

NESNE YÖNELİMLİ PROGRAMLAMA 2(Object Oriented Programming 2/OOP)

Öğr. Gör. Celil ÖZTÜRK

Marmara Üniversitesi

Teknik Bilimler Meslek Yüksekokulu

İçerik

- ✓ Grafik Sınıfları
 - Barındırıcı Sınıflar
 - Yardımcı Sınıflar
 - Komponent Sınıfları
- ✓ Buton Renklendirme
- ✓ Buton Font Ayarları
- ✓ Paneller

Barındırıcı Sınıflar

- Diğer sınıfları barındırmak için kullanılırlar.
- JFrame: Diğer swing komponentlerini barındıran ve başka bir pencere içine yerleşmeyen penceredir.
- JDialog: Pop-up penceresi veya mesaj kutusudur.
- JApplet: Swing tabanlı applet yazmak için kullanılır.
- JPanel: Diğer komponentleri barındıran ancak görünmeyen sınıftır. Paneller iç içe kullanılabilirler.

Yardımcı Sınıflar

- Bu sınıflar barındırıcı ve komponent sınıflar tarafından nesnelerin çizilmesi ve yerleştirilmesi için kullanılır.

Graphics: Soyut bir sınıftır. Geometrik şekillerin çizilmesini sağlar.

Color: Komponentlerin renklerinin belirlenmesini sağlar.

Font: Komponentlerde bulunan yazıların yazı karakterlerinin belirlenmesini sağlar.

Fontmetrics: Soyut bir sınıftır. Fontların özelliklerini sağlar.

Dimension: Komponentlerin en ve boy uzunluklarının belirlenmesini sağlar.

LayoutManager: Arabirimdir. Barındırıcılar içinde bulunan komponentlerin nasıl dizileceğini belirler

Komponent Sınıfları

- Nesnelerin olduğu sınıflardır.
 - JButton, JTextField, JLabel vb

Grafik Sınıfları

```
package guiprojesi1;
import javax.swing.JFrame;
public class Main {
    public static void main(String[] args) {

        JFrame pencere=new JFrame("İlk penceremiz"); //Pencerenin başlığı

        pencere.setSize(300,400); //Pencerenin boyutu

        Pencere.setLocationRelativeTo(null); //Pencerenin yerini başka bir komponente göre ayarlamak için kullanılır. null
        ise pencerenin ortasında çıkar.

        pencere.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //Pencere kapatıldığında programın kapatılmasını
        sağlar.

        pencere.setVisible(true); //Pencerenin görülmesini sağlar.
    } }
```

Grafik Sınıfları

```
package guiprojesi1;
import javax.swing.*.*;
public class Main {
    public static void main(String[] args) {
        JFrame pencere=new JFrame("İlk penceremiz");
        JButton buton=new JButton("Tamam");
        pencere.add(buton);
        pencere.setSize(300,400);
        pencere.setLocationRelativeTo(null); pencere.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pencere.setVisible(true);
    }
}
```

Buton bütün pencereyi kaplar. Bunun sebebi LayoutManager'dan herhangi birinin kullanılmaması nedeniyle default olarak kullanılan Layout yöneticisidir. Bu yönetici nesneyi tam ortaya yerleştirir

Yerleşim Yöneticileri/ **FlowLayout**

- Diğer programlama dillerinde nesneler pixel pixel koordinatlar belirlenerek yerleştirilirler. Ancak bu durum bazı problemlere yol açmaktadır. Bu nedenle Java'da yerleşim yöneticileri kullanılır.
FlowLayout:Barındırıcıda bulunan komponentlerin soldan sağa doğru yerleştirilmesini sağlar. Komponentler bir satıra sığmazsa alt satıra geçer.

Yerleşim Yöneticileri/ **FlowLayout**

- **LayoutManagerPenceresi.java**

- package guiprojesi1;
- import java.awt.FlowLayout;
- import javax.swing.*;
- public class LayoutManagerPenceresi extends JFrame{
- public LayoutManagerPenceresi(){
- FlowLayout layout=new FlowLayout(**FlowLayout.LEFT,15,30**);
- setLayout(layout);
- add(new JLabel("ad"));
- add(new JTextField(20));
- add(new JLabel("soyad"));
- add(new JTextField(20));
- add(new JLabel("yas"));
- add(new JTextField(4)); } }

Yerleşim Yöneticileri/ **FlowLayout**

FlowLayout.LEFT : Komponentleri sola yaslanmasını sağlar.

15: Komponentler arasında yataydaki boşluk (pixel)







30: Komponentler arasında dikeydeki boşluk (pixel)

```
FlowLayout layout=new FlowLayout(FlowLayout.LEFT,15,30);
```

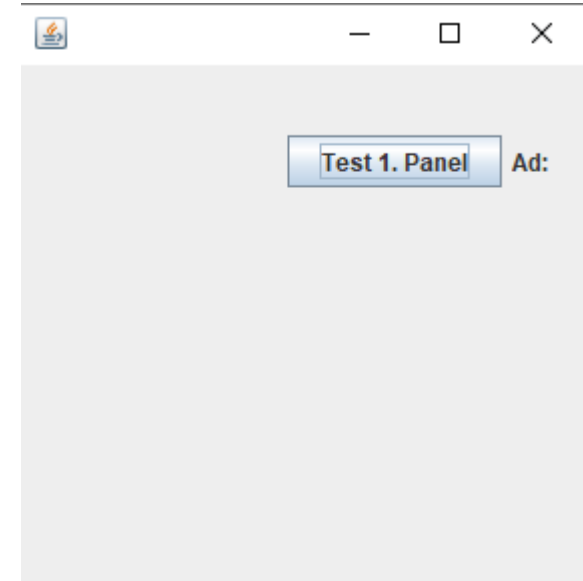
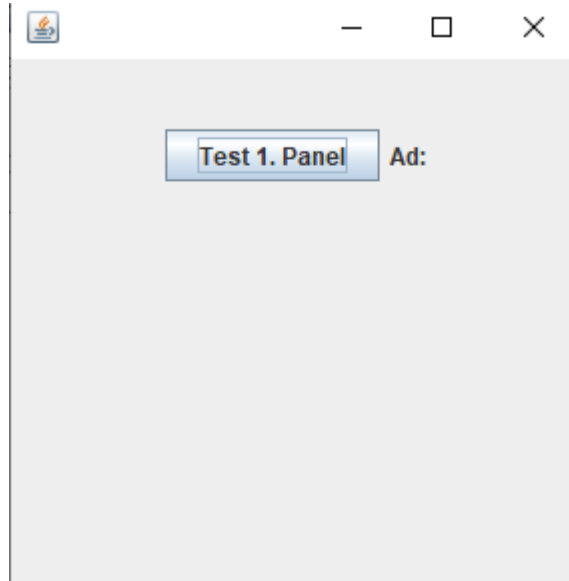
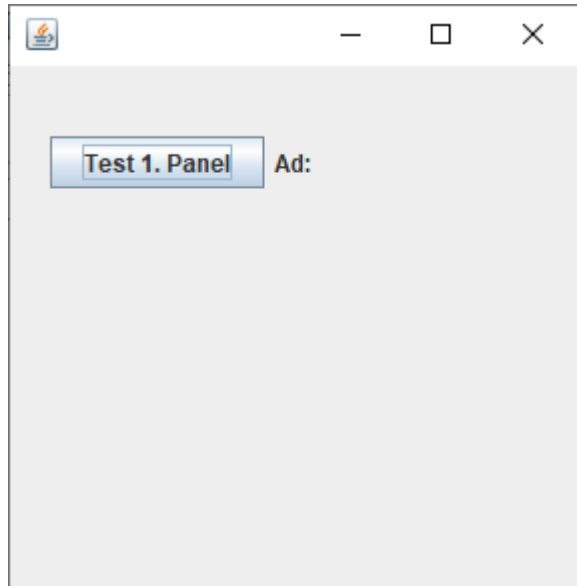
```
frame.setLayout(layout);
```

```
frame.setVisible(true);
```

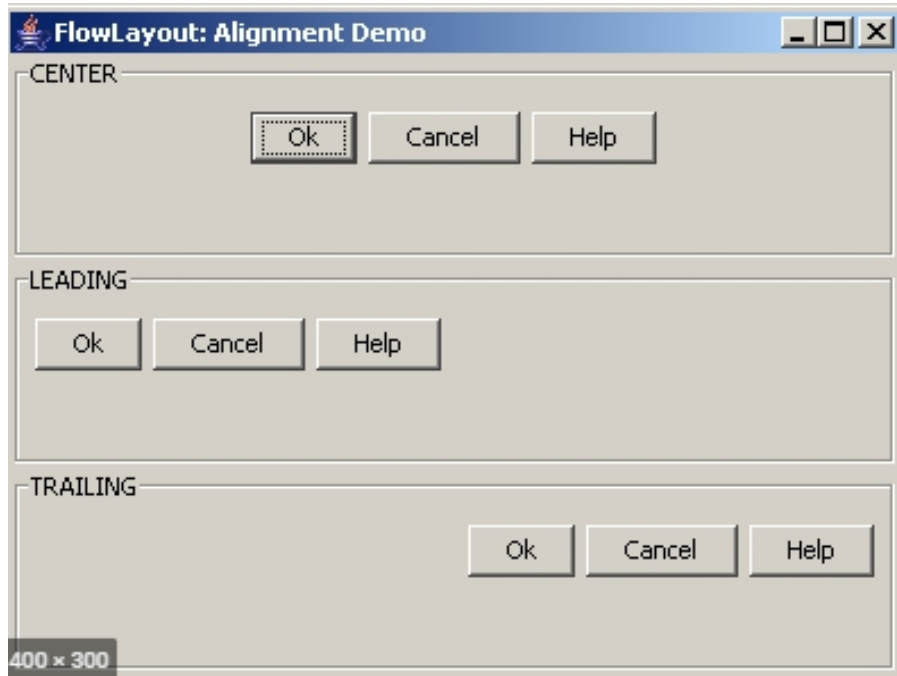
```
buton1.addActionListener(new ActionListener{
```

	CENTER	int
	LEADING	int
	LEFT	int
	RIGHT	int
	TRAILING	int
	class	

Yerleşim Yöneticileri/ **FlowLayout**



Yerleşim Yöneticileri/ **FlowLayout**



LEADING

Bu değer, her bileşen sırasının konteynerin yönünün ön kenarına, örneğin soldan sağa yönlerde sola hizalanması gerektiğini belirtir.

TRAILING

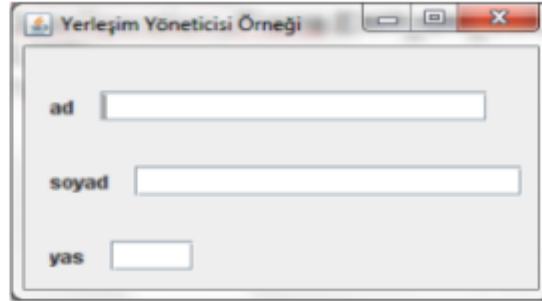
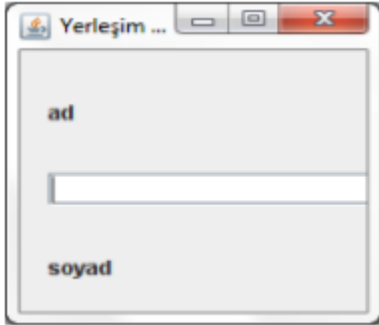
Bu değer, her bileşen sırasının, kabın yönünün arka kenarına, örneğin soldan sağa yönlerde sağa hizalanması gerektiğini belirtir.

Yerleşim Yöneticileri/ **FlowLayout**

Main.java

```
package guiprojesi1;
import javax.swing.*.*;
public class Main {
    public static void main(String[] args) {
        LayoutManagerPenceresi pencere=new LayoutManagerPenceresi();
        pencere.setTitle("Yerleşim Yöneticisi Örneği");
        pencere.setLocationRelativeTo(null);
        pencere.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pencere.setSize(200,200);
        pencere.setVisible(true);
    }
}
```

Yerleşim Yöneticileri/ **FlowLayout**



NOT: Eğer programda bir yerleşim yöneticisi atanmazsa, FlowLayout default yerleştirici olarak kullanılır.

Yerleşim Yöneticileri/ GridLayout

2.GridLayout:

Barındırıcıda bulunan komponentleri satır ve sütunları, daha önce belirlenen bir tablo şeklinde yerleştirir. Burada komponentler tabloya yerleştirme sırasına göre tablonun birinci satırından itibaren soldan sağa doğru sıralanır.

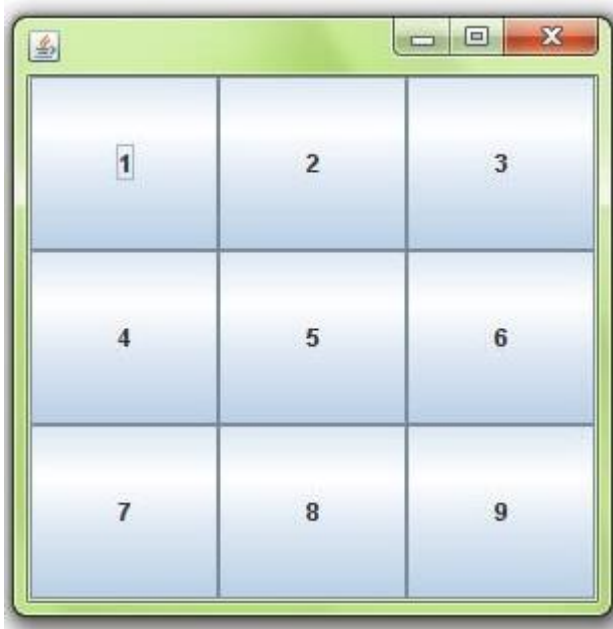
Satır ve sütun sayısından herhangi biri 0 olabilir. Ama ikisi birden 0 olamaz. Biri 0 olduğunda komponentler 0 olmayana göre sıralanır. Örneğin satır sayısı 3 ve sütun sayısı 0 ise ve 10 tane komponent var ise bu durumda satır sayısı sabit kalmakta ve sütun sayısı 4 olmaktadır.

Yerleşim Yöneticileri/ **GridLayout**

Eğer satır ve sütun sayıları 0 değil ise, önemli olan satır sayısıdır. Yani satır sayısı sabit kabul edilir. Sütun sayısı komponent sayısına göre ayarlanmaktadır. Örneğin satır sayısı 3, sütun sayısı 2 ve komponent sayısı 10 olsun. Bu durumda satır sayısı sabit kalacağından sütun sayısı otomatik olarak 4 olacaktır.

1.GridLayout(int rows, int columns, int hgap, int vgap): erilen yatay ve dikey boşlukların yanı sıra verilen satır ve sütunlarla bir ızgara düzeni oluşturur.

Yerleşim Yöneticileri/ **GridLayout**



```
f=new JFrame();

JButton b1=new JButton("1");
JButton b2=new JButton("2");
JButton b3=new JButton("3");
JButton b4=new JButton("4");
JButton b5=new JButton("5");
    JButton b6=new JButton("6");
    JButton b7=new JButton("7");
JButton b8=new JButton("8");
    JButton b9=new JButton("9");

f.add(b1);f.add(b2);f.add(b3);f.add(b4);f.add(b5);
f.add(b6);f.add(b7);f.add(b8);f.add(b9);

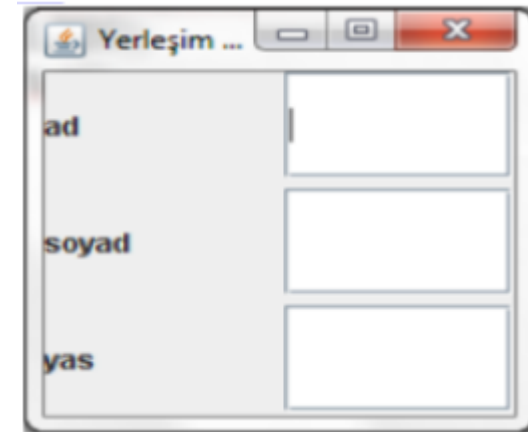
f.setLayout(new GridLayout(3,3));
//setting grid layout of 3 rows and 3 columns

f.setSize(300,300);
f.setVisible(true);
```

Yerleşim Yöneticileri/ GridLayout

LayoutManagerPenceresi

```
package guiprojesi1;
import java.awt.*;
import javax.swing.*;
public class LayoutManagerPenceresi extends JFrame{
    public LayoutManagerPenceresi(){
        GridLayout layout=new GridLayout(3,2,5,5);
        setLayout(layout);
        add(new JLabel("ad"));
        add(new JTextField(20));
        add(new JLabel("soyad"));
        add(new JTextField(20));
        add(new JLabel("yas"));
        add(new JTextField(4));
    }
}
```



Yerleşim Yöneticileri/ GridLayout

```
package guiprojesi1;
import javax.swing.*.*;
public class Main {
    public static void main(String[] args) {
        LayoutManagerPenceresi pencere=new LayoutManagerPenceresi();
        pencere.setTitle("Yerleşim Yöneticisi Örneği"); pencere.setLocationRelativeTo(null);
        pencere.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pencere.setSize(200,200);
        pencere.setVisible(true);
    }
}
```

Yerleşim Yöneticileri/ BorderLayout

- **3.BorderLayout**: Uygulandığı pencereyi 5 bölgeye ayırır. Doğu, Batı, Kuzey, Güney, Merkez.

Barındırıcıya komponentleri eklerken ekleyeceğimiz bölgeyi de belirtmemiz gerekir

Yerleşim Yöneticileri/ BorderLayout

LayoutManagerPenceresi.java

```
package guiprojesi1;
import java.awt.*;
import javax.swing.*;
public class LayoutManagerPenceresi extends JFrame{
    public LayoutManagerPenceresi(){
        BorderLayout layout=new BorderLayout(10,15);
        setLayout(layout);
        add(new JButton("batı"),BorderLayout.WEST);
        add(new JButton("doğu"),BorderLayout.EAST);
        add(new JButton("kuzey"),BorderLayout.NORTH);
        add(new JButton("güney"),BorderLayout.SOUTH);
        add(new JButton("merkez"),BorderLayout.CENTER);
    }
}
```

Yerleşim Yöneticileri/ BorderLayout

Main.java package guiprojesi1;

import javax.swing.*;

public class Main {

public static void main(String[] args) {

LayoutManagerPenceresi pencere=new LayoutManagerPenceresi();

pencere.setTitle("Yerleşim Yöneticisi Örneği");

pencere.setLocationRelativeTo(null);

pencere.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

pencere.setSize(200,200);

pencere.setVisible(true);

}}



Grafik Sınıfı/ Buton Renklendirme

Buton Renklendirme

```
Color renk=new Color(255,255,0);
```

```
JButton batibutonu=new JButton("batı");
```

```
batibutonu.setBackground(renk);
```

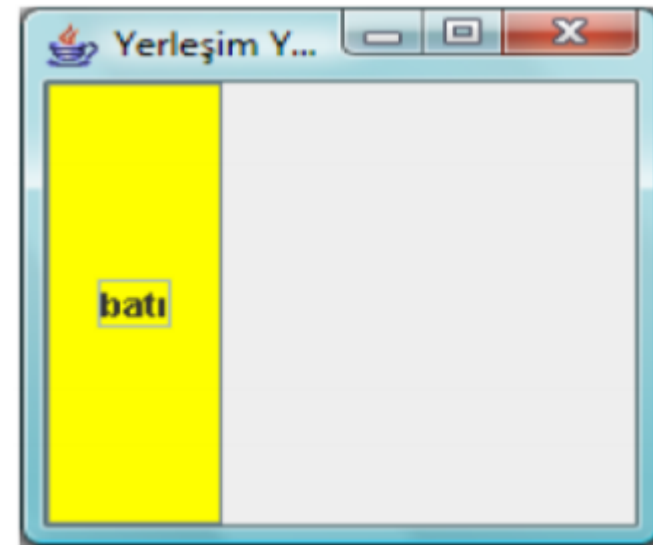
```
add(batibutonu,BorderLayout.WEST);
```

veya

```
JButton batibutonu=new JButton("batı");
```

```
batibutonu.setBackground(Color.YELLOW);
```

```
add(batibutonu,BorderLayout.WEST);
```



Grafik Sınıfı/ Buton Font Ayarları

```
Font yazi=new Font("SanSerif",Font.BOLD,20);  
batibutonu.setFont(yazi);
```



Grafik Sınıfı/ Paneller

Paneller Frame'ler gibi barındırıcıdır.

Ancak Frame'lerden farkı bir pencerede birden fazla panel kullanılabilir ve paneller içiçe kullanılabilir.

Paneller komponentleri pencereye istediğimiz gibi yerleştirmemizi sağlar.

Grafik Sınıfı/ Paneller

```
public static void main(String[] args) {  
    JFrame frame = new JFrame();  
    frame.setSize(300, 300);  
    JPanel panel1 = new JPanel();  
    JPanel panel2 = new JPanel();  
  
    JButton buton1=new JButton("Test 1. Panel");  
    JButton buton2 = new JButton("Test 2. Panel");  
    JLabel labell=new JLabel("Ad:");  
    panel1.add(buton1);  
    panel1.add(buton1);  
    panel1.add(buton1);  
  
    panel1.add(labell);  
    panel1.add(labell);  
  
    panel2.add(buton2);  
    frame.add(panel1);  
    FlowLayout layout=new FlowLayout (FlowLayout.LEFT,15,30);  
    frame.setLayout (layout);  
}
```

Grafik Sınıfı/ Paneller

PanelOrnegiPenceresi.java

```
package guiprojesi1;
import java.awt.*;
import javax.swing.*;
public class PanelOrnegiPenceresi extends JFrame {
public PanelOrnegiPenceresi(){
    JPanel p1=new JPanel();
    p1.setLayout(new GridLayout(2,2));
    p1.add(new JButton("button1"));
    p1.add(new JButton("button2"));
    p1.add(new JButton("button3"));
    p1.add(new JButton("button4"));
    JPanel p2=new JPanel();
    p2.setLayout(new BorderLayout());
    p2.add(p1,BorderLayout.CENTER);
    p2.add(new JButton("Sol Buton"),BorderLayout.WEST);
    add(p2);
}}
```

Grafik Sınıfı/ Paneller

Main.java

```
package guiprojesi1;
import javax.swing.*.*;
public class Main {
    public static void main(String[] args) {
        PanelOrnegiPenceresi pencere=new PanelOrnegiPenceresi();
        pencere.setTitle("Panel Örneği");
        pencere.setLocationRelativeTo(null);
        pencere.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pencere.setSize(400,250);
        pencere.setVisible(true);
    }
}
```

Kaynaklar

- Java ve Java Teknolojileri, *Tevfik KIZILÖREN* – Kodlab Yayınları
- <https://javatutorial.net/jframe-buttons-listeners-text-fields>
- <https://docs.oracle.com/search/?category=java&product=e25407-01&q=setLayout>
- <https://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>
- <https://javatutorial.net/jframe-buttons-listeners-text-fields>
- Dr Öğr. Üyesi Zehra Aysun ALTIKARDEŞ Nesne Yönelimli Programlama 2 Ders notları
- https://docs.oracle.com/cd/E17802_01/j2se/j2se/1.5.0/jcp/beta1/apidiffs/java/awt/FlowLayout.html