

□ GUI

○ GUI(Graphic User Interface) : 사용자가 그래픽을 이용하여 코드를 짜는 것

* AWT(Abstract Windowing Toolkit) : GUI라이브러리, 어떤 내용들을 덮어놨음

* Swing : 모든 AWT 기능을 가지고 있으며, 기능이 더 추가된 자바 라이브러리

- JComponent 상속받는 클래스 : Swing에서 제일 중요한 추상적 클래스(최상위)

- Container에서 상속받는 몇(3) 개의 클래스 : GUI오브젝트를 담은 용기

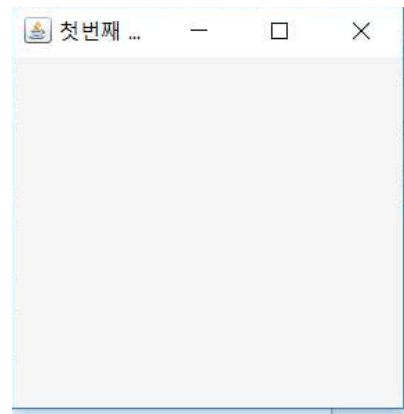
* JFrame : 창을 띄우기 위해 상속받는 클래스

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

public class Swing1 extends JFrame{
    //JFrame을 상속

    Swing1(){ //생성자를 만들
        setTitle("첫번째 프레임");
        setSize(300,300);
        setVisible(true);
    }

    public static void main(String[] args){
        Swing1 sw = new Swing1();
    }
}
```



창 생성

- 타이틀 : frame.setTitle("타이틀 이름");

- 컨테네판 : 컴포넌트를 붙일 수 있는 공간(컨테이너) // Container contentPane = frame.getContentPane();

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

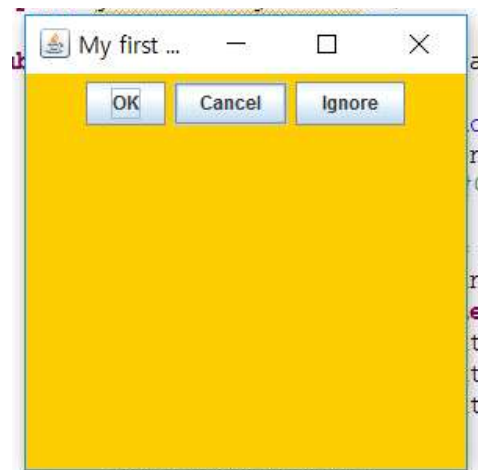
public class Swing2 extends JFrame{ //JFrame을 상속
    Swing2(){ //생성자를 만들
        setTitle("My first window");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //메인창의 X버튼 누르면 자바도 같이 종료

        Container contentPane = getContentPane();
        contentPane.setBackground(Color.ORANGE);
        contentPane.setLayout(new FlowLayout());
        contentPane.add(new JButton("OK"));
        contentPane.add(new JButton("Cancel"));
        contentPane.add(new JButton("Ignore"));

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

    public static void main(String[] args){
        new Swing2();
    }
}
```



○ 이벤트 기반 프로그래밍(Event Driven Programming) (콜백)

- * 이벤트오브젝트 : 모든 이벤트의 조상
- * 이벤트 소스 : 어떤 것이 이벤트를 발생 하였느냐에 따라 달라지는 것

○ 이벤트 리스너 : 이벤트를 취급하는 코드 / 클래스로 작성

- Component.addXXXListener(listener) // xxx:이벤트명 / listener : 이벤트 리스너 객체

○ ActionListener(액션 리스너)

- Class MyActionListener implements ActionListener { // 액션 리스너를 상속받은 클래스 생성
Public void actionPerformed(ActionEvent e) { } // 액션 리스너를 가능하게 해줌

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

public class ButtonAction extends JFrame {

    ButtonAction(){
        setTitle("My first window");
        setLayout(new FlowLayout());
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

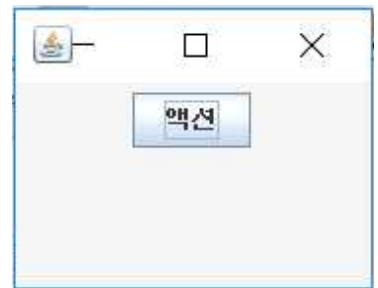
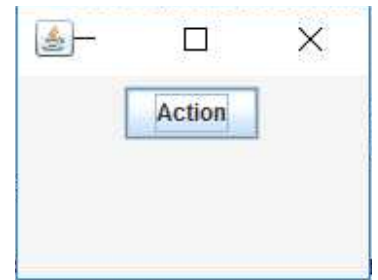
        JButton btn = new JButton("Action");
        MyActionListener listener = new MyActionListener();
        btn.addActionListener(listener);
        add(btn);

        setSize(512,360);
        setVisible(true);
    }

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

class MyActionListener implements ActionListener {
    public void actionPerformed(ActionEvent e){
        JButton b = (JButton)e.getSource();
        if(b.getText().equals("Action"))
            b.setText("액션");
        else
            b.setText("Action");
    }
}
```

// 버튼 눌렀을 경우 이벤트 //



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

public class MouseButtonAction extends JFrame {

    MouseButtonAction() {
        setTitle("Mouse리스너");
        setLayout(new FlowLayout());
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JButton btn = new JButton("Action");
        btn.setBackground(Color.YELLOW);
        MyMouseListener listener = new MyMouseListener();
        btn.addMouseListener(listener);
        add(btn);
        setSize(300,150);
        setVisible(true);
    }

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

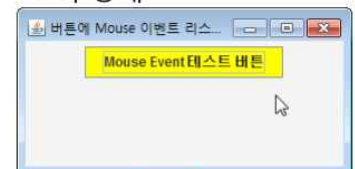
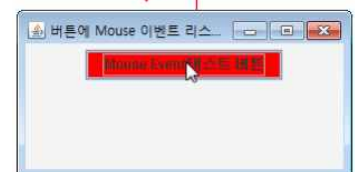
    class MyMouseListener implements MouseListener {
        public void mouseEntered(MouseEvent e) {
            JButton btn = (JButton)e.getSource();
            btn.setBackground(Color.RED);
        }

        public void mouseExited(MouseEvent e) {
            JButton btn = (JButton)e.getSource();
            btn.setBackground(Color.YELLOW);
        }

        public void mousePressed(MouseEvent e) {
            JButton b = (JButton)e.getSource();
            if(b.getText().equals("Action"))
                b.setText("액션");
            else
                b.setText("Action");
        }

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

초기 상태

마우스가
버튼에
올라갈
때마우스가
버튼에서
내려올 때

○ 어댑터 클래스 : 액션 리스너들의 어댑터 역할을 해줌

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

public class MouseAdaptorExample extends JFrame{
    JLabel label; // 클래스 멤버가 됨 (전역 label이 됨)
    JLabel lCode, lChar, lString;

    MouseAdaptorExample(){
        setTitle("Mouse Adaptor"); // 타이틀 제목
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //프로그램 종료시 메인도 닫아줌

        label = new JLabel("Hello"); // Hello라는 이름의 레이블 생성
        label.setSize(50,20); // setSize로 해서 label의 크기를 잡아줌
        label.setLocation(30,30); // label의 좌표

        lCode = new JLabel("Code");
        lChar = new JLabel("Char");
        lString = new JLabel("String");
        lCode.setSize(50,20);
        lCode.setLocation(100,30);
        lChar.setSize(50,20);
        lChar.setLocation(150,30);
        lString.setSize(50,20);
        lString.setLocation(200,30);

        JPanel panel = new JPanel(); //자신의 패널을 만들
        panel.setLayout(null); // JFrame에서 쓰는 레이아웃을 사용하지 않겠다
        setContentPane(panel); // 내용을 담는 판넬 (컨테이너)
        panel.add(label); // 패널에 label을 추가함
        panel.add(lCode);
        panel.add(lChar);
        panel.add(lString);

        panel.addMouseListener(new MyMouseListener2()); //마우스리스너를 추가
        panel.addKeyListener(new MyKeyListener());

        setSize(512,512); //화면 사이즈
        setVisible(true); // 위 작성한 것들을 Visible(실체화)함
        panel.requestFocus(); //키를 받으려면 포커스를 받아야 함(Visible 후에 리퀘스트해야됨)
    }

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

    private class MyKeyListener extends KeyAdapter{
        public void keyPressed(KeyEvent e){
            int code = e.getKeyCode();
            char c = e.getKeyChar();
            String text = e.getKeyText(code);
            lCode.setText(Integer.toString(code));
            lChar.setText(Character.toString(c));
            lString.setText(text);
        }
    }

    private class MyMouseListener2 extends MouseAdapter {
        //extends는 추상이 아닌 클래스를 상속
        public void mousePressed(MouseEvent e) { // 마우스 클릭 시 이벤트
            int x = e.getX(); // 마우스 위치 x값
            int y = e.getY(); // 마우스 위치 y값
            label.setLocation(x, y); //x.y의 위치를 잡아줌
        }
        public void mouseReleased(MouseEvent e){
            label.setLocation(30,30);
        }
    }
}
```

```
// 방향키로 위치 조절 //
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class FlyingTextEx extends JFrame {
    JPanel contentPane = new JPanel();
    JLabel la = new JLabel("HELLO");
    final int FLYING_UNIT = 10;

    FlyingTextEx() {
        setTitle("상,하,좌,우 키를 이용하여 텍스트 움직이기");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setContentPane(contentPane);

        contentPane.setLayout(null);
        contentPane.addKeyListener(new MyKeyListener());

        la.setLocation(50, 50);
        la.setSize(100, 20);
        contentPane.add(la);
        setSize(300, 300);
        setVisible(true);
        contentPane.requestFocus();
    }

    class MyKeyListener extends KeyAdapter {
        public void keyPressed(KeyEvent e) {
            int keyCode = e.getKeyCode();
            switch (keyCode) {
                case KeyEvent.VK_UP:
                    la.setLocation(la.getX(), la.getY() - FLYING_UNIT);
                    break;
                case KeyEvent.VK_DOWN:
                    la.setLocation(la.getX(), la.getY() + FLYING_UNIT);
                    break;
                case KeyEvent.VK_LEFT:
                    la.setLocation(la.getX() - FLYING_UNIT, la.getY());
                    break;
                case KeyEvent.VK_RIGHT:
                    la.setLocation(la.getX() + FLYING_UNIT, la.getY());
                    break;
            }
        }
    }

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

- 익명 클래스 : 메소드가 한 두 개인 클래스일 때 사용 / 지저분해질 수 있다.

이벤트 객체와 이벤트 소스

이벤트 객체	이벤트 소스	이벤트가 발생하는 경우
ActionEvent	JButton	마우스로 버튼을 클릭하거나 키로 버튼을 선택한 경우
	JList	리스트 아이템을 더블클릭하여 리스트 아이템을 선택한 경우
	JMenuItem	메뉴 아이템 선택을 선택한 경우
	JTextField	텍스트 입력 중 <Enter> 키를 누른 경우
ItemEvent	JCheckBox	체크박스의 선택 혹은 해제
	JCheckBoxMenuItem	체크박스 메뉴 아이템이 선택 혹은 해제 될 때
	JList	리스트 아이템이 선택될 때
KeyEvent	Component	모든 컴포넌트에 대해, 키가 눌러지거나 눌린 키가 떼어질 때
MouseEvent	Component	모든 컴포넌트에 대해, 마우스 버튼이 눌러지거나 떼어질 때, 클릭될 때, 컴포넌트 위에 마우스가 올라갈 때, 올라간 마우스가 내려올 때, 마우스가 드래그될 때, 마우스가 단순 움직일 때
FocusEvent	Component	모든 컴포넌트에 대해, 컴포넌트가 포커스를 받거나 잃을 때
TextEvent	TextField	텍스트가 변경될 때
	TextArea	텍스트가 변경될 때
WindowEvent	Window	Window를 상속받는 모든 컴포넌트에 대해, 윈도우가 활성화, 비활성화, 아이콘화, 아이 콘에서 복구 될 때, 윈도우가 열리거나 닫힐 때, 윈도우가 종료될 때 등
AdjustmentEvent	JScrollBar	스크롤바를 사용자가 움직였을 때
ComponentEvent	Component	모든 컴포넌트에 대해, 컴포넌트가 사라지거나, 나타나거나, 이동 하거나 크기 변경 될 때
ContainerEvent	Container	Container에 컴포넌트가 추가 혹은 삭제되었을 때

○ JLabel(레이블 컴포넌트) : 화면(판넬)에 정보를 표시를 하는 것

- 텍스트 : JLabel 라벨이름 = new JLabel("텍스트");
- 이미지 : ImageIcon 이미지아이콘이름 = new ImageIcon("images/beaty.jpg");
JLabel 이미지라벨 = new JLabel(이미지아이콘이름);

○ JButton(버튼 컴포넌트) : 화면(판넬)에 버튼을 생성하는 것

- 텍스트와 이미지 : JButton(String text, Icon icon)
- JButton btn = new JButton("hello");

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

public class ImageButton extends JFrame{
    ImageButton(){
        setTitle("ImageButton Test");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        ImageIcon Naver = new ImageIcon("./images/naver.jpg");
        ImageIcon Naver2 = new ImageIcon("./images/naver2.jpg");
        ImageIcon Naver3 = new ImageIcon("./images/naver3.jpg");

        JButton btn = new JButton(Naver);
        //기본이미지
        btn.setRolloverIcon(Naver2);
        //마우스를 가져다댔을 때 이미지
        btn.setPressedIcon(Naver3);
        //클릭했을 때 이미지
        c.add(btn);
        setSize(512,512);
        setVisible(true);
    }

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

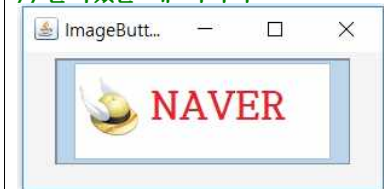
JButton btn = new JButton(Naver
//기본이미지



btn.setRolloverIcon(Naver2);
//마우스를 가져다댔을 때 이미지



btn.setPressedIcon(Naver3);
//클릭했을 때 이미지



○ Alignment(레이블과 버튼의 정렬)

- 수평 정렬 : void setHorizontalAlignment(int align)
- 수직 정렬 : void setVerticalAlignment(int align)

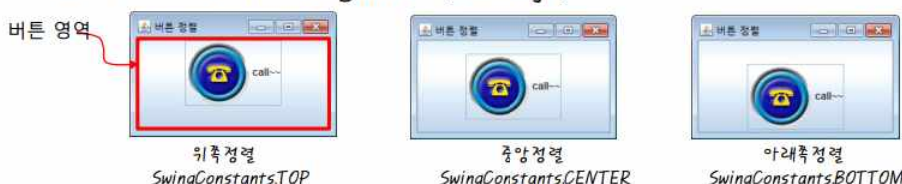
□ 수평 정렬

- 컴포넌트 영역 내에 이미지와 텍스트의 수평상의 위치 결정
- void setHorizontalAlignment(int align)



□ 수직 정렬

- 컴포넌트 영역 내에 콘텐츠(이미지와 텍스트)의 수직상의 위치
- void setVerticalAlignment(int align)



○ JCheckBox(체크박스 컴포넌트) : 화면(판넬)에 체크박스를 생성하는 것

- JCheckBox 체크박스이름 = new JCheckBox("사과");

* ItemListener : 체크했을 경우 나오는 이벤트

- class MyItemListener implements ItemListener

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

public class ItemEvent extends JFrame{
    Container c;
    JLabel sumlabel, ad;

    JCheckBox apple;
    JCheckBox pear;
    JCheckBox grape;

    int sum =0;

    ItemEvent(){
        setTitle("CheckBox Test");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        c = getContentPane();
        c.setLayout(new FlowLayout());

        ad = new JLabel("사과 3000원, 배 9000원, 포도 90000원");
        ad.setSize(50,30);
        c.add(ad);

        apple = new JCheckBox("사과");
        pear = new JCheckBox("배");
        grape = new JCheckBox("포도");

        apple.addItemListener(new MyItemListener());
        pear.addItemListener(new MyItemListener());
        grape.addItemListener(new MyItemListener());

        sumlabel = new JLabel("현재"+sum+"원 입니다.");

        c.add(sumlabel);
        c.add(apple);
        c.add(pear);
        c.add(grape);

        setSize(512,512);
        setVisible(true);
    }

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

    class MyItemListener implements ItemListener {
        public void itemStateChanged(ItemEvent e){
            if(e.getStateChange() == ItemEvent.SELECTED){
                if(e.getItem()==apple) sum+=3000;
                else if(e.getItem()==pear) sum+=9000;
                else if(e.getItem()==grape) sum+=90000;
            }
            else {
                if(e.getItem()==apple) sum-=3000;
                else if(e.getItem()==pear) sum-=9000;
                else if(e.getItem()==grape) sum-=90000;
            }
        }
    }
}
```

○ **JRadioButton(라디오버튼)** : 여러 버튼으로 그룹을 형성하고, 그 중 하나만 선택 상태가 되는 것

1. 버튼 그룹 객체 생성	→	ButtonGroup group = new ButtonGroup();
2. 라디오버튼 컴포넌트 생성	→	JRadioButton apple= new JRadioButton("사과"); JRadioButton pear= new JRadioButton("배"); JRadioButton cherry= new JRadioButton("체리");
3. 라디오 버튼을 버튼 그룹에 삽입	→	group.add(apple); group.add(pear); group.add(cherry);
4. 라디오 버튼을 컨테이너에 삽입	→	container.add(apple); container.add(pear); container.add(cherry);

○ **TextField(텍스트필드 컴포넌트)** : 한 줄짜리 텍스트 입력 창을 구현한 컴포넌트

- `contentPane.add(new JTextField(10));` // 10글자
- 텍스트 편집 불가 : `JTextField.setEditable(false);`
- 강제 문자열 출력 : `JTextField.setText("hello");`
- 텍스트 폰트 지정 : `JTextField.setFont(new Font("고딕체",Font.ITALIC, 20));`
- 문자열 선택하기 : `JTextField.select(0,5);` // 0번 문자에서 5번째까지 문자열 선택

○ **TextArea(텍스트영역 컴포넌트)** : 여러 줄을 입력할 수 있는 텍스트 입력 창(스크롤바 지원x)

- `JScrollPane` 객체에 삽입하는 방식으로 스크롤바 지원
-

○ **JList(리스트 컴포넌트)** : 항목을 리스트로 정렬하는 것(선택가능)

○ **JComboBox(콤보박스 컴포넌트)** :

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

public class Swing5 extends JFrame{
    Container c;
    String[] fruits = {"apple", "banana", "kiwi", "mango"};
    ImageIcon[] images = {
        new ImageIcon("images/apple.jpg"),
        new ImageIcon("images/banana.jpg"),
        new ImageIcon("images/kiwi.jpg"),
        new ImageIcon("images/mango.jpg")};
    JLabel imgLabel = new JLabel(images[0]);

    Swing5(){
        setTitle("ComboBox Test");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        c = getContentPane();
        c.setLayout(new FlowLayout());

        JComboBox strCombo = new JComboBox(fruits);
        strCombo.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                JComboBox cb = (JComboBox)e.getSource();
                int index = cb.getSelectedIndex();
                imgLabel.setIcon(images[index]);
            }
        });
        c.add(strCombo);
        c.add(imgLabel);

        setSize(600,400);
        setVisible(true);
    }

    public static void main(String[] arg){
        new Swing5();
    }
}
```

// Action 이벤트를 이용한 콤보 박스 활용 예
 콤보박스를 만들고 콤보박스를 선택한 애를 이미지 출력
 코드말에 약간의 문답형 문제 / 많으면 4문제
 오픈북 다음주 10시
 1공 309호

```
import java.*; //카드 뒤집기 게임//
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.util.Random;

public class CardGame2 extends JFrame {
    int save = 0, sv = 0, top = 0;
    Container contentPane;
    ImageIcon cur = new ImageIcon("./images/abc.jpg");
    ImageIcon clear = new ImageIcon("./images/clear.jpg");
    ImageIcon card[] = new ImageIcon[12];

    ImageIcon[] ima = { new ImageIcon("./images/a.jpg"), new ImageIcon("./images/b.jpg"),
    ImageIcon("./images/c.jpg"), new ImageIcon("./images/d.jpg"), new
    ImageIcon("./images/e.jpg"), new ImageIcon("./images/f.jpg") };

    JButton[] btn = new JButton[12]; // 버튼 12개 생성
    JLabel[] label = new JLabel[12]; // 라벨 12개 생성
    CardGame2() {
        setTitle("CardGame");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());
        for (int i = 0; i < 12; i++)
            btn[i] = new JButton(cur); // 뒷면 카드 생성
        //////////////////////////////////////
        int randcount = 0;
        int[] rand = new int[12];
        for (int i = 0; i < 12; i++) {
            rand[i] = (int) (Math.random() * 6); rand[i] = (int) (Math.random() * 6);
            for (int j = 0; j < i; j++) {
                if (rand[i] == rand[j])
                    randcount++;
                if (randcount == 2) {
                    rand[i] = (int) (Math.random() * 6);
                    i--;
                }
            }
            card[i] = ima[rand[i];
            randcount = 0;
        }
        // 카드 앞면 생성
        for (int i = 0; i < 12; i++) { // 카드 놀렸을 경우
            btn[i].addActionListener(new MyActionListener());
            contentPane.add(btn[i]);
        }
        setSize(600, 500);
        setVisible(true);
    }

    public static void main(String[] arg) {
        new CardGame2();
    }

    private class MyActionListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            JButton b = (JButton) e.getSource();
            save++;
            if (b.getIcon().equals(cur)) {
                for (int i = 0; i < 12; i++) {
                    if (b == btn[i]) {
                        b.setIcon(card[i]);
                        if (save == 2) {
                            if (b.getIcon().equals(btn[sv].getIcon())) { //
                                b.setIcon(clear);
                                btn[sv].setIcon(clear);
                                top++;
                                if (top == 6) {
                                    JOptionPane.showMessageDialog(null, "축하합니다.", "Message", JOptionPane.INFORMATION_MESSAGE);
                                } else { // 두장이 같은 카드가 아닐 때
                                    b.setIcon(cur);
                                    btn[sv].setIcon(cur);
                                    save = 0;
                                }
                                sv = i;
                            }
                        }
                    }
                }
            }
        }
    }
}
```


arg 창 띄우기

