args) {

new JComponentEx();

}

}

[실행결과]



[예제 11-2]

[소스코드]

// 예제 11-2

//JLabel을 이용한 레이블 만들기

// JLabel을 이용하여 실행 결과와 같이 문자열과 이미지를 출력하라.

// "사랑합니다" 문자열 레이블과 beauty.jpg 이미지를 출력한 레이블, 그리고 normalIcon.jpg 이미지와 "보고싶으면 전화하세요"를 함께 출력하는 레이블을 작성하라.

// beauty.jpg, normalIcon.jpg 이미지는 프로젝트 폴더 아래 Images 폴더에 저장되어 있어야 한다.

import java.awt.Container;

import java.awt.FlowLayout;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.SwingConstants;

public class LabelEx extends JFrame {

public LabelEx() {

setTitle("레이블 예제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.setLayout(new FlowLayout());

// 문자열 레이블 생성

JLabel textLabel = new JLabel("사랑합니다.");

// 이미지 레이블 생성

ImageIcon beauty = new ImageIcon("images/beauty.jpg"); // 이미지 로딩

JLabel imageLabel = new JLabel(beauty); // 이미지 레이블 생성

// 문자열과 이미지를 모두 가진 레이블 생성

ImageIcon normalIcon = new ImageIcon("images/normalIcon.jpg"); // 이미지 로딩

JLabel label = new JLabel("보고싶으면 전화하세요", normalIcon, SwingConstants.CENTER); // 레이블 생성

// 컨텐트팬에 3개의 레이블 부착

c.add(textLabel);

c.add(imageLabel);

c.add(label);

setSize(400, 600);

setVisible(true);

}

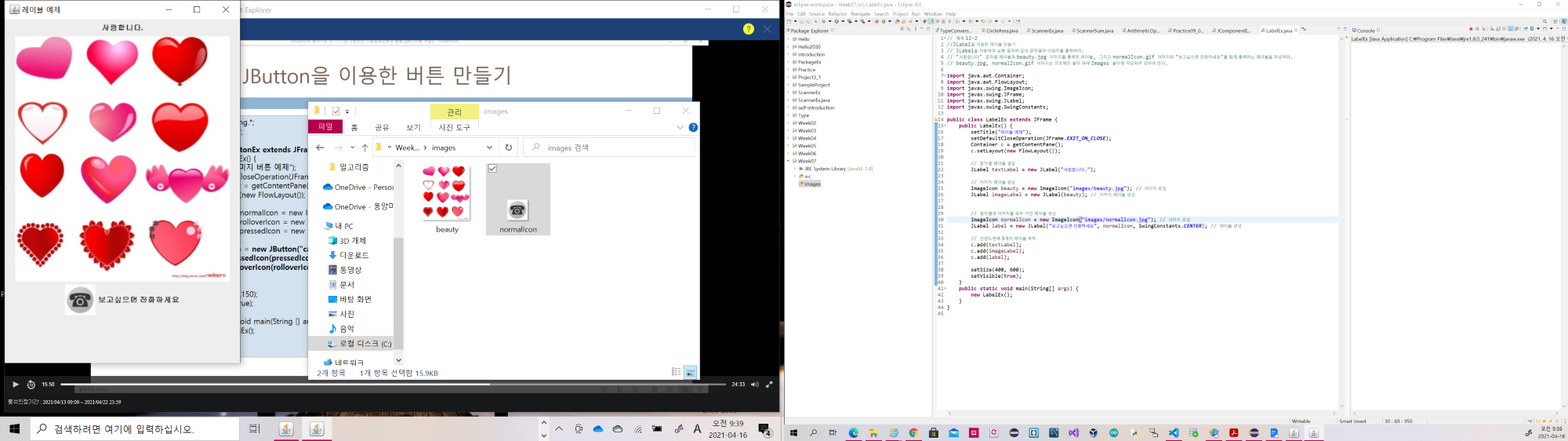
public static void main(String[] args) {

new LabelEx();

}

}

[실행결과]



[예제 11-3]

[소스코드]

// 예제 11-3

// JButton을 이용한 버튼 만들기

// images/normalIcon.jpg, images/rolloverIcon.jpg, images/pressedIcon.jpg 파일로부터 normalIcon, rolloverIocn, pressedIcon을 각각 생성하고

// JButton에 등록하여 다름과 같이 작동하는 버튼을 작성하라

// 이미지 파일은 프로젝트 폴더 밑의 images 폴더에 있어야 한다.

import java.awt.Container;

import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class ButtonEx extends JFrame {

public ButtonEx() {

setTitle("이미지 버튼 에제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.setLayout(new FlowLayout());

// 3개의 이미지를 파일로부터 읽어들인다.

ImageIcon normalIcon = new ImageIcon("images/normalIcon.jpg");

ImageIcon rolloverIcon = new ImageIcon("images/rolloverIcon.jpg");

ImageIcon pressedIcon = new ImageIcon("images/pressedIcon.jpg");

// 3개의 이미지를 가진 버튼 생성

JButton btn = new JButton("call~~", normalIcon); // normalIcon용 이미지 등록

btn.setPressedIcon(pressedIcon); // pressedIcon용 이미지 등록

btn.setRolloverIcon(rolloverIcon); // rolloverIcon용 이미지 등록

JLabel lbText = new JLabel("메시지 : ");

btn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

System.out.println("버튼을 클릭했어요..");

lbText.setText("버튼을 클릭했어요....");

}

});

c.add(btn);

c.add(lbText);

setSize(250, 150);

setVisible(true);

}

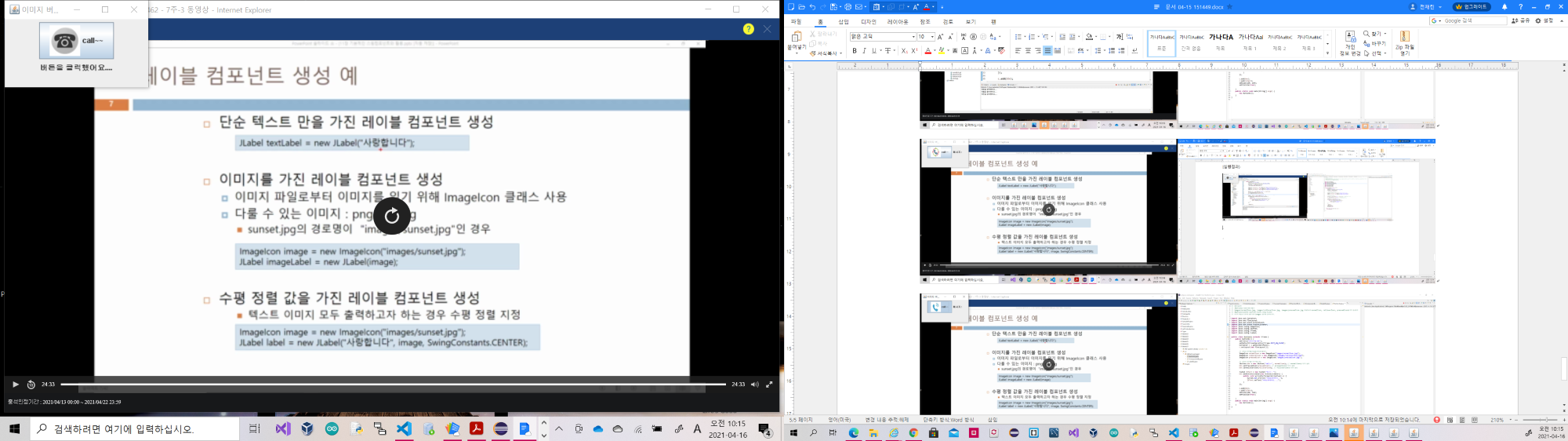
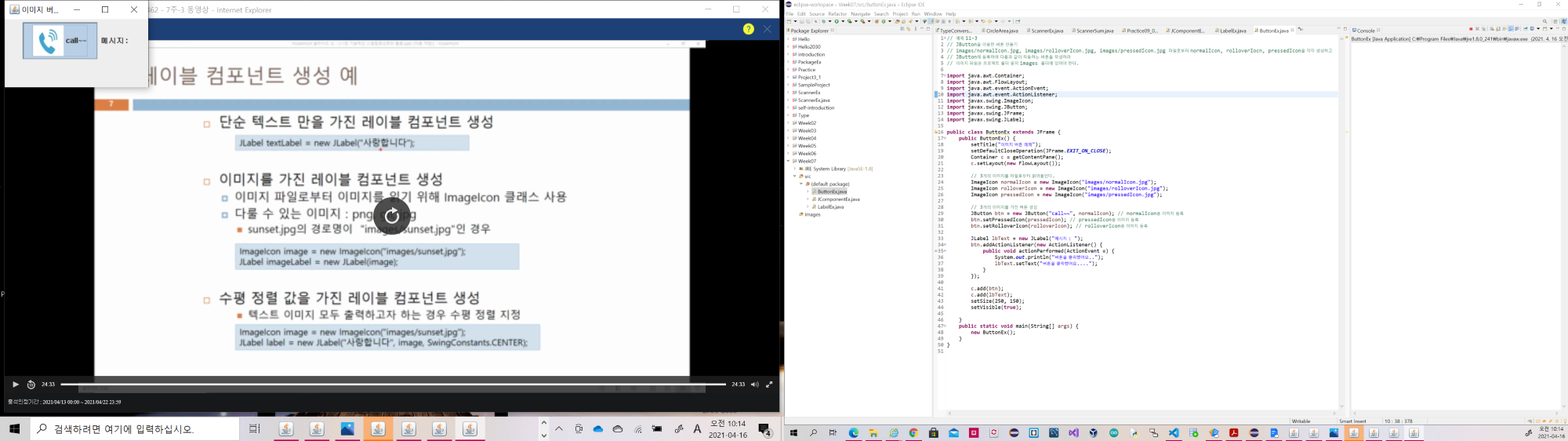
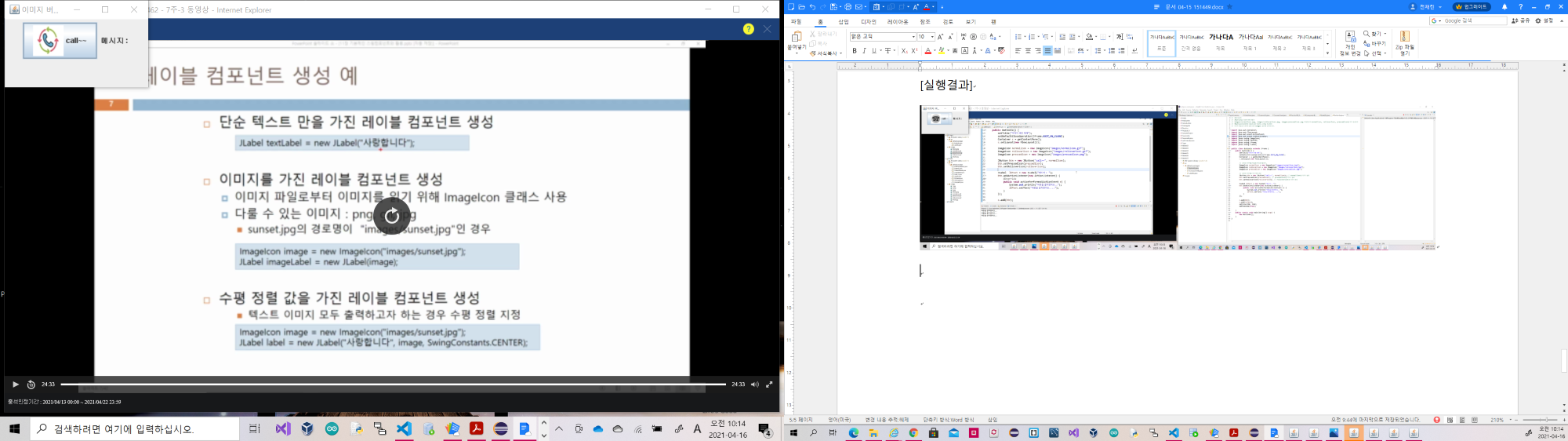
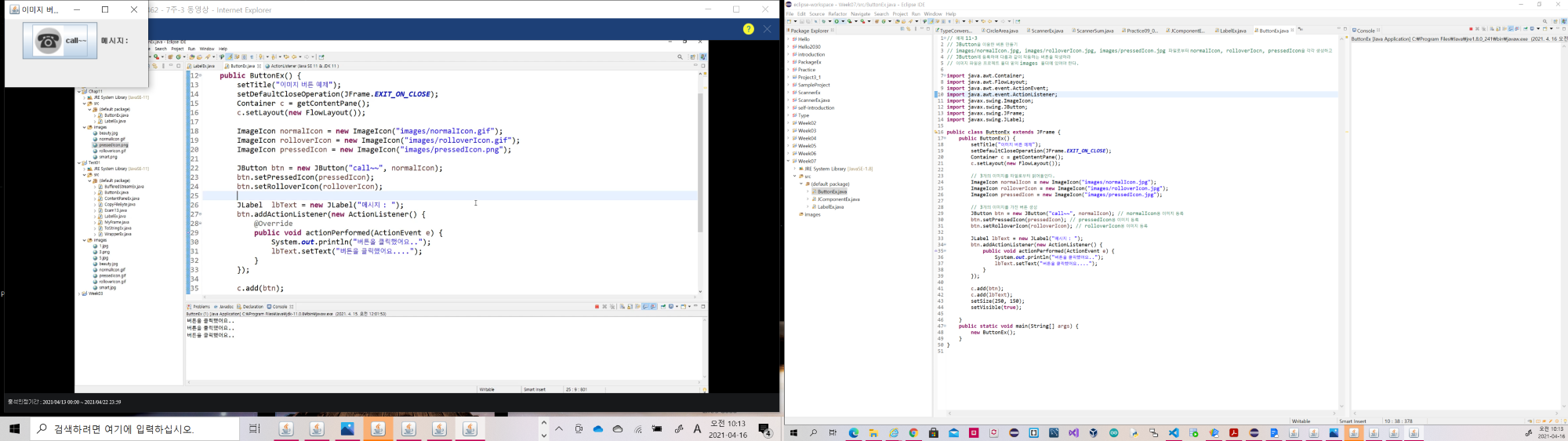
public static void main(String[] args) {

new ButtonEx();

}

}

[실행결과]



[10장 연습문제(이론문제)]

1. 자바의 이벤트 기반 프로그래밍에 대한 설명으로 틀린 것을 골라라.

이벤트 분배 스레드가 존재한다.

AWT나 스윙 응용프로그램은 이벤트 기반 응용프로그램이다.

키 이벤트를 처리하는 도중 마우스 이벤트가 발생하면, 마우스 이벤트롤 처리한 뒤 중단시킨 키 이벤트 처리를 계속 한다.

컴포넌트마다 처리할 수 있는 이벤트가 서로 다르다.

-> , 자바에서 이벤트는 이벤트 분배 스레드에 의해 하나씩 도착하는 순서대로 처리된다. 하나의 이벤트에 대한 처리가 완전히 종료된 후 다음 이벤트가 처리되므로, 한 이벤트 리스너의 처리 시간이 오래 걸리거나 리서너 내에서 잠을 자는 등 시간 지연이 과다하게 되면 다음 이벤트의 처리가 늦어지게 된다.

3. 다음 프로그램 코드를 익명 클래스를 이용하여 다시 작성하라.

JButton btn = new JButton("Hello");

btn.addActionListener(new MyActionListener());

class MyActionListener exteds ActionListener {

public void actionPerformed(ActionEvent e) {

System.out.println("Click");

}

}

JButton btn = new JButton("Hello");

btn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

System.out.println("Click");

}

});

6. 다음 Action 이벤트 리스너 코드가 있다.

class MyActionListener implements ActionListener {

private String msg;

public MyActionListener(String msg) { this.msg = msg; }

public void actionperformed(ActionEvent e) { System.out.println(msg); }

}

“Hello” 버튼에 다음과 같이 리스너를 등록하고, 버튼을 클릭하면 실행 결과는 무엇인가?

JButton btn = new JButton("Hello");

btn.addActionListener(new MyActionListener("1"));

btn.addActionListener(new MyActionListener("2"));

btn.addActionListener(new MyActionListener("3"));

3 2 1이 순서대로 출력된다.

1 2 3이 순서대로 출력된다.

3만 출력된다.

1만 출력된다.

-> , 컴포넌트는 동일한 이벤트에 대해 여러 개의 이벤트 리스너를 가질 수 있다. 위의 경우 btn에 Action 이벤트가 발생하면 3개의 리스너가 모두 실행되며, 이들은 등록된 반대순으로 실행된다.

[11장 연습문제(이론문제)]

1. 다음 중 스윙 컴포넌트가 가진 정보가 아닌 것은?

컴포넌트 자신의 위치

컴포넌트 자신의 배경색

컴포넌트가 만들어진 시간

컴포넌트의 부모 컨테이너

-> , 컴포넌트의 모양과 관련된 메소드, 컴포넌트의 상태와 관련된 메소드, 컴포넌트의 위치와 크기에 관련된 메소드, 컨테이너를 위한 메소드가 존재한다.

2. JLabel은 어떤 목적으로 사용하는가?

이미지나 텍스트 출력

이미지만 출력

이미지, 텍스트, 버튼 출력

텍스트만 출력

-> , JLabel은 이미지나 문자열을 출력하는 컴포넌트이다.

3. “java.jpg”를 가진 JLabel 컴포넌트를 생성하는 간단한 코드이다. 빈칸을 채워라.

ImageIcon icon = new ImageIcon("java.jpg"); // java.jpg 파일을 로딩한다.

JLabel label = new JLabel(); // 빈 JLabel 컴포넌트를 생성한다.

label.setIcon(icon); // 이미지를 레이블에 부착한다.

4. 스윙 컴포넌트 c에 대해 다음 빈칸을 채워라.

c.setVisible(false); // 컴포넌트가 보이지 않도록 만든다.

c.setFont(new Font("고딕", Font.PLAIN, 20)); // 글자체를 20픽셀의 고딕체로 한다.

c.setEnable(false); // 컴포넌트가 마우스나 키보드로 입력해도 반응이 없게 한다.

[11장 OpenChallenge 이미지 갤러리 만들기]

버튼과 이미지 레이블, Action 리스너를 활용하여 이미지를 순환하면서 보여주는 이미지 갤러리를 작성해보자. 아래에는 4개의 실행 화면이 있다. 컨텐트팬의 배치관리자를 BorderLayout 배치관리자로 설정하고 CENTER 영역에는 이미지를 출력하는 레이블 컴포넌트를 단다. SOUTH 영역에는 JPanel을 상속받은 MenuPanel을 작성하고 이곳에 좌(<-), 우(->) 화살표 이미지를 가진 2개의 버튼을 단다. 오른쪽 화살표 버튼을 클릭하면 CENTER에 있는 이미지 레이블에 다음 이미지를 출력하고 반대로 왼쪽 화살표 버튼을 클릭하면 이전 이미지를 레이블에 출력한다.

[소스코드]

// OpenChallenge11

// 버튼과 이미지 레이블, Action 리스너를 활용하여 이미지를 순환하면서 보여주는 이미지 갤러리를 작성해보자.

// 아래에는 4개의 실행 화면이 있다.

// 컨텐트팬의 배치관리자를 BorderLayout 배치관리자로 설정하고 CENTER 영역에는 이미지를 출력하는 레이블 컴포넌트를 단다.

// SOUTH 영역에는 JPanel을 상속받은 MenuPanel을 작성하고 이곳에 좌(<-), 우(->) 화살표 이미지를 가진 2개의 버튼을 단다.

// 오른쪽 화살표 버튼을 클릭하면 CENTER에 있는 이미지 레이블에 다음 이미지를 출력하고 반대로 왼쪽 화살표 버튼을 클릭하면 이전 이미지를 레이블에 출력한다.

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class OpenChallenge11 extends JFrame {

// 이미지를 images 배열에 저장

private ImageIcon[] images = { new ImageIcon("images/gallery0.jpg"), new ImageIcon("images/gallery1.jpg"),

new ImageIcon("images/gallery2.jpg"), new ImageIcon("images/gallery3.jpg") };

// 레이블의 초기 이미지는 images 배열의 첫번째 요소로 설정

private int index = 0;

private JLabel imageLabel = new JLabel(images[index]);

private JButton lbtn = null; // 왼쪽 화살표 버튼

private JButton rbtn = null; // 오른쪽 화살표 버튼

public OpenChallenge11() {

Container c = getContentPane();

c.setLayout(new BorderLayout(10, 10));

// 이미지 라벨은 CENTER에 부착

c.add(imageLabel, BorderLayout.CENTER);

// 메뉴는 SOUTH에 부착

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

setTitle("Open Challenge 11");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 550);

setVisible(true);

}

class MenuPanel extends JPanel { // JPanel을 상속받은 MenuPanel

public MenuPanel() { // 생성자

setLayout(new FlowLayout());

setBackground(Color.GRAY);

ImageIcon leftbtn = new ImageIcon("images/leftButton.jpg");

ImageIcon rightbtn = new ImageIcon("images/rightButton.png");

lbtn = new JButton(leftbtn);

rbtn = new JButton(rightbtn);

add(lbtn); // 왼쪽 화살표 버튼 추가

add(rbtn); // 오른쪽 화살표 버튼 추가

// 두 버튼에 Action 리스너 추가

lbtn.addActionListener(new MyActionListener());

rbtn.addActionListener(new MyActionListener());

}

}

class MyActionListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

JButton btn = (JButton)e.getSource();

if(btn == lbtn) { // 왼쪽 화살표 버튼이 클릭될 경우

if(index == 0)

index = images.length - 1; // index가 0이었다면 인덱스 최댓값으로 설정

else

index--; // 나머지 경우는 인덱스 1 감소

}

else { // 오른쪽 화살표 버튼이 클릭된 경우

if(index == images.length - 1)

index = 0; // index가 최댓값이었다면 0으로 설정

else

index++; // 나머지 경우는 인덱스 1 증가

}

imageLabel.setIcon(images[index]); // index 수정 후 해당 인덱스의 이미지로 라벨 변경

}

}

public static void main(String[] args) {

new OpenChallenge11();

}

}

[실행결과]

