

AWT 와 Swing



AWT(Abstract Window Toolkit)

GUI 프로그래밍을 위한 도구

GUI 프로그래밍에 필요한 다양한 컴포넌트를 제공한다.

OS 위에서 동작한다. 따라서, OS의 종류에 따라 보이는 모습이 달라질 수 있다.

Swing

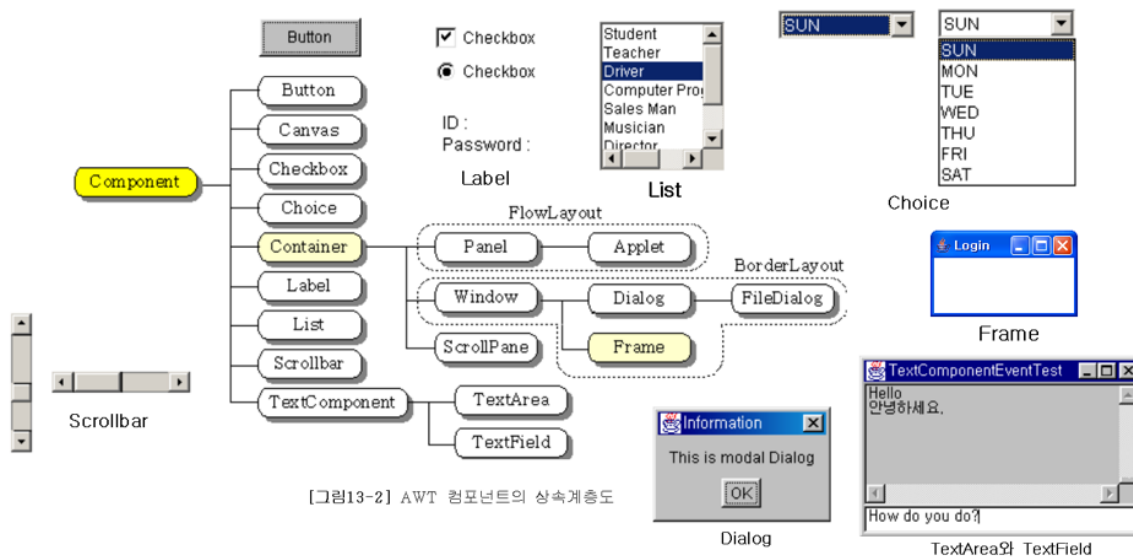
AWT를 확장한 GUI 프로그래밍 도구

AWT보다 더 많은 종류의 컴포넌트를 제공한다.

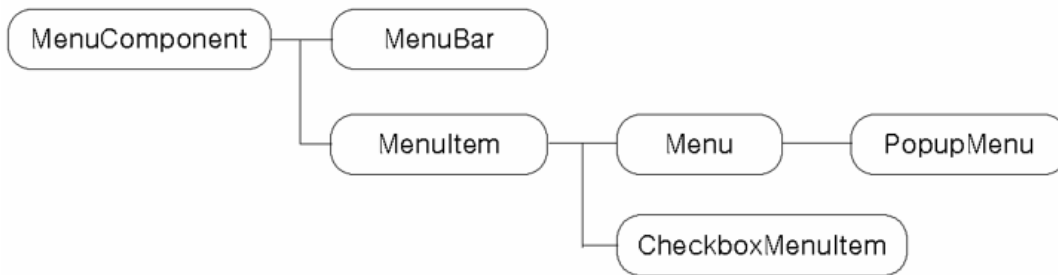
JAVA 위에서 동작한다. 따라서, OS에 상관없이 모습이 동일하다.

AWT의 구성

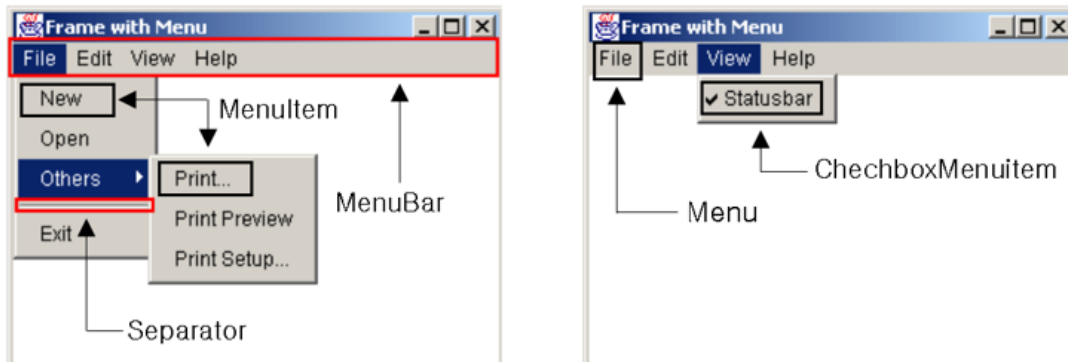
모든 AWT 컴포넌트의 최고 조상은 `java.awt.Component` 클래스이다.



메뉴관련 컴포넌트의 최고 조상은 `java.awt.MenuComponent` 클래스이다.



[그림13-4] AWT 메뉴컴포넌트의 상속계통도

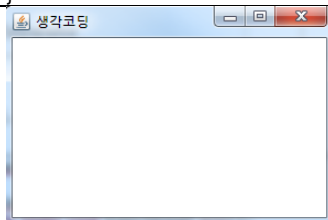


Frame 만들기 - 실습하기

```

import java.awt.Frame;

class FrameTest {
    public static void main(String[] args) {
        Frame t = new Frame("생각코딩");
        t.setSize(300, 200);
        t.setVisible(true);
    }
}
  
```

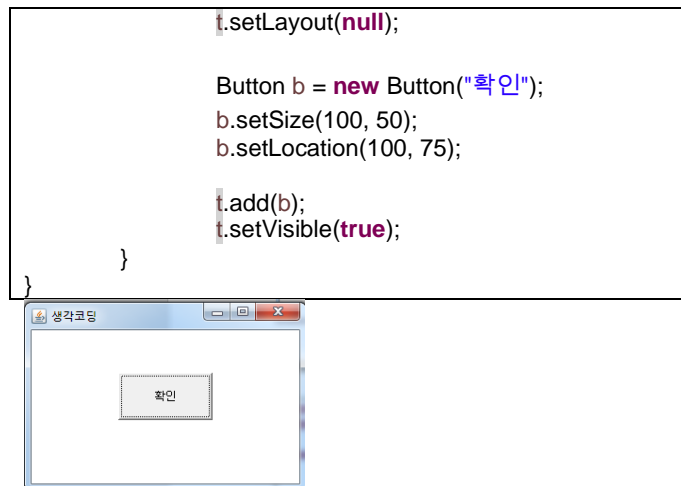


Button 추가하기 - 실습하기

```

import java.awt.Button;
import java.awt.Frame;

class FrameTest {
    public static void main(String[] args) {
        Frame t = new Frame("생각코딩");
        t.setSize(300, 200);
    }
}
  
```



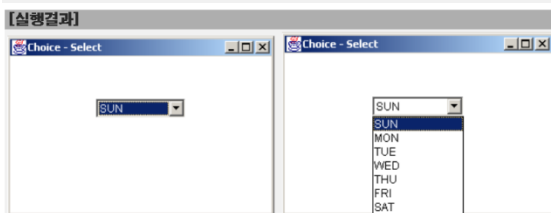
Choice 추가하기 - 실습하기

```
public static void main(String args[]) {
    Frame f = new Frame("Choice - Select");
    f.setSize(300, 200);
    f.setLayout(null);

    Choice day = new Choice(); // Choice객체를 생성한 다음
    day.add("SUN");           // Choice의 목록에 나타날 값들을 추가한다.
    day.add("MON");           day.add("TUE");
    day.add("WED");           day.add("THU");
    day.add("FRI");           day.add("SAT");

    day.setSize(100, 50);
    day.setLocation(100, 70);

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



Label 추가하기 - 실습하기

```
public static void main(String args[]) {
    Frame f = new Frame("Login");
    f.setSize(300, 200);
    f.setLayout(null);

    Label id = new Label("ID :");        // Label을 생성하고 크기와 위치를 지정한다.
    id.setBounds(50, 50, 30, 10);        // 50, 50위치에 크기가 가로 30, 세로 10

    Label pwd = new Label("Password :");
    pwd.setBounds(50, 65, 100, 10);

    f.add(id);                            // 생성한 Label을 Frame에 포함시킨다.
    f.add(pwd);
    f.setVisible(true);
}
```



Checkbox 추가하기 - 실습하기

```
public static void main(String args[]) {
    Frame f = new Frame("Questions");
    f.setSize(305, 250);
    // Frame의 LayoutManager를 FlowLayout으로 설정한다.
    f.setLayout(new FlowLayout());

    Label q1 = new Label("1. 당신의 관심 분야는?(여러개 선택가능)");
    Checkbox news = new Checkbox("news", true); // 선택된 상태로 생성
    Checkbox sports = new Checkbox("sports");
    Checkbox movies = new Checkbox("movies");
    Checkbox music = new Checkbox("music");

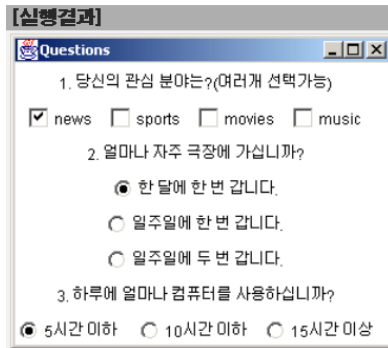
    f.add(q1); f.add(news); f.add(sports); f.add(movies); f.add(music);

    Label q2 = new Label("2. 얼마나 자주 극장에 가십니까?");
    CheckboxGroup group1 = new CheckboxGroup();
    Checkbox movie1 = new Checkbox("한 달에 한 번 갑니다.", group1, true);
    Checkbox movie2 = new Checkbox("일주일에 한 번 갑니다.", group1, false);
    Checkbox movie3 = new Checkbox("일주일에 두 번 갑니다.", group1, false);

    f.add(q2); f.add(movie1); f.add(movie2); f.add(movie3);

    Label q3 = new Label("3. 하루에 얼마나 컴퓨터를 사용하십니까?");
    CheckboxGroup group2 = new CheckboxGroup();
    Checkbox com1 = new Checkbox("5시간 이하", group2, true);
    Checkbox com2 = new Checkbox("10시간 이하", group2, false);
    Checkbox com3 = new Checkbox("15시간 이상", group2, false);

    f.add(q3); f.add(com1); f.add(com2); f.add(com3);
    f.setVisible(true);
}
```



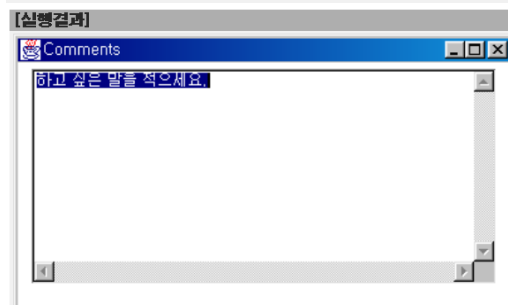
TextArea 추가하기 - 실습하기

```
import java.awt.*;

class TextAreaTest {
    public static void main(String args[]) {
        Frame f = new Frame("Comments");
        f.setSize(400, 220);
        f.setLayout(new FlowLayout());

        TextArea comments = new TextArea("하고 싶은 말을 적으세요.", 10, 50);

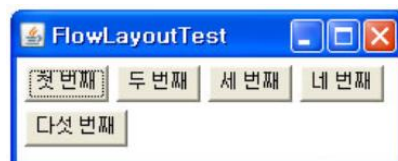
        f.add(comments);
        comments.selectAll(); // TextArea의 text 전체가 선택 되도록 한다.
        f.setVisible(true);
    }
}
```



레이아웃 (Layout)

컨테이너에 포함된 컴포넌트의 배치를 자동 관리한다.

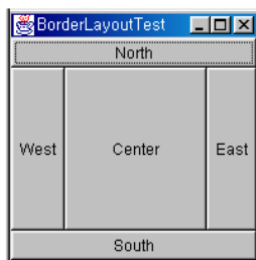
FlowLayout – 왼쪽부터 차례대로 컴포넌트를 배치하는 방식



실습하기

```
Frame f = new Frame("FlowLayoutTest");
f.setSize(250, 100);
f.setLayout(new FlowLayout(FlowLayout.LEFT));
f.add(new Button("첫 번째"));
f.add(new Button("두 번째"));
f.add(new Button("세 번째"));
f.add(new Button("네 번째"));
f.add(new Button("다섯 번째"));
f.setVisible(true);
```

BorderLayout – 동서남북 방향을 지정하여 배치하는 방식



실습하기

```
Frame f = new Frame("BorderLayoutTest");
f.setSize(200, 200);
// Frame은 기본적으로 BorderLayout로 설정되어 있으므로 따로 설정하지 않아도 됨
f.setLayout(new BorderLayout());
Button north = new Button("North");
Button south = new Button("South");
Button east = new Button("East");
Button west = new Button("West");
Button center = new Button("Center");

// Frame의 5개의 각 영역에 Button을 하나씩 추가한다.
f.add(north, "North"); // f.add("North", north);와 같이 쓸 수도 있다.
f.add(south, "South"); // South의 대소문자 정확히
f.add(east, "East"); // East대신, BorderLayout.EAST 사용가능
f.add(west, "West");
f.add(center, "Center");
```

GridLayout – 매트릭스 방식 (2 차원 배열)으로 배치하는 방식



실습하기

```
Frame f = new Frame("GridLayoutTest");
f.setSize(150, 150);
f.setLayout(new GridLayout(3, 2)); // 3행 2열의 테이블을 만든다.
f.add(new Button("1")); // 추가되는 순서대로 Button에 번호를 붙였다.
f.add(new Button("2"));
f.add(new Button("3"));
f.add(new Button("4"));
f.add(new Button("5"));
f.add(new Button("6"));
```

이벤트 (Event)

사용자 또는 프로그램에 의해 발생할 수 있는 하나의 사건.

종류	설명
이벤트 소스 (Event Source, 이벤트 발생지)	이벤트가 발생한 컴포넌트. 사용자가 Button을 눌렀을 때 이벤트가 발생하고, Button은 이 이벤트의 이벤트 소스가 된다.
이벤트 핸들러 (Event Handler, 이벤트 처리기)	이벤트가 발생했을 때 실행될 코드를 구현해 놓은 클래스
이벤트 리스너 (Event Listener, 이벤트 감지기)	이벤트를 감지하고 처리한다. 이벤트 핸들러를 이벤트 리스너로 이벤트 소스에 연결해야 이벤트가 발생했을 때 이벤트가 처리된다.

실습하기

```

class EventTest extends Frame {
    Button b = new Button("확 인");

    EventTest(String title) {
        super(title);
        setLayout(null);
        b.setBounds(100,75,100,50);
        // 이벤트 핸들러를 버튼의 이벤트 리스너를 등록한다.
        b.addActionListener(new EventHandler());

        add(b); // 버튼을 Frame에 추가한다.
        setSize(300, 200);
        setVisible(true);
    }

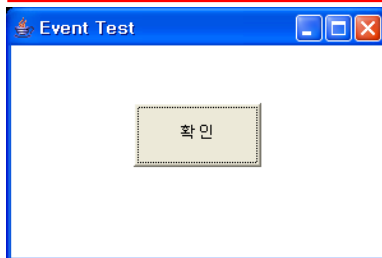
    public static void main(String args[]) {
        EventTest mainWin = new EventTest("Event Test");
    } // main
}

```

```

class EventHandler implements ActionListener { // 이벤트 핸들러
    public void actionPerformed(ActionEvent ae) {
        System.out.println("버튼이 눌러졌습니다.");
    }
}

```



애플릿 (Applet)

웹 브라우저를 통해 실행될 수 있는 작은 어플리케이션

1. 애플릿 관련정보가 포함된 HTML문서를 작성해야한다.
2. java.exe가 아닌 웹 브라우저를 통해 실행된다.
3. public static void main(String args[])이 필요하다.
4. 애플릿은 java.applet.Applet을 상속하는 public클래스이어야 한다.


```
import java.awt.*;

public class HelloApplet extends java.applet.Applet
{
    public void paint(Graphics g) {
        setBackground(Color.lightGray); // 애플릿의 바탕을 밝은 회색으로 설정
        g.drawString("Hello, Applet.", 50, 50);
    }
}
```

AppletTest.html

```
<html>
    <applet code="HelloApplet.class" width=200 height=100>
    </applet>
</html>
```



Swing

AWT 와 클래스 이름을 구분하기 위해 앞에 J 를 붙인다.

프레임 만들기 (실습하기)

```
import javax.swing.*;

class MyFrame extends JFrame {
    MyFrame() {
        setTitle("첫번째 프레임");
        setSize(300,300);
        setVisible(true);
    }
}

public class MyApp {
    public static void main(String [] args) {
        new MyFrame();
    }
}
```

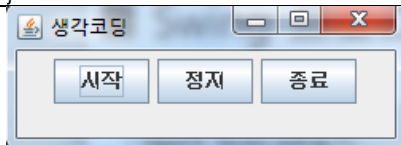
버튼 3 개를 만들고 FlowLayout 을 사용하여 배치하기 (실습하기)

```
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;

class FirstSwing {
    public static void main(String[] args) {
        JFrame frm = new JFrame("생각코딩");
        frm.setBounds(120, 120, 400, 100);
        frm.setLayout(new FlowLayout());

        JButton btn1 = new JButton("시작");
        JButton btn2 = new JButton("정지");
        JButton btn3 = new JButton("종료");

        frm.add(btn1);
        frm.add(btn2);
        frm.add(btn3);
        frm.setVisible(true);
    }
}
```



프레임 구성하는 예제 (실습하기)

```
import javax.swing.*;
import java.awt.*;

public class FrameSample extends JFrame{
    FrameSample() {
        setTitle("프레임 구성");

        // 메뉴 생성
        JMenuBar mb = new JMenuBar();
        JMenu fileMenu = new JMenu("File");
        JMenu editMenu = new JMenu("Edit");
        JMenu sourceMenu = new JMenu("Source");
        JMenu searchMenu = new JMenu("Search");
        JMenu windowMenu = new JMenu("Window");
        mb.add(fileMenu);
        mb.add(editMenu);
        mb.add(sourceMenu);
        mb.add(searchMenu);
        mb.add(windowMenu);
        setJMenuBar(mb);

        // contentPane
        this.getContentPane().setLayout(new FlowLayout());
        this.getContentPane().setBackground(new Color(186, 143, 45));
    }
}
```

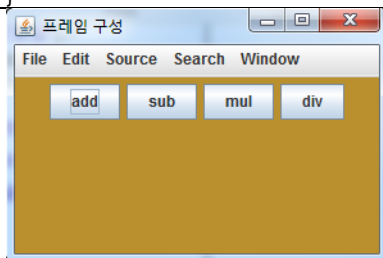
```

        this.add(new JButton("add"));
        this.add(new JButton("sub"));
        this.add(new JButton("mul"));
        this.add(new JButton("div"));
        setSize(300,200);
        setVisible(true);

        /*
        System.out.println(this.getSize());
        System.out.println(this.getJMenuBar().getSize());
        System.out.println(this.getContentPane().getSize());
        */
    }

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

```



BorderLayout – 개념은 AWT 와 동일하다 (실습하기)

```

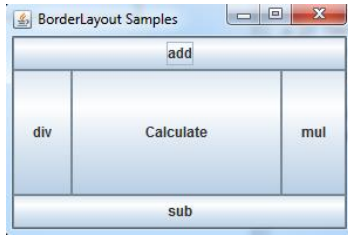
import javax.swing.*;
import java.awt.*;

public class BorderLayoutSampleEx extends JFrame {
    BorderLayoutSampleEx() {
        setTitle("BorderLayout Samples");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        add(new JButton("add"), BorderLayout.NORTH);
        add(new JButton("sub"), BorderLayout.SOUTH);
        add(new JButton("mul"), BorderLayout.EAST);
        add(new JButton("div"), BorderLayout.WEST);
        add(new JButton("Calculate"), BorderLayout.CENTER);

        setSize(300, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new BorderLayoutSampleEx();
    }
}

```



GridLayout

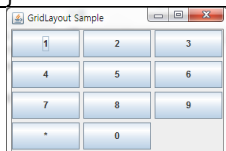
매트리스나 2 차원 배열과 같은 모양으로 디자인을 만들 때 사용. 계산기 프로그램이 대표적인 예가 될 수 있다. (실습하기)

```
import javax.swing.*;
import java.awt.*;

public class GridLayoutSample extends JFrame {
    GridLayoutSample() {
        setTitle("GridLayout Sample");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        GridLayout grid = new GridLayout(4, 3);
        grid.setVgap(5);
        grid.setHgap(5);
        setLayout(grid);
        for(int i=1; i<=9; i++)
            add(new JButton(Integer.toString(i)));

        add(new JButton("*"));
        add(new JButton("0"));

        setSize(300, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new GridLayoutSample();
    }
}
```



이벤트 처리

MouseListener - 마우스의 x, y 좌표의 움직임과 버튼의 동작에 따라 처리하도록 함. MouseListener 인터페이스를 implements 하여야 함. 즉, MouseListener 내에 있는 추상메서드를 자식클래스에서 구현해야 함.

실습하기

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
```

```

public class ListenerMouseEx extends JFrame {
    ListenerMouseEx() {
        setTitle("버튼에 Mouse 이벤트 리스너 작성");
        setLayout(new FlowLayout());
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // 버튼 컴포넌트를 생성하고 MouseListener 를 단다.
        JButton btn = new JButton("Mouse Event 테스트 버튼");
        btn.setBackground(Color.YELLOW);
        MyMouseListener listener = new MyMouseListener(); // 마우스 리스너를 생성한다.
        btn.addMouseListener(listener); // 마우스 리스너를 단다.

        // 버튼을 컨테이너에 단다
        add(btn);

        setSize(300,150);
        setVisible(true);
    }

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

```

// MouseListener 을 상속받아 Mouse 리스너를 작성한다.

// MouseListener 의 5 개 메소드를 모두 구현한다.

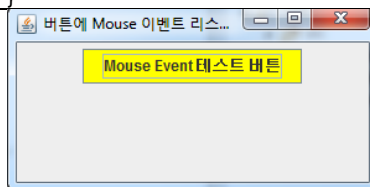
```

class MyMouseListener implements MouseListener {
    public void mouseEntered(MouseEvent e) { // 마우스가 버튼에 올라올때 호출
        JButton btn = (JButton)e.getSource(); //마우스가 올라간 버튼의 주소를 알아낸다.
        btn.setBackground(Color.RED); // 버튼의 배경색을 파란색으로 변경
    }

    public void mouseExited(MouseEvent e) { // 마우스가 버튼에서 내려갈때 호출
        JButton btn = (JButton)e.getSource(); //마우스가 올라간 버튼의 주소를 알아낸다.
        btn.setBackground(Color.YELLOW); // 버튼의 배경색을 노란색으로 변경
    }

    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseClicked(MouseEvent e) {}
}

```



실습하기

```

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

```

```

public class MouseListenerEx extends JFrame {
    JPanel contentPane = new JPanel(); // 컨텐트팬으로 사용할 패널
    JLabel la; // "hello" 문자열을 출력하기 위한 레이블 컴포넌트

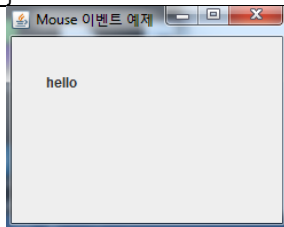
    MouseListenerEx() {
        setTitle("Mouse 이벤트 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setContentPane(contentPane); // 컨텐트팬 변경
        contentPane.setLayout(null); // 컨텐트팬의 배치관리자 삭제
        contentPane.addMouseListener(new MyMouseListener()); // 컨텐트팬에 Mouse 리스너 달기
        la = new JLabel("hello"); // "hello" 레이블 컴포넌트 생성
        la.setSize(50, 20); // 레이블의 크기 50x20 을 설정
        la.setLocation(30, 30); // 레이블의 위치 (30,30)으로 설정
        contentPane.add(la); //레이블 컴포넌트를 컨텐트팬을 추가
        setSize(200,200);
        setVisible(true);
    }

    // Mouse 리스너 구현
    class MyMouseListener implements MouseListener {
        public void mousePressed(MouseEvent e) {
            int x = e.getX(); // 마우스 클릭 좌표 x
            int y = e.getY(); // 마우스 클릭 좌표 y
            la.setLocation(x, y); // 레이블의 위치를 (x,y)로 이동
        }
        public void mouseReleased(MouseEvent e) {}
        public void mouseClicked(MouseEvent e) {}
        public void mouseEntered(MouseEvent e) {}
        public void mouseExited(MouseEvent e) {}
    }

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

```



KeyAdapter – 정상적인 class 이므로 extends 하여 필요한 method 를 오버라이딩하면 된다.

실습하기

```

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

```

```

public class KeyCodeEx extends JFrame {
    JPanel contentPane = new JPanel();// 콘텐츠팬으로 사용할 패널
    JLabel la = new JLabel();// 빈 레이블 컴포넌트 생성

    KeyCodeEx() {
        setTitle("Key Code 예제 : F1키:초록색, % 키 노란색");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setContentPane(contentPane);
        contentPane.addKeyListener(new MyKeyListener());
        contentPane.add(la);
        setSize(300,150);
        setVisible(true);

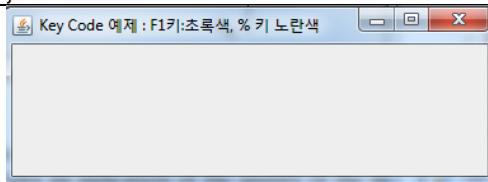
        // 콘텐츠팬이 키 입력을 받을 수 있도록 포커스 강제 지정
        contentPane.requestFocus();
    }

    // Key 리스너 구현
    class MyKeyListener extends KeyAdapter {
        public void keyPressed(KeyEvent e) {
            // la에 입력된 키의 키 이름 문자열을 출력하여 사용자에게 보고함
            la.setText(e.getKeyText(e.getKeyCode())+"키가 입력되었음");

            if(e.getKeyChar() == '%') // 입력된 키가 %인가
                contentPane.setBackground(Color.YELLOW);
            else if(e.getKeyCode() == KeyEvent.VK_F1) // 입력된 키가 F1 인가
                contentPane.setBackground(Color.GREEN);
        }
    }

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

```



Click 과 Double Click 이벤트 처리 – MouseAdapter 를 상속받아 mouseClicked 메서드를 오버라이딩 한다.

실습하기

```

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class ClickAndDoubleClickEx extends JFrame {

```

```

JPanel contentPane = new JPanel(); // 콘텐츠팬으로 사용할 패널

ClickAndDoubleClickEx() {
    setTitle("Click and DoubleClick 예제");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setContentPane(contentPane);
    contentPane.addMouseListener(new MyMouseListener()); // 콘텐츠팬에 Mouse 리스너 달기
    setSize(300,200);
    setVisible(true);
}

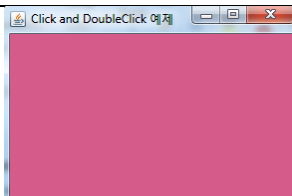
// MouseAdapter 를 상속받아 Mouse 리스너 구현
class MyMouseListener extends MouseAdapter {
    public void mouseClicked(MouseEvent e) {
        if(e.getClickCount() == 2) { // 더블클릭인가?
            // r,g,b 정수 값을 [0,255] 사이의 랜덤하게 생성
            int r = (int)(Math.random()*256);
            int g = (int)(Math.random()*256);
            int b = (int)(Math.random()*256);

            JPanel p = (JPanel)e.getSource(); //마우스가 클릭된 패널(컨텐츠팬)을
알아낸다.

            p.setBackground(new Color(r,b,g)); // 마우스가 클릭된 패널의 배경색으로
r,g,b 색으로 칠한다.
        }
    }
}

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

```



TextArea 에 텍스트 추가하기 (실습하기)

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class TextAreaEx extends JFrame {
    Container contentPane;
    TextAreaEx() {
        setTitle("텍스트영역 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane(); // 컨텐트팬의 주소를 알아낸다.

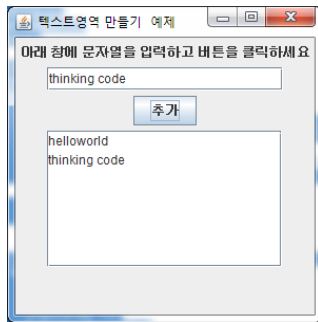
        // 컨텐트팬에 MyCenterPanel 을 부착한다.
        contentPane.add(new MyCenterPanel(), BorderLayout.CENTER);

        setSize(300,300);
        setVisible(true);
    }

    // 텍스트필드, 버튼, 텍스트영역 컴포넌트를 가지는 패널
    class MyCenterPanel extends JPanel {
        JTextField tf;
        JButton btn;
        JTextArea ta;
        MyCenterPanel() {
            tf = new JTextField(20);
            btn = new JButton("추가");

            // 버튼에 Action 리스너를 등록한다.
            btn.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    ta.append(tf.getText()+"\n");
                }
            });
            ta = new JTextArea("hello", 7, 20);
            add(new JLabel("아래 창에 문자열을 입력하고 버튼을 클릭하세요"));
            add(tf);
            add(btn);
            add(new JScrollPane(ta));
        }
    }

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



JSlider 클래스를 이용한 예제 (실습하기)

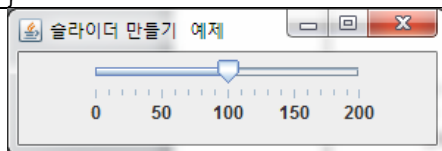
```
import javax.swing.*;
import java.awt.*;

public class SliderEx extends JFrame {
    Container contentPane;
    SliderEx() {
        setTitle("슬라이더 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

        // 슬라이더 컴포넌트 만들기
        JSlider slider = new JSlider(JSlider.HORIZONTAL, 0, 200, 100);
        slider.setPaintLabels(true); // 수치(label)가 보이도록 설정
        slider.setPaintTicks(true); // 눈금(tick)이 보이도록 설정
        slider.setPaintTrack(true); // 트랙(track)이 이 보이도록 설정
        slider.setMajorTickSpacing(50); // 큰 눈금 간격을 50 픽셀로 설정
        slider.setMinorTickSpacing(10); // 작은 눈금 간격을 10 픽셀로 설정

        contentPane.add(slider);
        setSize(300,100);
        setVisible(true);
    }

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



쓰레드 에서 Thread 클래스와 Timer 를 사용한 예제 (실습하기)

```
import java.awt.*;
import javax.swing.*;

class TimerThread extends Thread {
    JLabel timerLabel; // 타이머 값이 출력된 레이블

    public TimerThread(JLabel timerLabel) { // 생성자
        this.timerLabel = timerLabel; // 레이블 컴포넌트를 받아서 기억
    }

    // 스레드 코드.
    // run()이 종료하면 스레드 종료
    public void run() {
        int n=0; // 타이머 카운트 값
        while(true) { // 무한 루프
            timerLabel.setText(Integer.toString(n)); // 타이머 값을 레이블에 출력
            n++; // 카운트 증가
            try {
                Thread.sleep(1000); // 1 초동안 잠을 잔다.
            }
            catch (InterruptedException e) {
                return; // 예외가 발생하면 스레드 종료
            }
        }
    }
}

public class ThreadTimerEx extends JFrame {
    public ThreadTimerEx() {
        setTitle("ThreadTimerEx 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));

        // 타이머 스레드로 사용할 객체 생성.
        // 타이머 값을 출력할 레이블 컴포넌트를 생성자에 전달한다.
        TimerThread th = new TimerThread(timerLabel);
        c.add(timerLabel); // 레이블을 콘텐츠팬에 부착

        setSize(300,150);
        setVisible(true);
    }
}
```

```

        th.start(); // 타이머 스레드의 실행을 시작시킨다.
    }

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

```



쓰레드에서 Runnable 인터페이스와 Timer 를 사용한 예제 (실습하기)

```

import java.awt.*;
import javax.swing.*;

class TimerRunnable implements Runnable {
    JLabel timerLabel; // 타이머 값이 출력된 레이블

    public TimerRunnable(JLabel timerLabel) {
        this.timerLabel = timerLabel;
    }

    // 스레드 코드.
    // run()이 종료하면 스레드 종료
    public void run() {
        int n=0; // 타이머 카운트 값
        while(true) { // 무한 루프
            timerLabel.setText(Integer.toString(n)); // 타이머 값을 레이블에
출력

            n++; // 카운트 증가
            try {
                Thread.sleep(1000); // 1 초동안 잠을 잔다.
            }
            catch (InterruptedException e) {
                return; // 예외가 발생하면 스레드 종료
            }
        }
    }
}

public class RunnableTimerEx extends JFrame {
    public RunnableTimerEx() {
        setTitle("RunnableTimerEx 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));
    }
}

```

```

        // 타이머 스레드로 사용할 Runnable 객체 생성.
        // 타이머 값을 출력할 레이블 컴포넌트를 생성자에 전달한다.
        TimerRunnable runnable = new TimerRunnable(timerLabel);

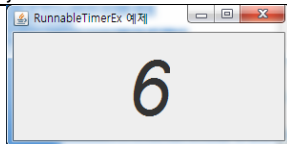
        Thread th = new Thread(runnable); // 스레드 객체 생성

        c.add(timerLabel); // 콘텐츠팬에 레이블 부착

        setSize(300,150);
        setVisible(true);

        th.start(); // 타이머 스레드의 실행을 시작시킨다.
    }
    public static void main(String[] args) {
        new RunnableTimerEx();
    }
}

```



메뉴를 만들어 주는 JMenu 클래스를 사용한 예제 (실습하기)

```

import javax.swing.*;

public class MenuEx extends JFrame {
    MenuEx() {
        setTitle("Menu 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        createMenu(); // 메뉴를 생성하여 프레임에 삽입
        setSize(250,200);
        setVisible(true);
    }

    // 메뉴를 만들어 프레임에 삽입한다.
    void createMenu() {
        JMenuBar mb = new JMenuBar(); // 메뉴바 생성
        JMenu fileMenu = new JMenu("File"); // File 메뉴 생성

        // File 메뉴에 메뉴아이템 생성 삽입
        fileMenu.add(new JMenuItem("New"));
        fileMenu.add(new JMenuItem("Open"));
        fileMenu.addSeparator(); // 분리선 삽입
        fileMenu.add(new JMenuItem("Save"));
        fileMenu.add(new JMenuItem("SaveAs"));

        // 메뉴바에 메뉴 삽입
        mb.add(fileMenu); // File 메뉴 삽입
    }
}

```

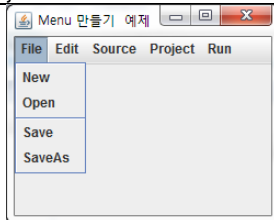
```

        mb.add(new JMenu("Edit")); // Edit 메뉴 생성 삽입
        mb.add(new JMenu("Source")); // Source 메뉴 생성 삽입
        mb.add(new JMenu("Project")); // Project 메뉴 생성 삽입
        mb.add(new JMenu("Run")); // Run 메뉴 생성 삽입

        // 메뉴바를 프레임에 삽입
        setJMenuBar(mb);
    }

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

```



Menu Action Event 실습하기

```

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class MenuActionEventEx extends JFrame {
    Container contentPane;
    JLabel label = new JLabel("Hello");

    MenuActionEventEx() {
        setTitle("Menu 에 Action 리스너 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        label.setHorizontalAlignment(SwingConstants.CENTER);
        contentPane.add(label, BorderLayout.CENTER);
        createMenu();
        setSize(250,200);
        setVisible(true);
    }

    // 메뉴바와 Text 메뉴를 생성하고 Text 메뉴에 4 개의 메뉴아이템을 삽입한다.
    void createMenu() {
        JMenuBar mb = new JMenuBar(); // 메뉴바 생성
        JMenuItem [] menuItem = new JMenuItem [4];
        String[] itemTitle = {"Color", "Font", "Top", "Bottom"};
        JMenu textMenu = new JMenu("Text");

        // 4 개의 메뉴아이템을 Text 메뉴에 삽입한다.
        for(int i=0; i<menuItem.length; i++) {
            menuItem[i] = new JMenuItem(itemTitle[i]); // 메뉴아이템 생성
        }
        textMenu.add(menuItem[0]);
        textMenu.add(menuItem[1]);
        textMenu.add(menuItem[2]);
        textMenu.add(menuItem[3]);
        mb.add(textMenu);
    }
}

```

등록

```
menuItem[i].addActionListener(new MenuActionListener()); //메뉴아이템에 Action 리스너

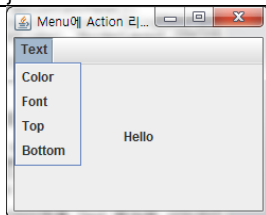
textMenu.add(menuItem[i]); // 메뉴아이템을 Text 메뉴에 삽입
}
mb.add(textMenu); // 메뉴바에 Text 메뉴 삽입
setJMenuBar(mb); // 프레임에 메뉴바를 삽입한다.
}

// Action 리스너로서, 메뉴아이템이 선택되었을 때 처리한다.
class MenuActionListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        if(cmd.equals("Color")) // Color 메뉴아이템 선택된 경우
            label.setForeground(Color.BLUE);
        else if(cmd.equals("Font")) // Font 메뉴아이템 선택된 경우
            label.setFont(new Font("Ravie", Font.ITALIC, 30));
        else if(cmd.equals("Top")) // Top 메뉴아이템 선택된 경우
            label.setVerticalAlignment(SwingConstants.TOP);

        else // Bottom 메뉴아이템 선택된 경우
            label.setVerticalAlignment(SwingConstants.BOTTOM);

    }
}

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



Dialog 박스 실습하기

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class MyModalDialog extends JDialog {
    JTextField tf = new JTextField(10);
    JButton okButton = new JButton("OK");

    public MyModalDialog(JFrame frame, String title) {
        super(frame,title, true);
        setLayout(new FlowLayout());
        add(tf);
        add(okButton);
        setSize(200, 100);
    }
}
```

```

        okButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                setVisible(false);
            }
        });
    }

    // 텍스트필드 창에 사용자가 입력한 문자열을 리턴한다.
    // 입력된 내용이 없으면 null 을 리턴한다.
    String getInput() {
        if(tf.getText().length() == 0) return null;
        else return tf.getText();
    }
}

public class DialogEx2 extends JFrame{
    MyModalDialog dialog; // 다이얼로그의 레퍼런스

    public DialogEx2() {
        super("DialogEx2 예제 프레임");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JButton btn = new JButton("Show Modal Dialog");

        // 모달 다이얼로그 생성
        dialog = new MyModalDialog(this, "Test Modal Dialog");

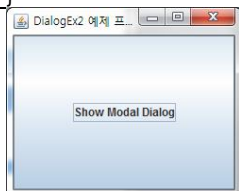
        btn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dialog.setVisible(true); // 모달 다이얼로그 작동 시작

                // 다이얼로그로부터 사용자가 입력한 문자열을 받아온다.
                String text = dialog.getInput();

                if(text == null) return; // 입력한 문자열이 없는 경우
                JButton btn = (JButton)e.getSource();
                btn.setText(text); // 입력한 문자열로 이 버튼의 문자열을
                변경한다.
            }
        });
        getContentPane().add(btn);
        setSize(250,200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new DialogEx2();
    }
}

```



Tab 생성하여 Label 에 image 추가하기 (실습하기)

image 는 프로젝트 폴더에 copy 해 놓아야 합니다.

```
import javax.swing.*;
import java.awt.*;

public class TabbedPaneEx extends JFrame {
    Container contentPane;
    TabbedPaneEx() {
        setTitle("탭팬 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        JTabbedPane pane = createTabbedPane(); // 탭팬을 생성한다.
        contentPane.add(pane, BorderLayout.CENTER); // 탭팬을 콘텐츠팬에 부착한다.
        setSize(250,250);
        setVisible(true);
    }

    // 탭팬을 생성하고 3 개의 탭을 생성하여 부착한다.
    JTabbedPane createTabbedPane() {
        JTabbedPane pane = new JTabbedPane(); // 탭팬 객체를 생성한다.
        pane.addTab("tab1", new JLabel(new ImageIcon("images/img1.jpg"))); // 첫번째 탭
        pane.addTab("tab2", new JLabel(new ImageIcon("images/img2.jpg"))); // 두번째 탭
        pane.addTab("tab3", new MyPanel()); // 세번째 탭
        return pane;
    }

    // 3 번째 탭의 컴포넌트로 사용될 패널
    class MyPanel extends JPanel {
        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.fillOval(10,70,50,50);
            g.setColor(Color.BLACK);
            g.drawString("tab 3 에 들어가는 JPanel 입니다. ", 30, 50);
        }
    }

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



MouseMotionListener 를 이용하여 재미있는 코드 작성 (실습하기)

```
import java.awt.*;
import java.awt.event.*;

class GraphicsEx4 extends Frame implements MouseMotionListener {
    int x = 0;
    int y = 0;

    Image img = null;
    Graphics glmg = null;

    public static void main (String[] args){
        new GraphicsEx4("GraphicsEx4");
    }

    public GraphicsEx4(String title) {
        super(title);
        addMouseListener(this);
        addWindowListener( new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });

        setBounds(100, 100, 500, 500);
        setVisible(true);

        img = createImage(500, 500);
        glmg = img.getGraphics();
        glmg.drawString("왼쪽버튼을 누른 채로 마우스를 움직여보세요.", 10, 50);
        repaint();
    }

    public void paint(Graphics g) {
        if(img != null)
            g.drawImage(img, 0, 0, this);
    }

    public void mouseMoved(MouseEvent me) {}

    public void mouseDragged(MouseEvent me) {
        if (me.getModifiersEx() == MouseEvent.BUTTON1_DOWN_MASK){
            x = me.getX();
            y = me.getY();
            glmg.drawString("x: " + x, x, y);
            repaint();
        }
    }
}
```

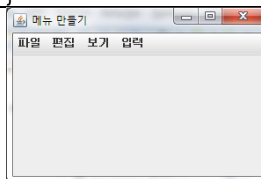


메뉴 만들기 실습

```
import javax.swing.*;

public class SimpleMenuFrame extends JFrame {
    public SimpleMenuFrame() {
        super("메뉴 만들기");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        createMenu();
        setSize(300,200);
        setVisible(true);
    }
    private void createMenu() {
        JMenu fileMenu = new JMenu("파일");
        JMenu editMenu = new JMenu("편집");
        JMenu viewMenu = new JMenu("보기");
        viewMenu.add(new JMenuItem("화면확대"));
        viewMenu.add(new JMenuItem("쪽윤곽"));
        JMenu inputMenu = new JMenu("입력");
        JMenuBar mb = new JMenuBar();
        mb.add(fileMenu);
        mb.add(editMenu);
        mb.add(viewMenu);
        mb.add(inputMenu);
        setJMenuBar(mb);
    }
    static public void main(String[] arg) {
        new SimpleMenuFrame();
    }
}
```



다얼로그 박스 만들기

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;

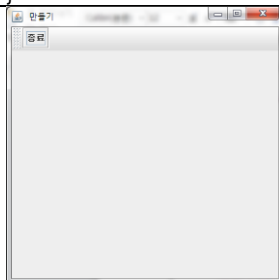
public class ToolBarAndOptionPaneFrame extends JFrame {
    public ToolBarAndOptionPaneFrame() {
        super(" 만들기");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JToolBar tBar = new JToolBar();
        JButton btn = new JButton("종료");
        btn.addActionListener(new CloseActionListener());
        tBar.add(btn);
        add(tBar, BorderLayout.NORTH);

        setSize(400,400);
        setVisible(true);
    }

    class CloseActionListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            int result = JOptionPane.showConfirmDialog(null, "정말 종료하시겠습니까", "옵션판",
                JOptionPane.YES_NO_OPTION);
            if(result == JOptionPane.CLOSED_OPTION) {
                // 사용자가 "예"나 "아니오"의 선택없이 다이얼로그창을 닫은 경우
                return; // not exited
            }
            else if(result == JOptionPane.YES_OPTION) {
                // 사용자가 "예"를 선택한 경우
                System.exit(0);
            }
            else { // 사용자가 "아니오"를 선택한 경우
                return; // not exited
            }
        }
    }

    static public void main(String[] arg) {
        new ToolBarAndOptionPaneFrame();
    }
}
```



수고했습니다. GUI는 내용이 방대하다 보니 아직 다루지 못한 클래스가 많지만 주로 많이 사용되는 것 위주로 실습을 해 보았습니다. GUI는 주로 다른 기능과 접목해서 사용하므로 사용자들이게 친숙하게 프로그램을 접근할 수 있도록 하는 중요한 역할을 합니다. 스마트폰 같은 경우 대부분이 GUI를 통해 App을 사용하고 있습니다.

THINKING CODING 