

9차시 정리

```
// 자바의 GUI

// GUI의 목적
// 그래픽 이용, 사용자에게 이해하기 쉬운 모양으로 정보 제공
// 사용자는 마우스나 키보드를 통해서 쉽게 입력

// 자바 GUI의 특징
// 강력한 GUI 컴포넌트 제공, 쉬운 GUI 프로그래밍

// 자바의 GUI 프로그래밍 방법
// java.awt 패키지의 AWT 클래스와 javax.swing 패키지의 Swing 클래스 사용

// AWT(Abstract Windowing Toolkit)
// 자바가 처음 나왔을 때 함께 배포된 GUI 라이브러리
// java.awt 패키지에서 사용
// native 운영체제의 GUI 컴포넌트의 도움을 받아 작동을 하며 운영체제에 많은 부담을 준다.
// 따라서 중량컴포넌트라고 하며, 처리속도는 빠르다는 장점을 가지고 있다.

// Swing(스윙)
// AWT 기술을 기반으로 순수 자바 언어로 만들어진 라이브러리
// AWT 컴포넌트에 J자로 덧붙여진 이름의 클래스로 J로 시작하는 클래스가 해당한다.
// javax.swing 패키지에서 사용
// native 운영체제에 의존하지 않으며 경량 컴포넌트라고 불린다.

// Swing 컴포넌트는 2가지 유형으로 나뉜다.
// JComponent를 상속받는 클래스(대부분이다.)
// AWT의 Container를 상속받는 몇개의 클래스(JApplet, JFrame, JDialog )

// Swing 컴포넌트의 예시
// JButton, JCheckBox, JRadioButton, JSlider, JPasswordField
// JSpinner, JTextArea, JComboBox, JList, JProgressBar, JToolTip
// JScrollPane, JColorChooser, JFrame, JDialog, JMenu, JTable, JTree
// JToolBar, JTabbedPane, JSplitPane

// GUI 라이브러리 계층 구조
// Object -> Component -> JComponent, JApplet, JFrame, JDialog
// Swing클래스는 JComponent를 상속받아서 사용한다.
// JComponent : 스윙 컴포넌트의 공통적인 속성을 구현한 추상 클래스
// AWT의 Component를 상속받음.

// 컨테이너
// 다른 GUI 컴포넌트를 포함할 수 있는 컴포넌트로 다른 컨테이너에 포함될 수 있음
// java.awt.Container 상속
// AWT 컨테이너 : Panel, Frame, Applet, Dialog, Window
// Swing 컨테이너 : JPanel, JFrame, JApplet, JDialog, JWindow

// 최상위 컨테이너
// 다른 컨테이너에 속하지 않고 독립적으로 출력가능한 컨테이너(JFrame, JDialog, JApplet)
// 모든 컴포넌트는 컨테이너에 포함되어야 화면에 출력 가능

// 컴포넌트
// 컨테이너에 포함되어야 화면에 출력될 수 있는 순수 컴포넌트
// 모든 컴포넌트는 java.awt.Component를 상속받음
// 모든 Swing 컴포넌트는 javax.swing.JComponent를 상속받음
```

```
// 컨테이너와 컴포넌트의 포함관계
```

```
// Swing GUI 프로그램 만들기
```

```
// 1. 스윙 프레임 작성 -> 2. main() 메소드 작성 -> 3. 프레임에 스윙 컴포넌트 붙이기
```

```
// Swing 패키지 사용을 위한 import문
```

```
// import.java.awt.* : 그래픽 처리를 위한 클래스들의 경로명
```

```
// import.java.awt. : AWT 이벤트를 사용하기 위한 경로명
```

```
// import.java.swing.* : 스윙 컴포넌트 클래스들의 경로명
```

```
// import.java.swing.event.* : 스윙 이벤트를 위한 경로명
```

```
// 스윙 프레임
```

```
// 모든 Swing 컴포넌트를 담는 최상위 GUI 컨테이너
```

```
// JFrame을 상속받아 구현을 하며, 컴포넌트가 화면에 보이려면 스윙 프레임에 부착되어야 한다.
```

```
// 프레임을 닫으면 프레임 내의 모든 컴포넌트가 보이지 않게 됨.
```

```
// 스윙 프레임(JFrame) 기본 구성
```

```
// 프레임 - 스윙 프로그램의 기본 틀
```

```
// 메뉴바 - 메뉴들이 부착되는 공간
```

```
// 콘텐츠 팬 - GUI 컴포넌트들이 부착되는 공간
```

```
import javax.swing.*;
```

```
public class MyFrame extends JFrame {
```

```
    public MyFrame() {
```

```
        setTitle("300x300 스윙 프레임 만들기");
```

```
        setSize(300,300); // 프레임 크기 300x300
```

```
        setVisible(true); // 프레임 출력
```

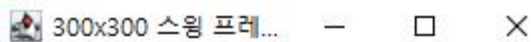
```
    }
```

```
    public static void main(String[] args) {
```

```
        MyFrame frame = new MyFrame();
```

```
    }
```

```
}
```



// main()의 위치를 프레임 클래스 내의 멤버로 작성하거나,
// 다른 클래스에 작성하여 불러올 수 도 있지만 교제는 전자를 권장한다.

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

```
public class ContentPaneEx extends JFrame {
    public ContentPaneEx() {
        setTitle("ContentPane과 JFrame"); // 타이틀 달기
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // 오른쪽 맨 위 X포시 하기

        Container contentPane = getContentPane(); // 컨텐트팬 알아내기
        contentPane.setBackground(Color.ORANGE); // 배경 설정
        contentPane.setLayout(new FlowLayout()); // 레이아웃 설정

        contentPane.add(new JButton("OK")); // 컨텐트 팬에 컴포넌트 달기
        contentPane.add(new JButton("Cancel")); // 컨텐트 팬에 컴포넌트 달기
        contentPane.add(new JButton("Ignore")); // 컨텐트 팬에 컴포넌트 달기

        setSize(300, 150); // 창 사이즈 설정
        setVisible(true); // 화면에 보이게 하는 작업
    }

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



```
// 응용프로그램 내에서 스스로 종료
// System.exit(0); 사용

// 프레임 종료버튼(X)이 클릭되면 어떤 일이 벌어지는가?
// 프레임을 종료하여 프레임 윈도우가 닫힘
// 프레임이 화면에서 보이지 않게 되고 응용프로그램이 사라짐

// 프레임이 보이지 않게 되지만 응용프로그램이 종료한 것 아님
// 키보드나 마우스 입력을 받지 못함
// 다시 setVisible(true)를 호출하면 보이게 되고 이전 처럼 작동함

// 프레임 종료버튼이 클릭될 때 프레임을 닫고 응용 프로그램이 종료하도록 하는 방법
// setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// 스윙 프로그램이 실행되는 동안 생성되는 스레드
// 메인 스레드 - main()을 실행하는 스레드, 자바 응용프로그램의 실행을 시작한 스레드
// 이벤트 분배 스레드 : 스윙 응용프로그램이 실행될 때 자동으로 실행되는 스레드

// 자바 응용프로그램의 종료 조건
// 실행 중인 사용자 스레드가 하나도 없을 때 종료

// 스윙 프로그램 main() 종료 뒤 프레임이 살아 있는 이유
// 메인 스레드가 종료되어도 이벤트 분배 스레드가 살아 있어 프레임 화면을
// 그리고 마우스나 키 입력을 받기 때문에 살아 있다.

// 컨테이너와 배치 개념
// 컨테이너 마다 하나의 배치 관리자가 존재하며, 삽입되는 모든 컴포넌트의 위치와 크기를 결정 및 배치
// 컨테이너의 크기가 변하면 내부 컴포넌트들의 위치와 크기를 모두 재조정하고 재배포한다.
// 배치 관리자(Layout Manager)

// 배치 관리자 대표 유형 4가지, java.awt 패키지에 구현되어 있음
// FlowLayout, BorderLayout, GridLayout, CardLayout

// 컨테이너의 디폴트 배치관리자
// 표

// FlowLayout
// 배치 방법 : 컨테이너 공간 내에 왼쪽에서 오른쪽, 위에서 아래쪽으로 컴포넌트를 배치한다.
// 컨테이너의 크기가 바뀌면 컴포넌트가 재배포된다.
// 생성자
// FlowLayout()
// FlowLayout(정렬방법)
// FlowLayout(정렬방법, 수평간격, 수직간격)
// 정렬방법 : FlowLayout.LEFT , FlowLayout.CENTER , FlowLayout.RIGHT
```

```

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

public class FlowLayoutEx extends JFrame {
    public FlowLayoutEx() {
        setTitle("FlowLayout Sample"); // 제목 입력
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // _ □ X 출력
        Container c = getContentPane(); // 콘텐츠팬 알아내기

        // 컨테이너에 새로운 배치관리자 설정, 왼쪽 정렬, 수평 30픽셀, 수직 40픽셀 조정
        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); // 창 크기
        setVisible(true); // 화면에 출력
    }

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

```



```
// BorderLayout 배치방법
// 컨테이너 공간을 5구역으로 분할, 배치(동, 서, 남, 북, 중앙 순서)
// 배치 방법, add(Component comp, int index)

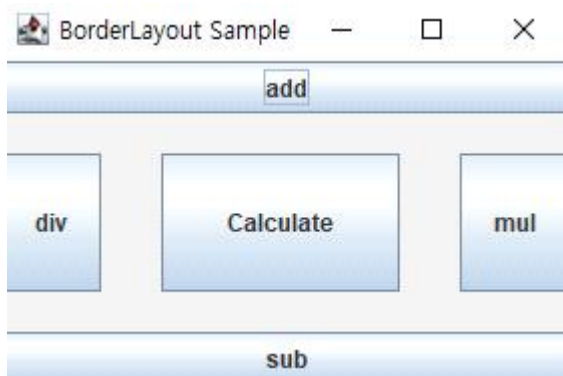
// 생성자
// void add(Component comp, int index)
// comp : 컨테이너에 삽입되는 컴포넌트
// index: 컴포넌트의 위치(동 : BorderLayout.EAST, 서 : BorderLayout.WEST 등등 )

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

public class BorderLayoutEx extends JFrame {
    public BorderLayoutEx() {
        setTitle("BorderLayout Sample");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // _ □ X 출력

        Container c = getContentPane(); // 콘텐츠팬 알아내기
        c.setLayout(new BorderLayout(30, 20)); // 생손자를 바탕으로 가로 30, 세로 20로 표시
        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 300×200 설정
        setVisible(true); // 프레임을 화면에 출력
    }
    public static void main(String[] args) {
        new BorderLayoutEx();
    }
}
```



```
// GridLayout
// 배치방법 : 컨테이너 공간을 동일한 사각형 격자로 분할하고 각 셀에 하나의 컴포넌트 배치
// 격자 구성은 생성자에 행수와 열수 지정, 셀에 왼쪽에서 오른쪽으로, 위에서 아래로 순서대로 배치
// 컨테이너의 크기가 변하면 재배포
// 생성자
// GridLayout()
// GridLayout(행수, 열수)
// GridLayout(행수, 열수, 수평간격, 수직간격)
```

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

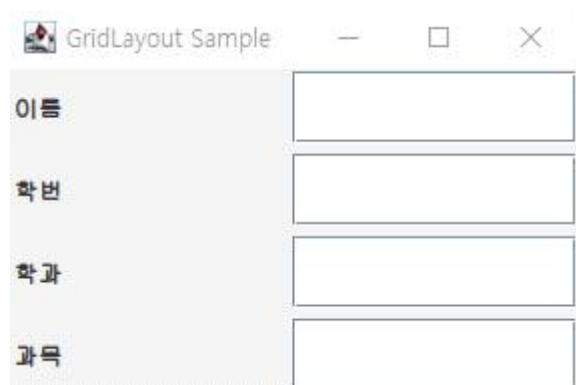
```
public class GridLayoutEx extends JFrame {
    public GridLayoutEx() {

        setTitle("GridLayout Sample");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // _ □ X 출력

        GridLayout grid = new GridLayout(4, 2);
        grid.setVgap(5);

        Container c = getContentPane();
        c.setLayout(grid);
        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();
    }
}
```



이름	<input type="text"/>
학번	<input type="text"/>
학과	<input type="text"/>
과목	<input type="text"/>

```

// 배치관리자가 없는 컨테이너란
// 응용프로그램에서 컴포넌트의 절대 크기와 절대 위치 결정

// 용도
// 컴포넌트의 크기나 위치를 개발자 임의로 결정하고 하고자 하는 경우
// 입력에 따라 컴포넌트들의 위치와 크기가 달라지는 경우
// 여러 컴포넌트들이 서로 겹쳐 출력하는 경우

// 컨테이너의 관리배치자 제거 방법
// container.setLayout(null);

// 컴포넌트의 절대 크기와 절대 위치 설정
// 크기 설정 : component.setSize(너비, 높이)
// 위치 설정 : component.setLocation(x좌표, y좌표);
// 위치와 크기 동시에 설정 : component.setBounds(x좌표, y좌표, 너비, 높이)

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

public class NullContainerEx extends JFrame {
    public NullContainerEx() {

        setTitle("Null Container Sample");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

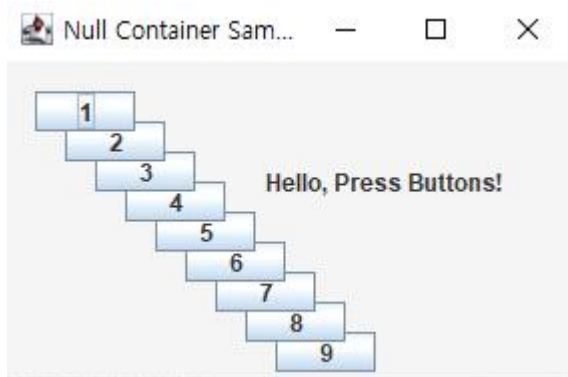
        Container c = getContentPane();
        c.setLayout(null);

        JLabel la = new JLabel("Hello, Press Buttons!");
        la.setLocation(130, 50);
        la.setSize(200, 20);
        c.add(la);

        for(int i=1; i<=9; i++) {
            JButton b = new JButton(Integer.toString(i)); // 버튼 생성
            b.setLocation(i*15, i*15);
            b.setSize(50, 20);
            c.add(b); // 버튼을 컨텐트팬에 부착
        }

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

```




```

import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.*;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.*;

public class Project extends JFrame {
    public Project() {
        setTitle("MyFrame");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container fra = getContentPane();
        fra.setLayout(null);

        JLabel tes = new JLabel("자바 피자에 오신 것을 환영합니다. 피자의 종류를 선택하십시오.");
        tes.setBounds(200, 30, 400, 10);
        fra.add(tes);

        JButton b1 = new JButton("콤보피자"); // 버튼 생성
        b1.setBounds(100, 50, 100, 20);
        fra.add(b1); // 버튼을 컨테인트팬에 부착

        JButton b2 = new JButton("포테이토 피자"); // 버튼 생성
        b2.setBounds(210, 50, 150, 20);
        fra.add(b2); // 버튼을 컨테인트팬에 부착

        JButton b3 = new JButton("불고기 피자"); // 버튼 생성
        b3.setBounds(370, 50, 100, 20);
        fra.add(b3); // 버튼을 컨테인트팬에 부착

        JLabel tes1 = new JLabel("개수: ");
        tes1.setBounds(480, 50, 40, 20);
        fra.add(tes1); // 버튼을 컨테인트팬에 부착

        JTextField textbox = new JTextField("3");
        textbox.setBounds(530, 50, 100, 20);
        fra.add(textbox); // 버튼을 컨테인트팬에 부착

        setSize(1000,300); // 프레임 크기 300x300
        setVisible(true); // 프레임 출력
    }
    public static void main(String[] args) {
        Project frame = new Project();
    }
}

```

