

Görsel Programlama

DERS 07

KULLANICI ARAYÜZÜ GELİŞTİRME

Kullanıcı arayüzleri (GUI-Graphical User Interface) kullanıcının işletim sisteminin pencere yöneticisinde görebildiği, mouse, klavye gibi girdi araçlarıyla etkileşimde bulunduğu nesnelerdir.

Kullanıcı arayüzü geliştirirken Java API'leri içerisinde daha önceden hazırlanmış olan sınıfları-görsel bileşenleri (component) kullanacağız.

KULLANICI ARAYÜZÜ GELİŞTİRME

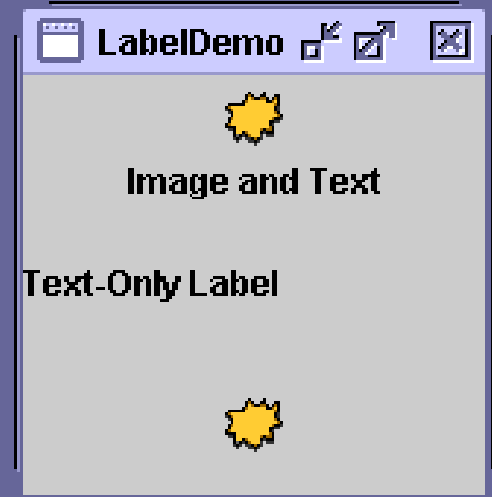
Java içinde 2 farklı kullanıcı arayüzü geliştirme API si vardır:

- 1.) **java.awt**: paketindeki bileşenlerdir. (AWT-Abstract Windowing Toolkit) java da ilk geliştirilen görsel bileşenlerdir ve performansları kötüdür.
- 2.) **javax.swing**: paketindeki bileşenlerdir. Performansları iyidir.

KULLANICI ARAYÜZÜ GELİŞTİRME

Swing Bileşenleri

JLabel: Ekranda bir yazı ya da ikon gösterir.



JTextField: Kullanıcının klavyeden veri girişi yaptığı yazı kutusudur.

Years:

KULLANICI ARAYÜZÜ GELİŞTİRME

JButton,
JCheckBox,
JRadioButton:

Tıklama olayına tepki veren düğmeler.

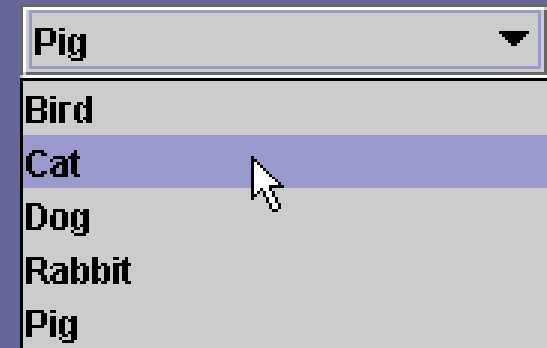


KULLANICI ARAYÜZÜ GELİŞTİRME

JList: Elemanlarını bir liste şeklinde tutar. Kullanıcı listedeki elemanlara tıklayarak seçim yapar.

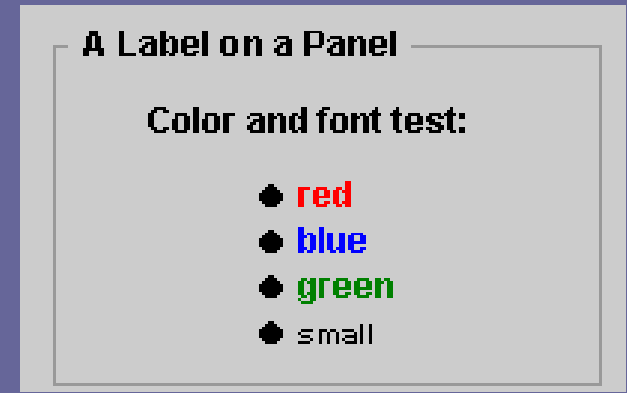


JComboBox: Tık Açılır listede elemanlarını tutar.



KULLANICI ARAYÜZÜ GELİŞTİRME

JPanel: Bir içerici(container) dır, üzerinde başka bileşenleri tutmakla görevlidir.



JFrame: Penceredir ve üzerinde diğer bileşenleri tutar.



KULLANICI ARAYÜZÜ GELİŞTİRME

JTable: Bir tablo şeklinde bileşenleri tutar.

First Name	Last Name	Favorite Food
Jeff	Dinkins	
Ewan	Dinkins	
Amy	Fowler	
Hania	Gajewska	
David	Geary	

JTree: Ağaç yapısında elemanları tutar.



Bileşen Hiyerarşisi

Tüm GUI bileşenleri (component) dolaylı ya da direkt olarak **Component** sınıfından türetilmiştir. Bu bileşenin en önemli metotları **paint**, **repaint** ve **update** tir. Bu metotları kendini ekrana çizmek için kullanır.

```
java.lang.Object
├── java.awt.Component
│   ├── java.awt.Container
│       └── javax.swing.JComponent
```

Bileşen Hiyerarşisi

javax.swing

Class JComponent

```
java.lang.Object
├── java.awt.Component
│   └── java.awt.Container
│       └── javax.swing.JComponent
```

All Implemented Interfaces:

[ImageObserver](#), [MenuContainer](#), [Serializable](#)

Direct Known Subclasses:

[AbstractButton](#), [BasicInternalFrameTitlePane](#), [Box](#), [Box.Filler](#), [JColorChooser](#), [JComboBox](#), [JFileChooser](#), [JInternalFrame](#), [JInternalFrame.JDesktopIcon](#), [JLabel](#), [JLayeredPane](#), [JList](#), [JMenuBar](#), [JOptionPane](#), [JPanel](#), [JPopupMenu](#), [JProgressBar](#), [JRootPane](#), [JScrollBar](#), [JScrollPane](#), [JSeparator](#), [JSlider](#), [JSpinner](#), [JSplitPane](#), [JTabbedPane](#), [JTable](#), [JTableHeader](#), [JTextComponent](#), [JToolBar](#), [JToolTip](#), [JTree](#), [JViewport](#)

Yerleşim Yöneticileri(Layouts)

Yerleşim yöneticileri (Layout Manager), kap (container) {JFrame,JPanel,Panel,Frame} üzerine eklenen bileşenlerin nasıl konumlandırılacağını belirleyen özel Java sınıflarıdır.

Java da farklı yerleşim yöneticileri bulunmaktadır

- FlowLayout
- BorderLayout
- GridLayout
- CardLayout
- GridBagLayout
- BoxLayout
- SpringLayout
- null

Yerleşim Yöneticileri(Layouts)

FlowLayout:

Bileşenler ardışık olarak soldan sağa doğru satır üzerinde yer bitene kadar dizilirler. Satır bittiğinde bir sonraki satırdan bileşenler eklenmeye devam edilir.

Yerleştirilen bileşenler sola,ortaya,sağa dayalı olarak yerleştirilebilirler.

Bunu belirlemek için FlowLayout un yapıcı fonksiyonuna bu bilgi verilir.

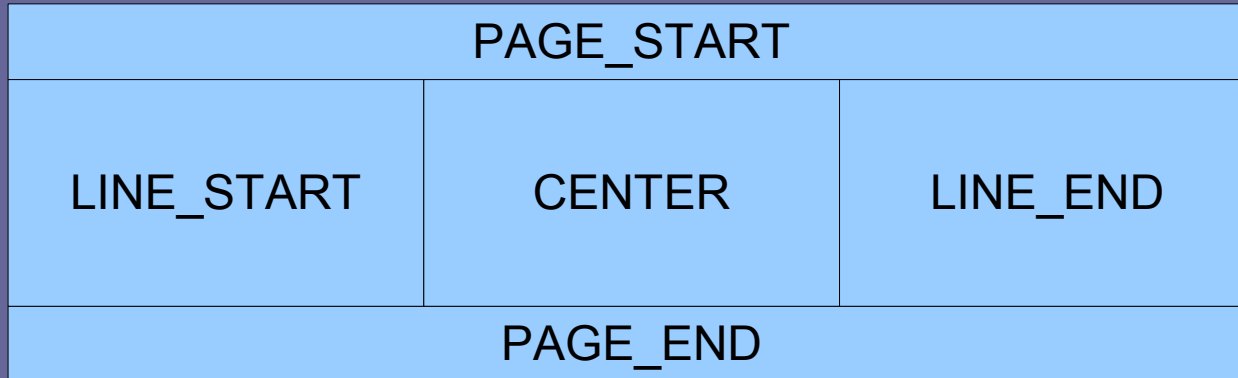
- FlowLayout.RIGHT
- FlowLayout.CENTER
- FlowLayout.LEFT

FlowLayout

```
FlowLayoutOrnek.java X
package com.comu.gorsel_programlama.ders07;
import java.awt.Container;
public class FlowLayoutOrnek extends JFrame {
    private JButton b1,b2,b3;
    public FlowLayoutOrnek() {
        super("Flow Layout Örnek");
        createGUI();
    }
    private void createGUI() {
        Container con = getContentPane();
        //bilesenlerin uzerine yerlestirilecegi icerici
        con.setLayout(new FlowLayout());
        b1 = new JButton("Düğme1");
        b2 = new JButton("Düğme2");
        b3 = new JButton("Düğme3");
        con.add(b1);
        con.add(b2);
        con.add(b3);
        this.setSize(500,300);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public static void main(String[] args) {
        FlowLayoutOrnek app = new FlowLayoutOrnek();
        app.setVisible(true);
    }
}
```

BorderLayout

Kap (container) bileşenini 5 bölgeye ayırır, bileşenler bu 5 farklı bölgeden birine yerleştirilir.



BorderLayout

```
BorderLayoutOrnek.java X
package com.comu.gorsel_programlama.ders07;
import java.awt.BorderLayout;

public class BorderLayoutOrnek extends JFrame {
    public BorderLayoutOrnek() {
        createGUI();
    }
    private void createGUI() {
        Container con = getContentPane();
        con.setLayout(new BorderLayout());
        con.add(new JButton("üst"), BorderLayout.PAGE_START);
        con.add(new JButton("sol"), BorderLayout.LINE_START);
        con.add(new JButton("orta"), BorderLayout.CENTER);
        con.add(new JButton("sağ"), BorderLayout.LINE_END);
        con.add(new JButton("alt"), BorderLayout.PAGE_END);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(500, 300);
    }
    public static void main(String[] args) {
        BorderLayoutOrnek app = new BorderLayoutOrnek();
        app.setVisible(true);
    }
}
```

GridLayout

Bu yerleşim yöneticisi kabı eşit hücrelerden oluşan bir tablo gibi parçalara ayırır. Eklenen her bileşen bu tablo üzerindeki bir hücre içerisine konumlandırılır.

GridLayout

```
GridLayoutOrnek.java X
package com.comu.gorsel_programlama.ders07;
import java.awt.Color;

public class GridLayoutOrnek extends JFrame {
    public GridLayoutOrnek() {
        createGUI();
    }
    private void createGUI() {
        Container con = getContentPane();
        con.setLayout(new GridLayout(0,3));
        JButton bjk = new JButton("BJK");
        bjk.setBackground(Color.BLACK);
        bjk.setForeground(Color.WHITE);
        con.add(bjk);
        JButton gs = new JButton("GS");
        gs.setBackground(Color.YELLOW);
        gs.setForeground(Color.RED);
        con.add(gs);
        JButton fb = new JButton("FB");
        fb.setBackground(Color.BLUE);
        fb.setForeground(Color.YELLOW);
        con.add(fb);
        this.setSize(250,150);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public static void main(String[] args) {
        GridLayoutOrnek app = new GridLayoutOrnek();
        app.setVisible(true);
    }
}
```

NULL LAYOUT

Herhangi bir yerleşim yöneticisi kullanmak istemiyorsak `setLayout(null)` metoduna `null` değerini atarız. Bileşenler kaba eklenirken bizim verdiğimiz koordinatlara göre yerleştirilir.

NULL LAYOUT

```
NullLayoutOrnek.java X
package com.comu.gorsel_programlama.ders07;
import java.awt.Container;

public class NullLayoutOrnek extends JFrame {
    public NullLayoutOrnek() {
        super("Null Layout Ornek");
        createGUI();
    }
    private void createGUI() {
        Container con = getContentPane();
        con.setLayout(null);
        JButton b1 = new JButton("B-1");
        JTextField f = new JTextField();
        f.setBounds(10, 40, 100, 25);
        JButton b2 = new JButton("B-2");
        b1.setBounds(new Rectangle(10,10,100,25));
        b2.setBounds(10,70,100,25);
        con.add(b1);
        con.add(b2);
        con.add(f);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setSize(250, 125);
    }
    public static void main(String[] args) {
        NullLayoutOrnek app = new NullLayoutOrnek();
        app.setVisible(true);
    }
}
```

KAPLAR (CONTAINERS)

Kaplar içerisinde başka bileşenleri içeren bileşenlerdir.

1. **Üst Seviyeli Kaplar:** Ana uygulama pencereleridir. Bu seviyeli kaplar başka kapların içerisine yerleştirilemezler. Bu kapların içerisine alt seviyeli kaplar konabilir.

(JFrame,JDialog,JApplet : swing; Frame,Dialog,Applet:awt)

2. **Ara Kaplar :** Atomik bileşenleri ve diğer ara kapları içerir.
(Jpanel,JScrollPane,JSplitPane,...)

3. **Atomik Bileşenler:** Kaplar içerisine yerleştirilen temel bileşenlerdir. (JButton,JLabel,JTextField,...)

```
CokluPanel.java X
public class CokluPanel extends JFrame {
    public CokluPanel() {
        createGUI();
    }
    private void createGUI() {
        this.getContentPane().setLayout(new GridLayout(1,2,10,10));
        JPanel p1 = new JPanel();
        JPanel p2 = new JPanel();
        JPanel p3 = new JPanel();
        JPanel p4 = new JPanel();
        p1.setLayout(new GridLayout(9,1,5,5));
        p1.add(new JLabel("Ürünler"));
        p1.add(new JCheckBox("T-shirt"));
        p1.add(new JCheckBox("Gomlek"));
        p1.add(new JCheckBox("Şort"));
        p1.add(new JCheckBox("Sapka"));
        p1.add(new JCheckBox("Ayakkabı"));
        p1.add(new JCheckBox("Çanta"));
        p1.add(new JLabel("Toplam Tutar"));
        p1.add(new JTextField());
        p2.add(new JButton("Gönder"));
        p3.add(new JLabel("Adres ve Telefon"));
        p4.setLayout(new BorderLayout(3,3));
        JTextArea tAlan = new JTextArea();
        JScrollPane sPane = new JScrollPane(tAlan);
        p4.add(sPane,BorderLayout.PAGE_START);
        p4.add(p2,BorderLayout.PAGE_END);
        this.getContentPane().add(p1);
        this.getContentPane().add(p4);
        this.setLocation(50,30);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(400,250);
    }
}
```

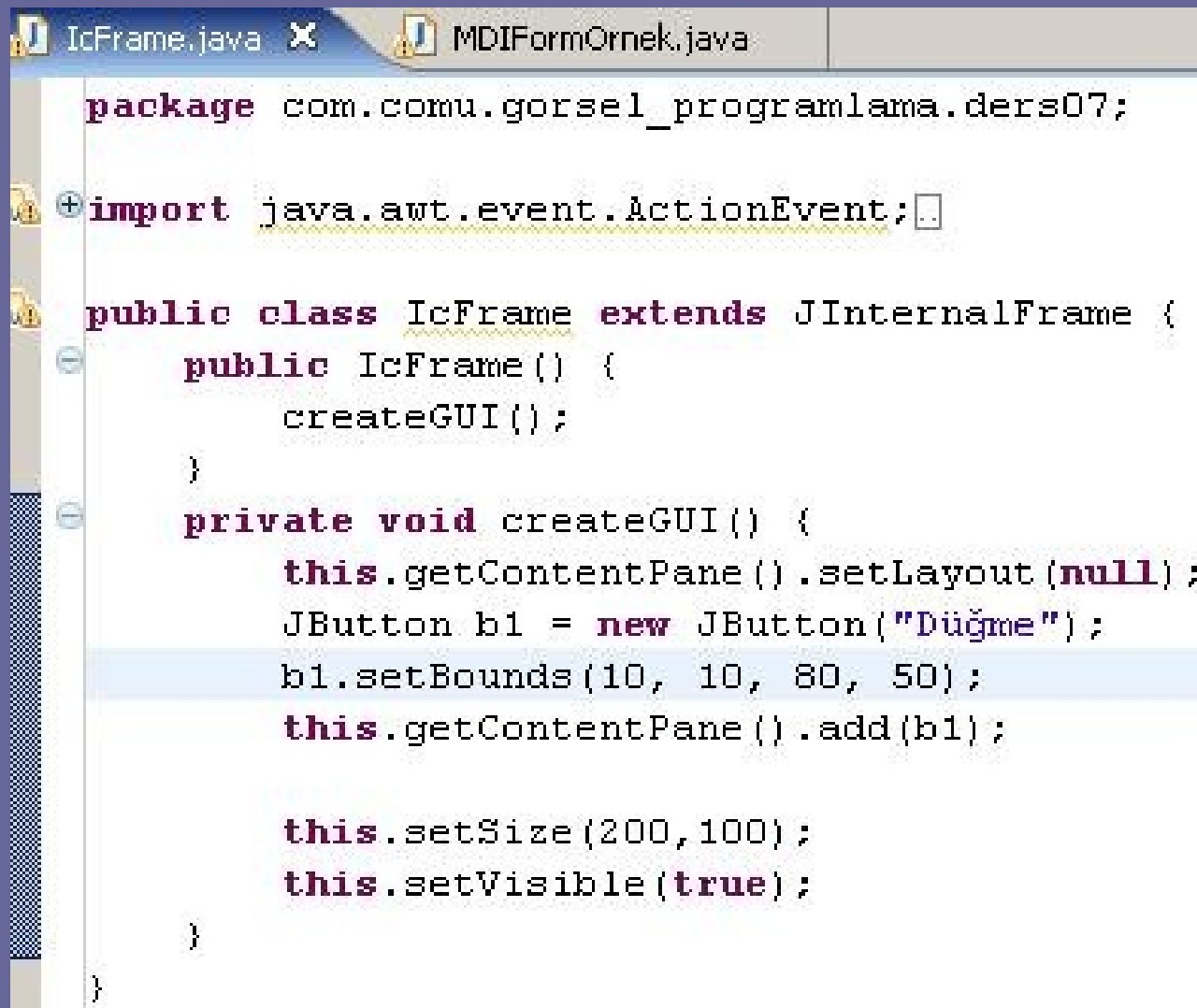
MDI (Multiple Document Interface)

Çoklu pencereler oluşturmak ve bu pencereleri bir ana pencere içinde göstermek için kullanılır.

İç çocuk pencereler **JInternalFrame** sınıfından türetilirler.

Ana pencere **JDesktopPane** sınıfından türetilir.

MDI (Multiple Document Interface)



```
package com.comu.gorsel_programlama.ders07;

import java.awt.event.ActionEvent;

public class IcFrame extends JInternalFrame {
    public IcFrame() {
        createGUI();
    }
    private void createGUI() {
        this.getContentPane().setLayout(null);
        JButton b1 = new JButton("Düğme");
        b1.setBounds(10, 10, 80, 50);
        this.getContentPane().add(b1);

        this.setSize(200, 100);
        this.setVisible(true);
    }
}
```

MDI (Multiple Document Interface)

```
MDIFormOrnek.java X
package com.comu.gorsel_programlama.ders07;

import javax.swing.JDesktopPane;
import javax.swing.JFrame;

public class MDIFormOrnek extends JFrame {
    public MDIFormOrnek() {
        createGUI();
    }

    private void createGUI() {
        JDesktopPane desktop = new JDesktopPane();
        this.setContentPane(desktop);
        JFrame f1 = new JFrame();
        f1.setIconifiable(true);
        f1.setMaximizable(true);
        f1.setResizable(true);
        f1.setClosable(true);
        desktop.add(f1);

        JFrame f2 = new JFrame();
        f1.setIconifiable(true);
        f2.setLocation(90,150);
        desktop.add(f2);
        f2.toFront();
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(500,300);
    }

    public static void main(String[] args) {
        MDIFormOrnek app = new MDIFormOrnek();
        app.setVisible(true);
    }
}
```


Görsel Programlama

DERS 07