

AWT'ye Giriş

Window ve Layouts

AWT (Abstract Windowing Toolkit) (Soyut Pencereleme Takımı)

- Java'nın grafikli kullanıcı arayüz ortamıdır.
- Pencere, düğme, radyo buton ve menü gibi görsel bileşenler içeren kütüphanedir.

AWT (Abstract Windowing Toolkit) (Soyut Pencereleme Takımı)

- AWT diğer yapılara göre daha büyük bir önem taşımaktadır.Çünkü Java, her işletim sisteminde çalışmak gibi bir iddiayla ortaya çıkan bir dil ama görsel bileşenler her işletim sisteminde çok farklıdır. Bazı bileşenler farklı işletim sistemlerinde aynı şekilde olmuyor. Yani bütün işletim sistemlerinde olan bileşenleri kullanıyor.
- Dahası her işletim sisteminde olan bileşenler de birbirlerinden farklı özellikler içeriyor.

AWT ve Swing

- Java'da daha zengin bir arayüz sağlayan **Swing** kütüphanesi var. O da her işletim sisteminde aynı şekilde çalışmasına rağmen zengin bileşenler içeriyor. Ancak Swing, bunu işletim sisteminin bileşenlerini kullanmadan, her şeyi baştan kendisi yaparak başarabiliyor. Yani Swing'in düğme bileşeni, işletim sistemindeki düğme bileşeni değil. Ona benzetilen, ama baştan yaratılan bir bileşen.

AWT ve Swing

- Swing, kendi bileşenlerini kendi yapıyor, sadece her işletim sistemine uygun bir şekilde görünmelerini sağlıyor. Hatta istenirse bir işletim sisteminde başka bir işletim sisteminin görünümü sağlanabiliyor. Yada hiç bir işletim sisteminde olmayan bir görünüm ortaya çıkabiliyor.

AWT ve Swing

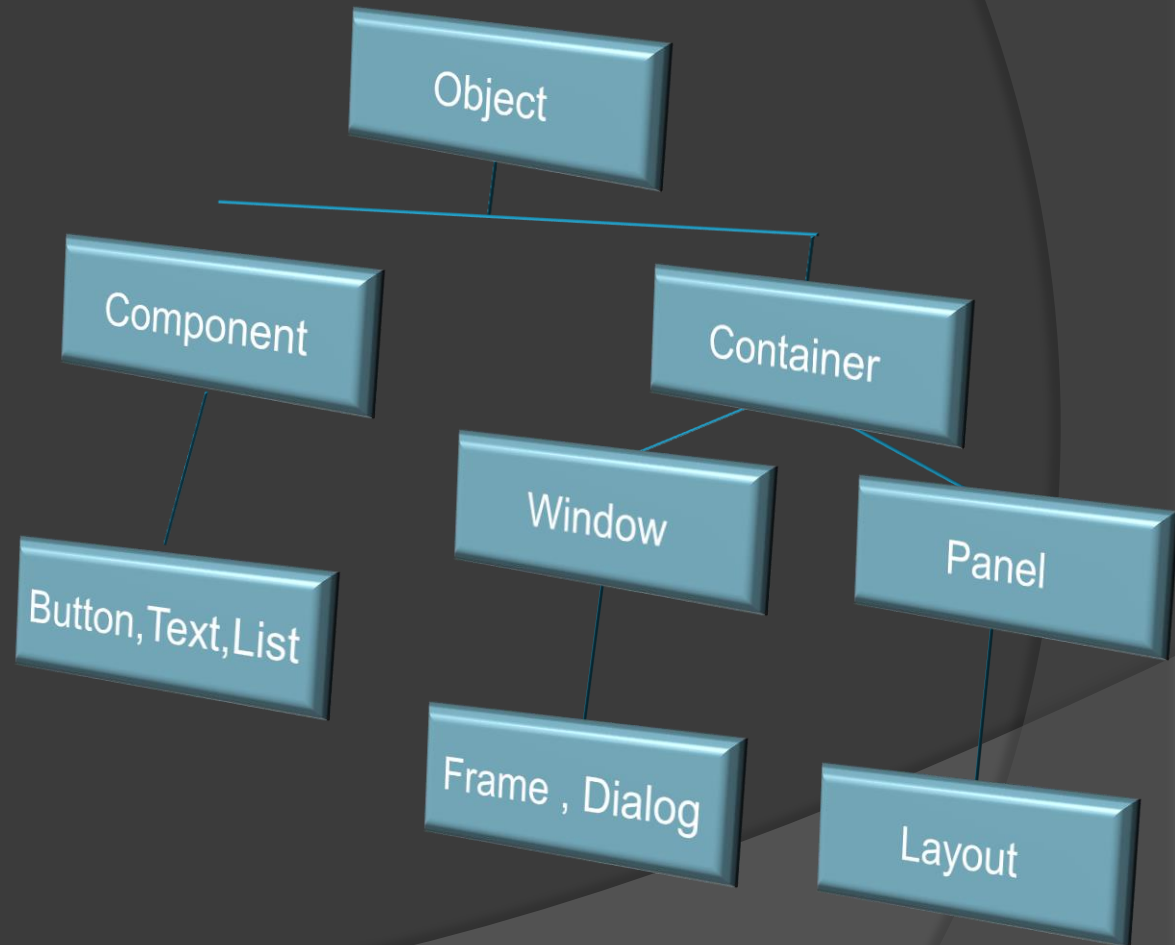
- Swing, AWT'den farklı bir arayüz sağlıyor ama yine de kendisi AWT üzerine kurulu. Yani Swing, AWT'nin temel bileşenlerini kullanıyor. Zaten işletim sisteminin arayüzünün üzerine kurulan AWT. Swing, AWT'deki temel özellikleri kullanarak kendi bileşenini kendi yapmış oluyor. O yüzden AWT'de öğrenilen konuların %90'ı aynen Swing'de de geçerli. AWT'yi basit arayüzü nedeniyle tercih etmeyenler de Swing kullanabilmek için AWT'yi bilmek zorunda.

Çalışma Mantığı

- AWT bütün işletim sistemlerinde çalışabileceği için çalışma mantığı diğerlerinden farklılık göstermektedir. Bütün kütüphanelerde çalışmak zorunda olan bir kütüphane olması gerekmektedir.
- Örneğin çok büyük ekranlı çok yüksek çözünürlüklü makinelerde de küçük ekran ve düşük çözünürlüklü makinelerde de çalışmalı. Bir browser içerisinde de çalışmalı bağımsız bir application olarak da çalışmalı. Bu gibi nedenlerden dolayı farklı bir arayüz sistemlerinde çalışanlar için biraz garip gelebilecek bir yapıya sahip.
- Örneğin, AWT' istediğimizi X ve ya Y noktasını koyamamaktayız. Onun yerine şunu ortaya şunu sola şunu yukarıya koy diyebiliyorsunuz. Belki bir anlamda sınırlayıcı olabilen bu özellik, pencerenin büyümesi veya küçülmesi durumunda da bileşenlerin görünmesini sağladığı için çok avantajlı olabiliyor.
- Bir başka özelliği de arayüzün sadece kod yazarak kolaylıkla oluşturulabilmesi. Basit bir pencerenin içerisine bir tane düğme ve bir yazı alanı koymak için sadece bir kaç satır kod yeterli. Hiç bir görsel editör kullanmaya gerek yok. Elbette isteyen kullanabilir ama editör kullanmadan da çok iyi arayüzler üretilebilir.

Temel Class'lar

- Component
- Container
- Window
-



Component

- Tek başına işlev gören en küçük bileşendir. **Button**, **TextField**, **List** gibi bileşenler birer component'tir.

Container

- İçerisinde başka componentler bulunduran bir component grubudur.

Window

- İçerisinde çeşitli componentler veya container'lar bulunan, ekranda dikdörtgen şeklinde görülebilen bir alan anlamına gelir.

Frame ve Dialog

- Üzerinde title bar, minimize, maximize ve close bulunan window'dur ve birer window'dur.

Label

Container da bulunan textler yazılardır.

Button

Bu sınıf, bir etiketli düğme oluşturur.

Checkbox

Açık ise TRUE kapalı ise FALSE şeklinde onay kutusudur.

Check Box Grup

Gruplanmış halde bulunan onay kutularıdır.

List

List bileşeni içerikleri sıralı bir şekilde listeli kullanıcıya sunmaktadır.

Text Field

Bir metin alanını tek satırlık düzenleme sağladığı yazı alanıdır.

Text Area

Birden fazla metni düzenlemeye yarayan yazı alanıdır.

Choice

Pop up gibi açılır menülerde seçme işlemi için kullanılmaktadır.

Canvas

Kullanıcının çizim yapabildiği aynı zamanda giriş alınabilen alanlardır.

Image

Bir resim düzenleme ekleme alanıdır.

Scroll Bar

Bir kaydırma çubuğu bileşenidir

Dialog

Bilgi almak için ya da işlevsel bir görev için bir üst düzey alanı temsil etmektedir.

File Dialog

Kullanıcıların bunları kullanarak bir dosya ekleme yapabilmelerini sağlayan bileşendir.

JAVA SWING

- Swing Java Foundation Classes (JFC) kütüphanesi içinde Javax.Swing paketidir. 134 sınıfı olan, 25 arayüz ile büyük bir pakettir.
- Java.AWT (Java Abstract Window Toolkit) paketinin üzerine kuruludur ve uygulamalarda onun işlevini üstlenmiştir.
- Bir görsel arayüz hazırlarken, en alta bir çerçeve (JFrame) konulur. Bu çerçeve içine genellikle bir panel konulur. Panel üzerine Button, Text alanı, liste, menü gibi görsel araçlar yerleştirilir.

Örn;

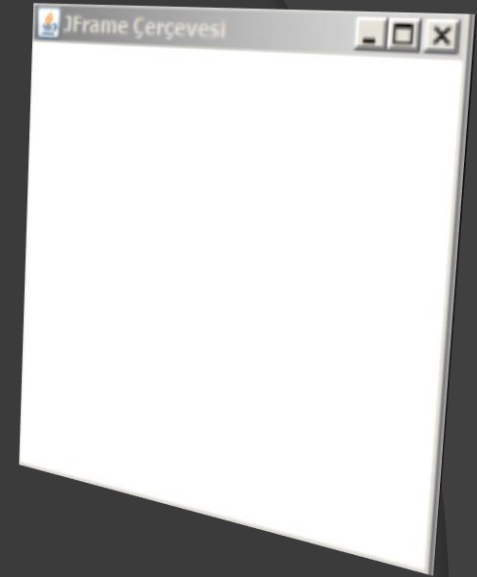
```
import java.awt.*;
import javax.swing.*;

public class JFrameOrn {

    public static void main( String [] args)
    {

        JFrame f = new JFrame("JFrame Çerçevesi");
        f.setSize(250, 250);
        f.setLocation(300, 200);
        f.getContentPane().add(BorderLayout.CENTER, new
        JTextArea(10, 40));
        f.setVisible(true);

    }
}
```



Örn;

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

public class JTableTest extends JFrame {

    String data[][] = {"Didem", "Damla", "Berna"},
        {"Volkan", "Koray", "İlknur"},
        {"Merve", "Fehime", "Sabri"},
        {"Neslihan", "Ceyda", "Mine"},};
    String fields[] = {"Kolon1", "Kolon2", "Kolon3"};

    public static void main(String[] argv) {
        JTableTest jtt = new JTableTest("JTable Örneği");

    }
```

```

public JTableTest(String title) { 8

super(title);
setSize(150, 150);
addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent we) {
dispose();
System.exit(0);
}
});
init();
pack();
setVisible(true);
}

private void init() {
JTable jt = new JTable(data, fields);
JScrollPane pane = new JScrollPane(jt);
getContentPane().add(pane);
}
}

```



The screenshot shows a standard Java Swing window with a title bar containing minimize, maximize, and close buttons. The window contains a JTable with the following data:

Kolon1	Kolon2	Kolon3
Didem	Damla	Berna
Volkan	Koray	Ilknur
Merve	Fehime	Sabri
Neslihan	Ceyda	Mine

Örn;

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

AWT' yi projeye import ettik

```
public class KorelKerem
```

```
{
```

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

```
{
```

```
Frame fra=new Frame();
```

```
fra.setBounds(100,100,400,300);
```

```
fra.setVisible(true);
```

```
}
```

Görülebilir yaptık ve böylece çalıştığında görünecek

Yeni bir "fra" adında frame tanımladık

Frame'in boyutlarını girdik

Component, tek başına belli bir işlev gören en küçük görsel bileşendir. Bir pencere içerisine **Button**, **TextArea**, **CheckBox**, **List** gibi component'ler konulabilir

Örn;

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

```
public class KorelKerem
```

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

```
{
```

```
Frame f=new Frame();
```

```
f.setLayout(new BorderLayout());
```

```
f.setBounds(100,100,400,300);
```

```
Button dugme=new Button("dugme");
```

```
dugme.setBounds(100,100,20,20);
```

```
f.add(dugme);
```

```
f.setVisible(true);
```

```
f.addWindowListener(new WindowAdapter() {  
    public void windowClosing(WindowEvent aw)
```

```
{
```

```
        System.exit(0);
```

```
}
```

```
});
```

Yeni bir BorderLayout tanımladık

Kapatma tuşunun aktifliği

Örn;

```
import java.awt.Frame;  
import java.awt.Color;
```

```
public class mulfra extends Frame
```

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

```
{  
for(int i=0; i<5; i++)
```

```
{  
    new mulfra("Frameler/ KeremKorel " + i);
```

```
}  
}  
public mulfra(String s)
```

```
{  
    super(s);
```

```
    setBackground(new Color((float)Math.random(),  
                             (float)Math.random(),  
                             (float)Math.random()));
```

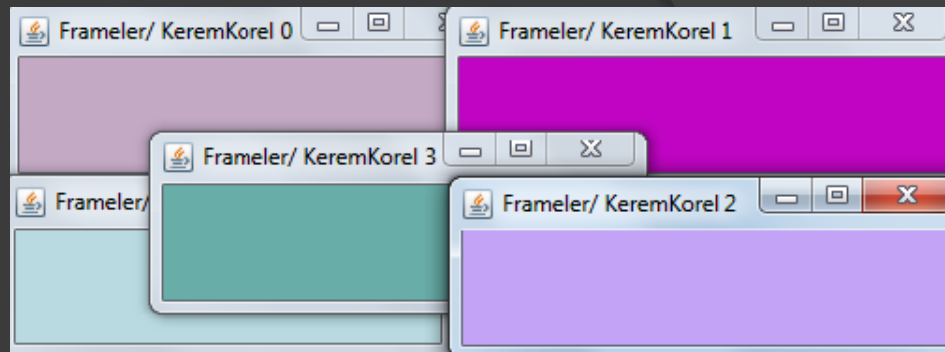
```
    setSize(200,100);
```

```
    setLocation((int)(Math.random() * 600), (int)(Math.random() * 400));
```

```
    setVisible(true);
```

```
}
```

```
}
```



Yeni bir mulfra oluşturuyoruz

Public bir mulfra frame oluşturuyoruz

Layout Manager

- ◎ Java'da Component'lerin nereye nasıl konacağını belirleyen **LayoutManager**'lar vardır.
- ◎ Bir Component'in, dikdörtgen olması nedeniyle ekranda x,y, width ve height gibi özellikleri bulunur. Layout manager'lar sayesinde her component için bu değerleri girmek gerekmez.
 - BorderLayout
 - BoxLayout
 - CardLayout
 - FlowLayout
 - GridLayout
 - GroupLayout
 - SpringLayout

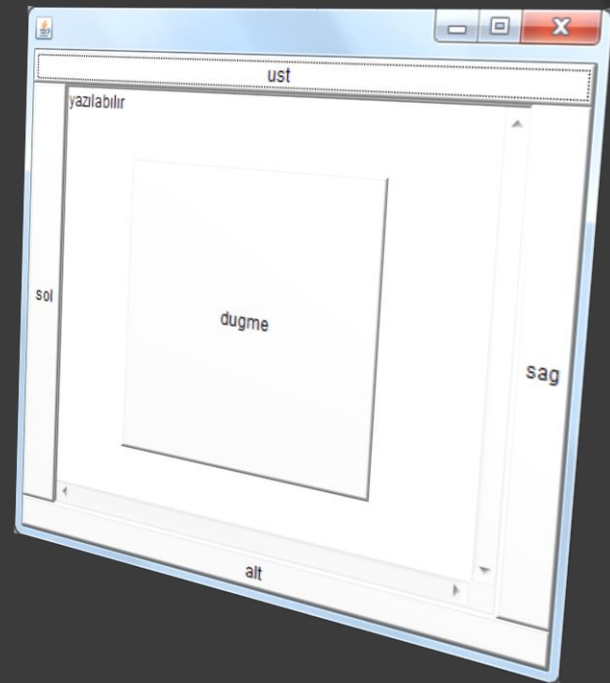
Border Layout

En fazla kullanılan layout'lardan biri **Border Layout**'tur.
Component'leri,

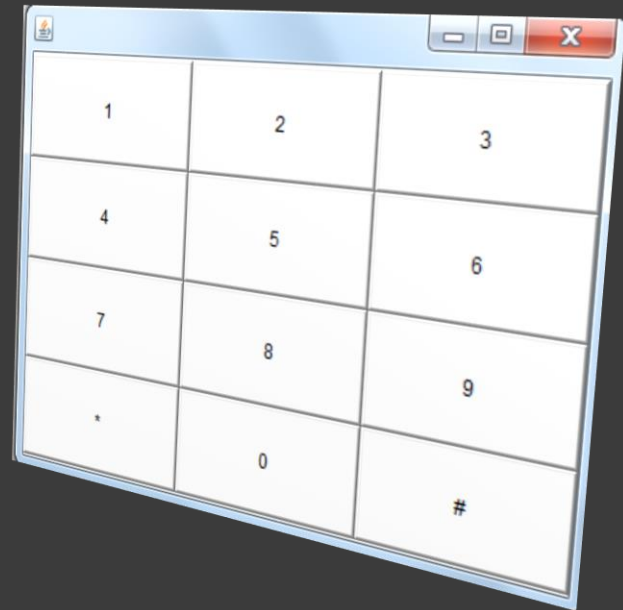
North
West Center East
South

gibi konumlara koyabilir.

```
Frame f= new Frame();  
    f.setLayout(new BorderLayout());  
    f.setBounds (100,100,400,300);  
  
    Button ust=new Button("ust");  
    Button alt=new Button("alt");  
    Button sag=new Button("sag");  
    Button sol=new Button("sol");  
    Button dugme=new Button("dugme");  
    f.add(dugme);  
  
    f.add (ust,BorderLayout.NORTH);  
    f.add (alt,BorderLayout.SOUTH);  
    f.add (sol,BorderLayout.WEST);  
    f.add (sag,BorderLayout.EAST);  
    dugme.setBounds(100,100,20,20);  
    f.add(dugme);  
    TextArea t=new TextArea("yazılabilir");  
  
    f.add(t,BorderLayout.CENTER);  
    f.setVisible(true);
```

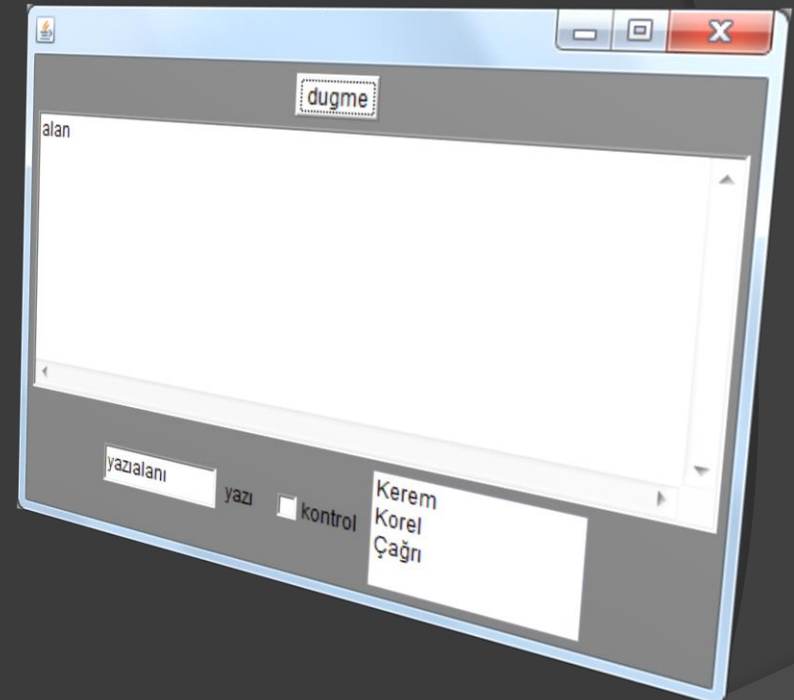


```
import java.awt.*;
public class GridLayouts{ public static void main(String[] args)
{
    Frame f=new Frame();
    f.setLayout(new GridLayout(4,3));
    f.add(new Button("1"));
    f.add(new Button("2"));
    f.add(new Button("3"));
    f.add(new Button("4"));
    f.add(new Button("5"));
    f.add(new Button("6"));
    f.add(new Button("7"));
    f.add(new Button("8"));
    f.add(new Button("9"));
    f.add(new Button("*"));
    f.add(new Button("0"));
    f.add(new Button("#"));
    f.setBounds(100,100,400,300);
    f.setVisible(true);
}
```



```
Frame f=new Frame();

f.setBackground ( SystemColor.GRAY);
f.setLayout(new FlowLayout());
f.add(new Button("Button"));
f.add(new TextArea("Area"));
f.add(new TextField("TextField"));
f.add(new Label("Label"));
f.add(new Checkbox("Checkbox"));
List liste=new List();
liste.add("1");
liste.add("2");
liste.add("");
f.add(liste);
f.setBounds(100,100,600,300);
f.setVisible(true);
```



component'leri soldan sağa doğru sırayla koyar. Bir sıra bitince sol altından yeniden dizer

```

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

public class cardornek {
    private Frame anaframe;
    private Label ustyazi;
    private Label statusLabel;
    private Panel kontrolpanel;

    public cardornek(){
        prepareGUI();
    }

    public static void main(String[] args){
        cardornek awtLayoutDemo = new cardornek();
        awtLayoutDemo.showCardLayoutDemo();
    }

    private void prepareGUI(){
        anaframe = new Frame("Java AWT örneği / KRMKorel");
        anaframe.setSize(400,400);
        anaframe.setLayout(new GridLayout(3, 1));

        ustyazi = new Label();
        ustyazi.setAlignment(Label.CENTER);
        statusLabel = new Label();
        statusLabel.setAlignment(Label.CENTER);
        statusLabel.setSize(350,100);

        kontrolpanel = new Panel();
        kontrolpanel.setLayout(new FlowLayout());

        anaframe.add(ustyazi);
        anaframe.add(kontrolpanel);
        anaframe.add(statusLabel);
        anaframe.setVisible(true);
    }
}

```

```

private void showCardLayoutDemo(){

    ustyazi.setText("CardLayout");

    final Panel panel = new Panel();
    panel.setBackground(Color.CYAN);
    panel.setSize(300,300);

    CardLayout layout = new CardLayout();
    layout.setHgap(10);
    layout.setVgap(10);
    panel.setLayout(layout);

    Panel buttonPanel = new Panel(new FlowLayout());

    buttonPanel.add(new Button("OK"));
    buttonPanel.add(new Button("Cancel"));

    Panel textBoxPanel = new Panel(new FlowLayout());

    textBoxPanel.add(new TextField(20));

    panel.add("Button", buttonPanel);
    panel.add("Text", textBoxPanel);
}

```

```
Choice choice = new Choice();
    choice.add("Button");
    choice.add("Text");

    choice.addItemListener(new ItemListener() {
        public void itemStateChanged(ItemEvent e) {
            CardLayout cardLayout = (CardLayout)(panel.getLayout());
            cardLayout.show(panel, (String)e.getItem());
        }
    });

kontrolpanel.add(choice);

kontrolpanel.add(choice);
kontrolpanel.add(panel);

anaframe.setVisible(true);
}
}
```

