객체지향프로그래밍응용 과제11 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

[예제12-1]

JPanel을 상속받아 만든 패널을 컨텐트팬으로 붙이고, 3개의 사각형을 그리는 응용프로그램

[소스코드]

// 예제 12-1

// JPanel을 상속받아 도형 그리기

// JPanel을 상속받아 만든 패널을 컨텐트팬으로 붙이고, 3개의 사각형을 그려라.

import javax.swing.\*;

import java.awt.\*;

public class paintJPanelEx extends JFrame {

private MyPanel panel = new MyPanel();

public paintJPanelEx() {

setTitle("JPanel의 paintComponent() 예제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.add(panel);

// setContentPane(panel); // 생성한 panel 패널을 컨텐트팬으로 사용

setSize(250, 220);

setVisible(true);

}

// JPanel을 상속받는 새 패널 구현

class MyPanel extends JPanel {

public void paintComponent(Graphics g) { // JPanel의 내부를 그리는 메소드로서 3개의 사각형을 그린다.

super.paintComponent(g); // JPanel의 paintComponent() 호출. 패널 내에 이전에 그려진 잔상을 지우기 위해 호출한다.

g.setColor(Color.BLUE); // 파란색 선택

g.drawRect(10, 10, 50, 50); // (10,10) 위치에 50x50 크기의 사각형 그리기

g.drawRect(50, 50, 50, 50); // (50,50) 위치에 50x50 크기의 사각형 그리기

g.setColor(Color.MAGENTA); // 마젠타색 선택

g.drawRect(90, 90, 50, 50); // (90,90) 위치에 50x50 크기의 사각형 그리기

}

}

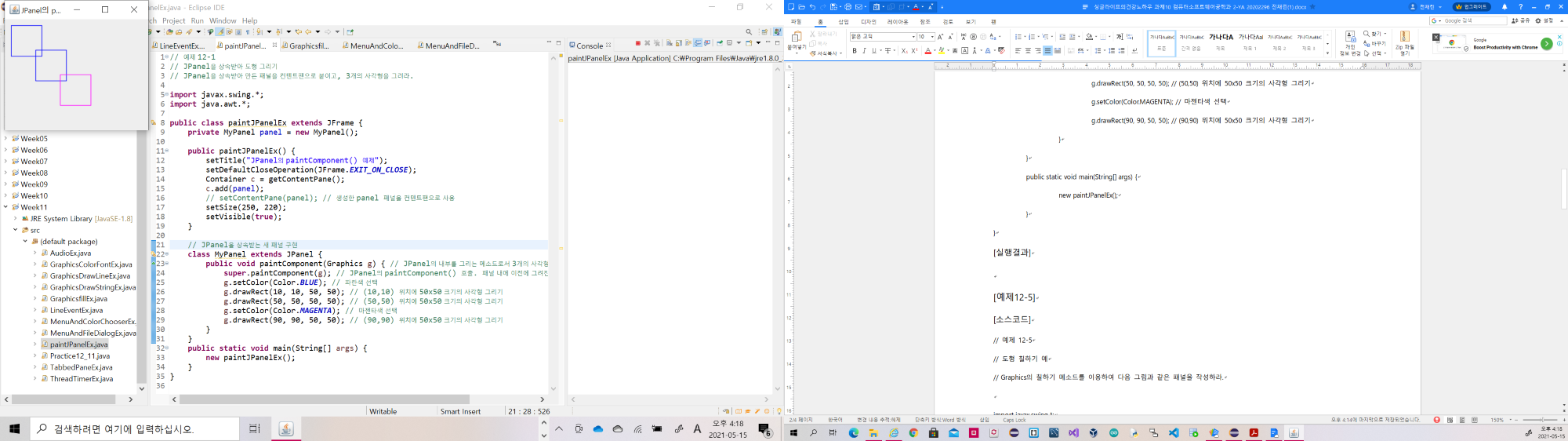
public static void main(String[] args) {

new paintJPanelEx();

}

}

[실행결과]



[예제12-5]

Graphics의 칠하기 메소드를 이용하여 도형을 칠하는 응용프로그램

[소스코드]

// 예제 12-5

// 도형 칠하기 예

// Graphics의 칠하기 메소드를 이용하여 다음 그림과 같은 패널을 작성하라.

import javax.swing.\*;

import java.awt.\*;

public class GraphicsfillEx extends JFrame {

private MyPanel panel = new MyPanel();

public GraphicsfillEx() {

setTitle("fillXXX 사용 예제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.add(panel);

// setContentPane(panel); // 생성한 panel 패널을 켄텐트팬으로 사용

setSize(100, 350);

setVisible(true);

}

// JPanel을 상속받는 새 패널

class MyPanel extends JPanel {

public void paintComponent(Graphics g) {

super.paintComponent(g);

g.setColor(Color.RED);

g.fillRect(10, 10, 50, 50); // 빨간색 사각형 칠하기

g.setColor(Color.BLUE);

g.fillOval(10, 70, 50, 50); // 파란색 타원 칠하기

g.setColor(Color.GREEN);

g.fillRoundRect(10, 130, 50, 50, 20, 20); // 초록색 둥근 사각형 칠하기

g.setColor(Color.MAGENTA);

g.fillArc(10, 190, 50, 50, 0, 270); // 마젠타색 원호 칠하기

g.setColor(Color.ORANGE);

int [] x = {30, 10, 30, 60};

int [] y = {250, 275, 300, 275};

g.fillPolygon(x, y, 4); // 오렌지색 다각형 칠하기

}

}

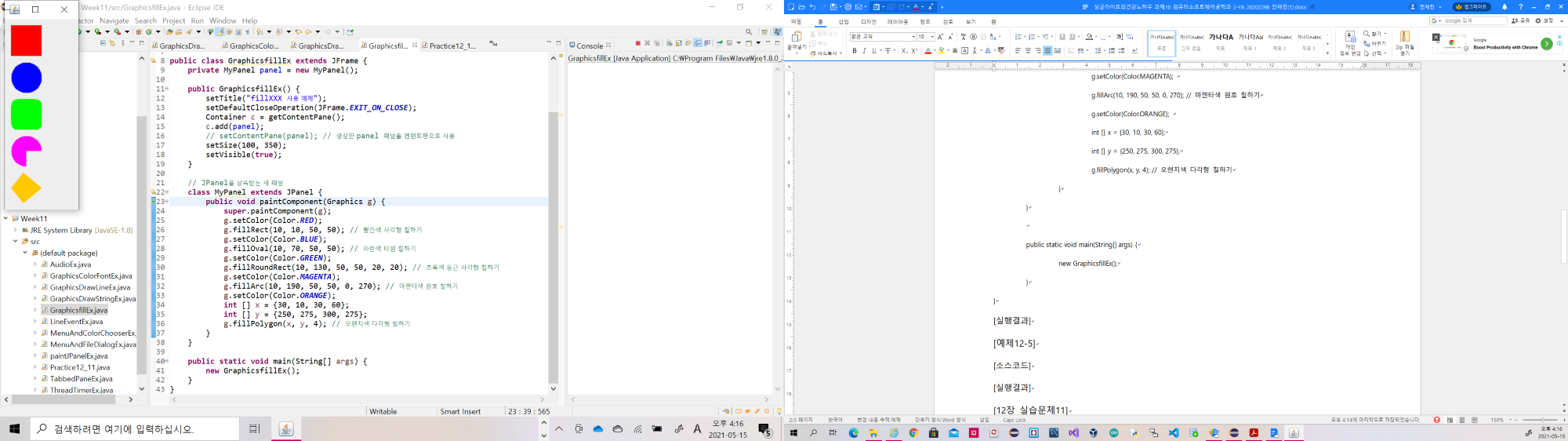
public static void main(String[] args) {

new GraphicsfillEx();

}

}

[실행결과]



[12장 실습문제11]

apple, cherry, strawberry, prune의 4가지 종류의 과일 판매량을 입력하고 <Enter>키를 치면 전체 판매량에서 백분율(%)을 계산하여 문자열과 파이 차트로 동시에 출력하는 응용프로그램

[소스코드]

// 자바 프로그래밍 12장 실습문제 11

// 파이 차트를 만들어보자.

// 다음 그림과 같이 apple, cherry, strawberry, prune의 4가지 종류의 과일 판매량을 입력하고 <Enter>키를 치면

// 전체 판매량에서 백분율(%)을 계산하여 문자열과 파이 차트로 동시에 출력되게 하라.

// JTextField 창에 <Enter>키를 치면 Action 이벤트가 발생하는데 Action 리스너를 작성하여 백분율을 계산하고 파이 차트를 그리면 된다.

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Practice12\_11 extends JFrame {

private int [] data = {0, 0, 0, 0};

private int [] arcAngle = new int [4];

private Color [] color = {Color.RED, Color.BLUE, Color.MAGENTA, Color.ORANGE};

private String [] itemName = {"apple", "cherry", "strawberry", "prune"};

private JTextField [] tfIn = new JTextField [4];

private ChartPanel chartPanel = new ChartPanel();

private JPanel inputPanel = new JPanel(); // 과일 개수 입력 창

public Practice12\_11() {

setTitle("20202296 전채린");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

createGUI();

c.add(inputPanel, BorderLayout.NORTH);

c.add(chartPanel, BorderLayout.CENTER);

setSize(500, 350);

setVisible(true);

drawChart();

}

// 과일 개수 입력 만들기 & 리스너 설정

private void createGUI() {

inputPanel.setBackground(Color.LIGHT\_GRAY);

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

tfIn[i] = new JTextField("0", 5);

tfIn[i].addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JTextField t = (JTextField)e.getSource();

int n;

try {

n = Integer.parseInt(t.getText());

} catch(NumberFormatException ex) {

t.setText("");

return;

}

drawChart();

}

});

inputPanel.add(new JLabel(itemName[i]));

inputPanel.add(tfIn[i]);

}

};

private void drawChart() {

int sum=0;

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

data[i] = Integer.parseInt(tfIn[i].getText());

sum += data[i];

}

if(sum == 0) return;

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

arcAngle[i] = (int)Math.round((double)data[i] / (double)sum\*360);

chartPanel.repaint();

}

private class ChartPanel extends JPanel {

public void paintComponent(Graphics g) {

super.paintComponent(g);

int startAngle = 0;

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

g.setColor(color[i]);

g.drawString(itemName[i] + " " + Math.round(arcAngle[i]\*100./360.) + "%", 50+i\*100, 20);

}

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

g.setColor(color[i]);

g.fillArc(150, 50, 200, 200, startAngle, arcAngle[i]);

startAngle += arcAngle[i];

}

}

}

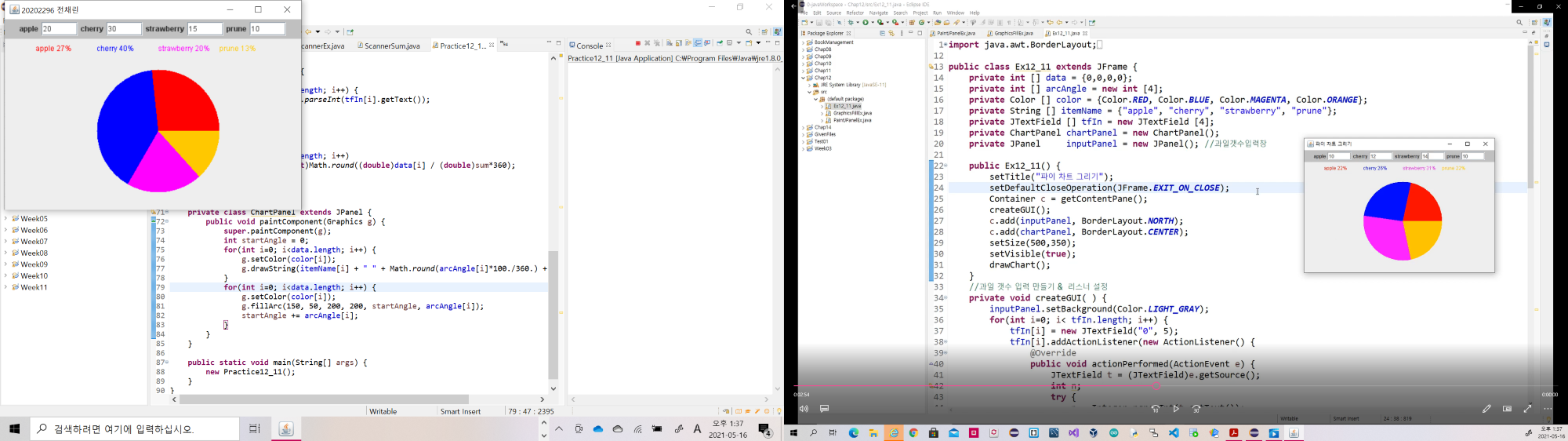
public static void main(String[] args) {

new Practice12\_11();

}

}

[실행결과]



[예제14-9]

JFileChooser를 이용하여 파일 열기 다이얼로그를 출력하고 사용자가 선택한 이미지 파일을 화면에 그리는 응용프로그램

[소스코드]

// 예제 14-9

// 파일 열기 다이얼로그 생성 및 출력

// JFileChooser를 이용하여 파일 열기 다이얼로그를 출력하고 사용자가 선택한 이미지 파일을 화면에 그리는 응용프로그램을 작성하라.

// 다음 그림과 같이 메뉴를 만들고, Open 메뉴아이템이 선택되면 파일 열기 다이얼로그를 출력하라.

// 사용자로부터 이미지 파일의 이름을 입력받아 JLabel 컴포넌트를 이용항 이미지를 출력한다.

// jpg나 gif형식의 파일을 필터링하도록 파일 필터를 만들어서 사용하라.

import javax.swing.\*;

import javax.swing.filechooser.\*;

import java.awt.\*;

import java.awt.event.\*;

public class MenuAndFileDialogEx extends JFrame {

private JLabel imageLabel = new JLabel();

private JFileChooser chooser = new JFileChooser();

public MenuAndFileDialogEx() {

setTitle("Menu와 JFileChooser 활용 예제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.add(imageLabel);

createMenu();

setSize(350, 200);

setVisible(true);

}

private void createMenu() {

JMenuBar mb = new JMenuBar();

JMenu fileMenu = new JMenu("File");

JMenuItem openItem = new JMenuItem("Open");

// Open 메뉴아이템에 Action 리스너를 등록한다.

openItem.addActionListener(new ActionListener() {

// Open 메뉴아이템이 선택되면 호출되는 Action 리스너

@Override

public void actionPerformed(ActionEvent e) {

FileNameExtensionFilter filter = new FileNameExtensionFilter(

"JPG & GIF Images", // 파일 이름난에 출력될 문자열

"jpg", "gif"); // 파일 필터로 사용되는 확장자. \*.jpg. \*.gif만 나열됨

chooser.setFileFilter(filter); // 파일 다이얼로그에 파일 필터 설정

int ret = chooser.showOpenDialog(null); // 파일 다이얼로그 출력

if(ret != JFileChooser.APPROVE\_OPTION) { // 사용자가 창을 강제로 닫았거나 취소 버튼을 누른 경우

JOptionPane.showMessageDialog(null, "파일을 선택하지 않았습니다", "경고", JOptionPane.WARNING\_MESSAGE);

return;

}

// 사용자가 파일을 선택하고 "열기" 버튼을 누른 경우

String filePath = chooser.getSelectedFile().getPath(); // 파일 경로명 리턴

imageLabel.setIcon(new ImageIcon(filePath)); // 이미지 출력

pack(); // 이미지의 크기에 맞추어 프레임 크기 조절

}

});

fileMenu.add(openItem);

mb.add(fileMenu);

setJMenuBar(mb);

}

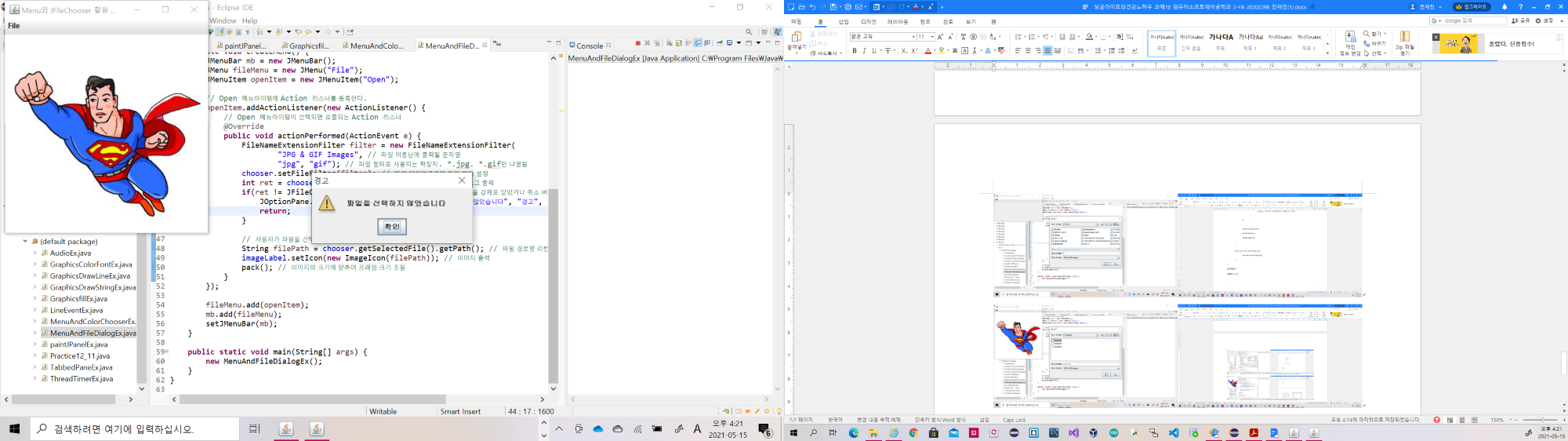
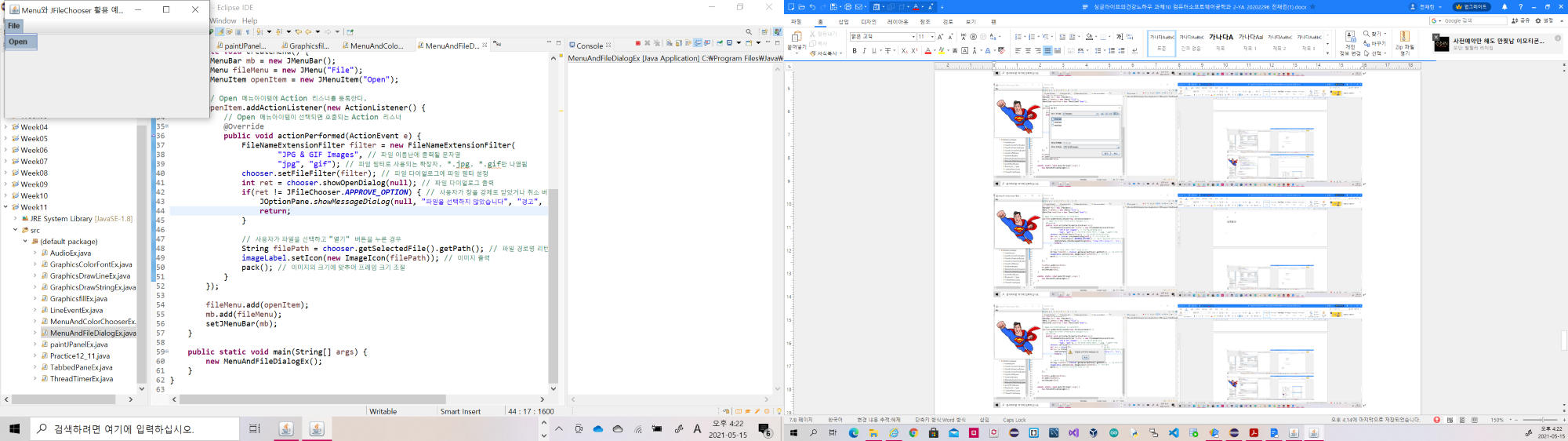
public static void main(String[] args) {

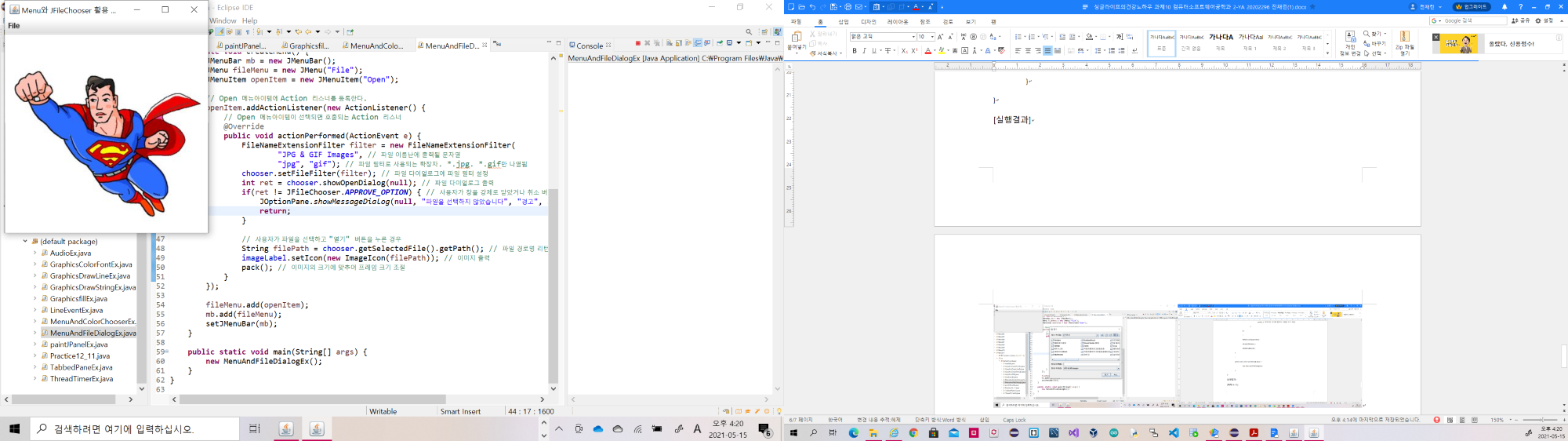
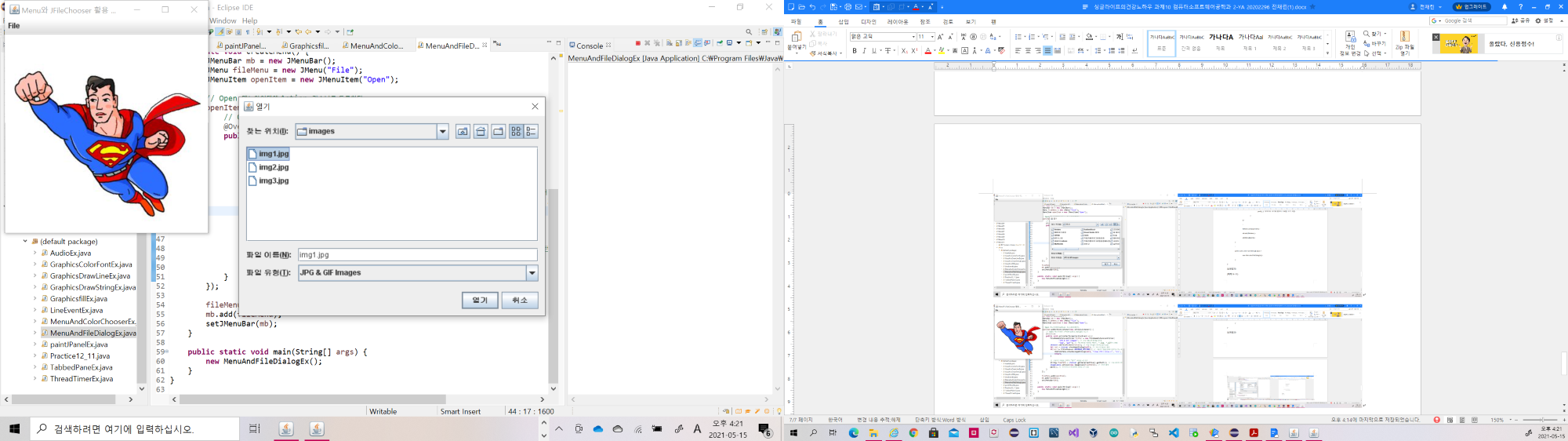
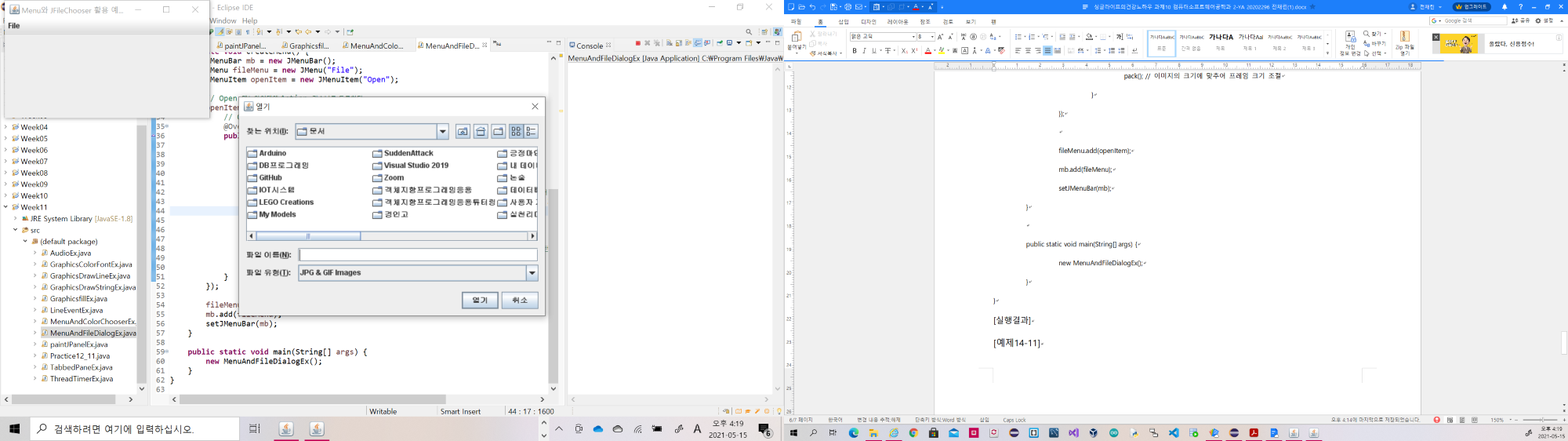
new MenuAndFileDialogEx();

}

}

[실행결과]





[예제14-11]

3개의 탭을 가진 탭팬을 만드는 응용프로그램

[소스코드]

// 예제 14-11

// 탭팬 만들기

// 다음 그림과 같이 3개의 탭을 가진 탭팬을 작성하라.

import javax.swing.\*;

import java.awt.\*;

public class TabbedPaneEx extends JFrame {

public TabbedPaneEx() {

setTitle("탭팬 만들기 예제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

JTabbedPane pane = new JTabbedPane(); // 탭팬 객체를 생성한다.

pane.addTab("tab1", new JLabel(new ImageIcon("images/img1.jpg"))); // 첫 번째 탭

pane.addTab("tab1", new JLabel(new ImageIcon("images/img2.jpg"))); // 두 번째 탭

pane.addTab("tab3", new JLabel(new ImageIcon("images/img3.jpg"))); // 세 번째 탭

c.add(pane, BorderLayout.CENTER); // 탭팬을 컨텐트팬에 부착한다.

setSize(250, 250);

setVisible(true);

}

/\*

// 탭팬을 생성하고 3개의 탭을 생성하여 부착한다.

private JTabbedPane createTabbedPane() {

JTabbedPane pane = new JTabbedPane(); // 탭팬 객체를 생성한다.

pane.addTab("tab1", new JLabel(new ImageIcon("images/img1.jpg"))); // 첫 번째 탭

pane.addTab("tab1", new JLabel(new ImageIcon("images/img2.jpg"))); // 두 번째 탭

pane.addTab("tab3", new JLabel(new ImageIcon("images/img3.jpg"))); // 세 번째 탭

// pane.addTab("tab1", new MyPanel()); // 세 번째 탭

return pane;

}

// 3번째 탭의 컴포넌트로 사용될 패널

class MyPanel extends JPanel {

public MyPanel() {

this.setBackground(Color.YELLOW);

}

public void paintComponent(Graphics g) {

super.paintComponent(g);

g.setColor(Color.RED);

g.fillRect(10, 10, 50, 50);

g.setColor(Color.BLUE);

g.fillRect(10, 70, 50, 50);

g.setColor(Color.BLACK);

g.drawString("tab 3에 들어가는 JPanel 입니다. ", 30, 50);

}

}

\*/

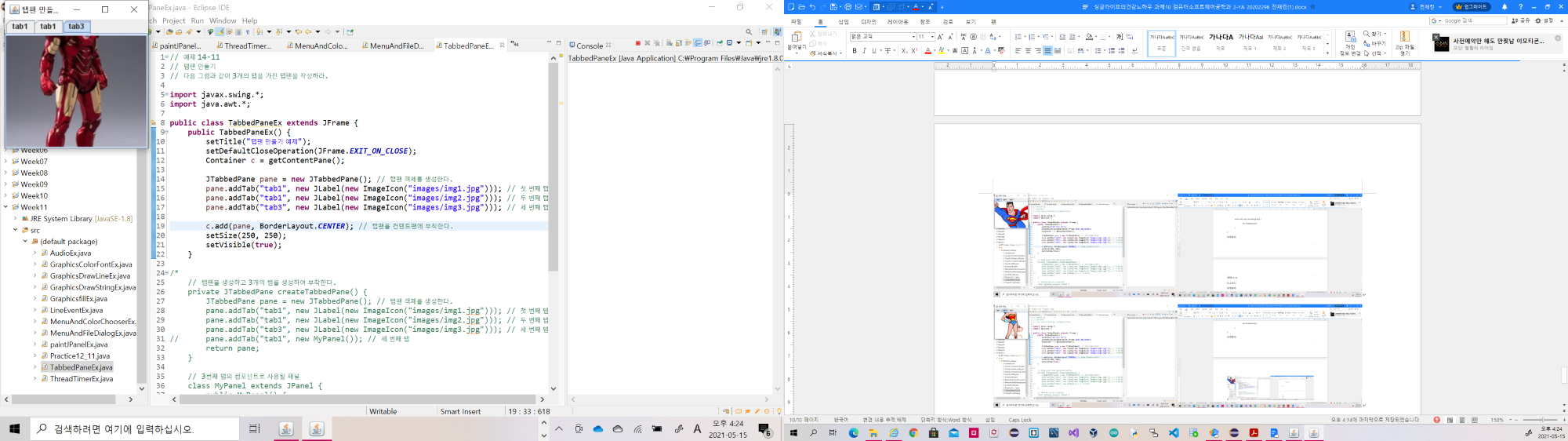
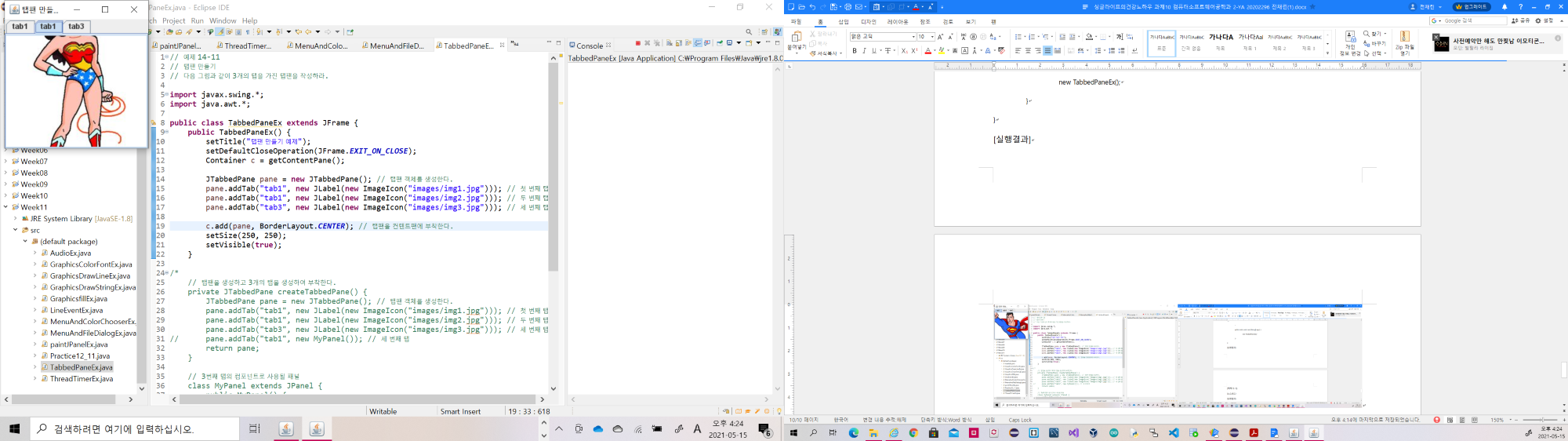
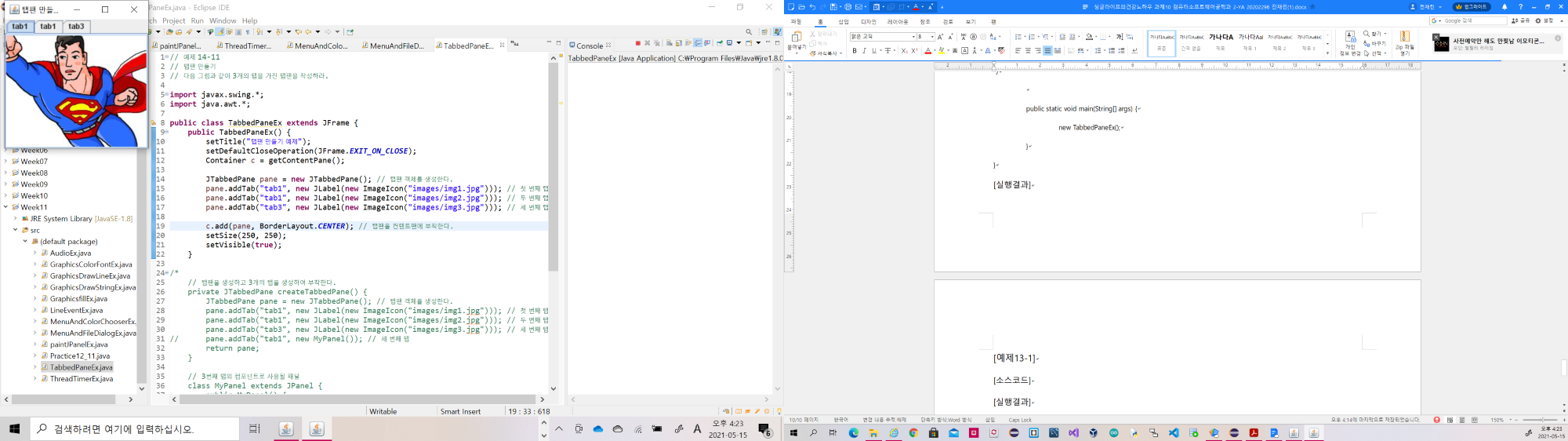
public static void main(String[] args) {

new TabbedPaneEx();

}

}

[실행결과]



[예제13-1]

Thread를 상속받아 1초 단위로 출력하는 타이머 스레드를 만드는 응용프로그램

[소스코드]

// 예제 13-1

// Thread를 상속받아 1초 단위로 출력하는 타이머 스레드 만들기

// 1초 단위로 증가하는 스레드를 만들고 레이블을 이용하여 타이머 값을 출력한다.

// 응용프로그램의 시작과 함께 타이머가 바로 작동한다.

// 타이머 값은 JLabel 컴포넌트를 생성하여 출력하고, Thread 클래스를 상속받는 TimerThread 클래스로 스레드를 만든다.

// 이 스레드는 1초마다 1씩 증가시킨 정수 값을 JLabel의 텍스트에 출력한다.

import java.awt.\*;

import javax.swing.\*;

class TimerThread extends Thread {

private JLabel timerLabel; // 타이머 값이 출력되는 레이블

public TimerThread(JLabel timerLabel) {

this.timerLabel = timerLabel; // 타이머 카운트를 출력할 레이블

}

// 스레드 코드. run()이 종료하면 스레드 종료

public void run() { // run()은 스레드 코드로서 start() 메소드가 호출된 후 스레드가 실행을 시작하는 메소드이다.

int n=0; // 타이머 카운트 값

while(true) { // 무한 루프

timerLabel.setText(Integer.toString(n));

n++; // 카운트 증가

try {

Thread.sleep(1000); // 1초 동안 잠을 잔다.

}

catch(InterruptedException e) {

return; // 예외가 발생하면 스레드 종료

}

}

}

}

public class ThreadTimerEx extends JFrame {

public ThreadTimerEx() {

setTitle("Thread를 상속받은 타이머 스레드 예제");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container c = getContentPane();

c.setLayout(new FlowLayout());

// 타이머 값을 출력할 레이블 생성

JLabel timerLabel = new JLabel();

timerLabel.setFont(new Font("Gothic", Font.ITALIC, 80));

c.add(timerLabel); // 레이블을 컨텐트팬에 부착

// 타이머 스레드 객체 생성. 타이머 값을 출력할 레이블을 생성자에게 전달

TimerThread th = new TimerThread(timerLabel); // 스레드 객체를 만든다.

setSize(300, 170);

setVisible(true);

th.start(); // 스레드를 동작시킨다. 이 호출의 결과 TimerThread의 run() 메소드가 실행을 시작한다.

}

public static void main(String[] args) {

new ThreadTimerEx();

}

}

[실행결과]

