Swing


```
import javax.swing.JFrame
```

Close Operation

JFrame.EXIT_ON_CLOSE : Pencerenin sağ üst kısmında (x) oluşur.

Java.lang.System.exit() metodunu çağırır.

JFrame.HIDE_ON_CLOSE : Frame gizlenir ve arka planda çalışır.

JFrame.DISPOSE_ON_CLOSE: Frame yok olur " " " " .

JFrame.DO_NOTHING_ON_CLOSE: Hiçbir işlem yapmaz.

JFrame Methodları

SetTitle(" "); Pence ismini ayarlar.

Set Size (int w, int h): Genişlik ve yükseklik pixel olarak ayarlar

Set Visible (): Bileşen görünürlüğü için

toBack() toFront(): Pencereyi en arka ve en öne atar.

add(): Parametre olarak girilen bileşeni ekler.

SetLocation (int x, int y): X, y kordinatlarına yerleştirilir.

SetBounds (int x, int y, int w, int h): Size + Location

SetLocationRelativeTo(null): Pencere ekranın ortasına yerleşir.

Bir alt bileşen için kullanıldığında null yerine bileşen ismi o ortada olur.

JFrame Olayları

KeyPressed Klavyeden herhangi bir tuşa basıldığında.

WindowActivated Pencere aktif hale geldimi yapılan

Window Opened " açıldığında yapılan

Window Closed " dispose ile kapatıldığında yapılan

Window Closing " kapatılırken yapılması istenen olay.

Mouse Clicked " üzerinde mouse ile tıklamada olacak olay.

JLABEL

Yazı yazmayı sağlayan Komponent. Genellikle açıklama veya başka bir Kontrolün ne işe yaradığını belirtmek için kullanılır.

SetFont(): Yazı tipi, büyüklüğü, kalınlığını ayarlar.

SetForeground(): Yazı rengini ayarlar.

SetText(): Yazının değiştirilmesini sağlar.

SetHorizantalTextPosition(): TRAILING-RIGHT-LEADING-CENTER-BOTTOM-TOP

SetAlignmentX() setAlignmentY(): Float türünden X-Y Kordinatları

SetAutoscrolls(): Metinde otomatik kaydırma yapılmasını sağlar.

SetBorder(): Bevel Compound Line Vb. türleri var.

SetEnabled Komponenti etkinleştirmek için kullanılır.

SetName Komponente isim vermek için kullanılır.

JButton

Herhangi bir işlem için butona basma olayı

ActionPerformed() Buton üzerinde herhangi bir işlem old. çağrılan fonk.

focusGained() focusLost Programda tab tuşuyla geçiş yapılmasıyla

JTextField

Tek hücreli bilgiler için kullanılan metin alanı

SetDisabledTextColor() Yazı rengi için. Enable özelliği false olursa yazı rengi sbt.

SetEditable() Komponent içine yazı yazmaya izin verir

SetFocusable() " arasında TAB geçişi olup olmaması

Private Void KeyTyped() Komponent içinde tuşa basıldığında çağrılan fonk.

JRadio Button

Birden fazla seçeneğin sadece birinin seçilmesi için kullanılır.

setSelected : Radio button düğmesinin seçili old. gösterir

buttonGroup RadioButtonların bağlı olacağı grubu belirler.

Private Void ActionPerformed: Fare üzerine tıklanıldığında işlem yapar.

Private Void StateChanged Komponent üzerine gelip ayrılması durum değişikliği

Private Void ItemStateChanged Komponentin seçim durumu değişirse çalışır.

ButtonGroup

Seçme işleri gerçekleştiren komponentleri gruplandırmak için kullanılır.

JCheckBox

Kullanıcı birden fazla seçim yapabilir ancak Radio Button gibi gruplanırsa yalnız 1 tane Checkbox işaretlenmesine izin verilir.

JComboBox

Aşağı doğru açılan liste. Değeri daha önceden belli olan elemanlar için.

setEditable: Komponent içine yazı yazmak için.

SetSelectedIndex Liste içinde seçilmiş elemanın index numarasını verir.
İlk elemanın index numarası "0" dır.

model komponente ait liste içinde gösterilecek elemanları belirlemek için kullanılır

addItem Sona eleman ekler.

insertItemAt index numarasına göre eleman ekler.

Private Void ItemStateChanged Liste üzerindeki eleman seçiminde değişiklik yapıldığında çağ. fonk.

→ String[] liste = {"A1", "A2", "A3"};
DefaultComboBoxModel myModel = new DefaultComboBoxModel(liste);
jcomboBox.setModel(myModel);

JList

Listeyi göstermek için kullanılan Komponent. CTRL birden çok, SHIFT ile 2 eleman arası seçilir.

SetSelectionMod(): Single → tek bir kayıt.
SINGLE_INTERVAL → tek bölümde birden çok kayıt.
MULTIPLE_INTERVAL → Birden çok bölümde birden çok kayıt.

getSelectedIndices: Liste içinden seçilmiş tüm elemanların pot. verir.

getSelectedValue: Seçilmiş elemanın değerini verir.

getSelectedValuesList: Listeden seçilmiş tüm elemanların değerini verir

addItem(" "): Listenin en sonuna eleman ekler.

insertItemAt(" ", index): index sıralı yere eleman ekler.

SetSelectedIndex: index numaralı elemanı default olarak seçer

SetSelectedItem: isimdeki elemanı " " " .

Private Void ValueChanged: Liste üzerindeki eleman seçiminde değişiklik yapıldığında çağrılan fonk.

JList (Model Veri Tutulması)

```
JList.setModel(myModel);
ListModel listModel;
DefaultListModel myModel;
Private Void modelGetir() {
    listModel = JList.getModel;
    MyModel = New DefaultListModel();
    for(i=0; i<listModel.getSize(i++)
    { myModel.addElement(listModel.getElementAt(i));
```

Açılışta boş olacaksa
myModel = New DefaultListModel();
jList.setModel(myModel);
DefaultListModel myModel;

JProgressBar

ilerleme çubuğu Uzun işlem yapılırken yapılan işlemin ilerleme durumu.

setMaximum - setMinimum ilerleme çubuğunun en büyük ve en küçük değ. belirlemek için

Orientation Progress bar yatay mı dikeymi olacağını belirler. (0 yatay 1 dikey)

setValue Progress bar başlangıçtaki durumu.

cursor. Progress bar üstüne gelindiğinde fare işaretçisinin görünümü.

setStringPainted() Progress bar üstünde ilerlemeye ait etiketin gösterilmesi

```
Private Void jButton1 ActionPerformed (...)
{ int min=0; int max=50; int val=0;
  jProgBar.setMinimum(min); jProgBar.setMaximum(max); jProgBar.setValue(val);

  for(int i=0 ; i<=50 ; i++) {
    try { Thread.sleep(50);
    }
    Catch (Exception ex) {
        System.out.println("Hata:" + ex.getMessage());
    }
    jProgBar.setValue(val+1); Val = jProgBar.getValue();
    JProgBar.setString("%" + 100* (val-min) / (max-min));
    JProgBar.update(jProgBar.getGraphics);
    JProgBar.setString("işlem tamam");
```

JFormattedTextField

Düzenlenmiş metin alanı. ↑ MaskFormatter olursa formatı bir belirleniz.

setFormattedFactory Metin kutusuna girecek olan bilginin formatını belirler.

getValue Metin kutusuna girecek olan bilginin değeri döner.

JPasswordField

Şifreli metin alanı

Sifre.setEchoChar('*');

SetEchoChar Bilginin hangi karakter ile gizleneceği belirtir.
SetEchoChar((char)0) olursa gizlenmez.

getPassword Şifre bilgisini alır.

```
String Karakter = String.ValueOf(jPassField.getEchoChar());
System.out.Print(JPassField.getText().replaceAll(".", Karakter));
```

JTable

2 boyutlu tablo için kullanılır.

SetModel Tablo yapısını satır sütun sayısı başlıkları, tabloya bilgi girişi olup olmayacağı

setCellSelectionEnabled Tek hücre seçimi yapılıp yapılmaması.

SetColumnSelectionAllowed Sütun ya da satırı komple seçip seçmeyeceğimiz

SetRowHeight Satır yüksekliği ayarlar.

ShowHorizontalLines ShowVerticalLines Yatay ve dikey ayraç çizgileri gösterilmesi

tableHeader Sütun sıralaması ve sütun uzunluğunun değişip değişmeyeceği

getValueAt(int satir int sütun) Satır ve sütundaki bilgiyi okur.

getColumnName (int sütun) Sütunun ismini verir.

insertRow(int satır, Object[] değerDizi) DegerDizi isimli dizideki elemanları ekler.

SetColumnIdentifiers(Object[] sutunIsim) Mevcut sutunlardaki ismi sutunIsim dizisi ile yer değişti

getSelectedRow - getSelectedColumn Seçili satır ve sütunun indeksini döndürün

```
String parola = new String(Sifre.getPassword());
    if(parola.equals("---")){ ---
```

```
DefaultTableModel model = tablo.getModel();
Object[] eklenecek{ ------ };   model.addRow(eklenecek);
```

```
tabloMouseClicked
{
  DefaultTableModel model = isrinTablosu.getModel();
  int secili = isrinTablosu.getSelectedRow();
      :                          getValueAt(secili
```

JTREE

Ağaç yapısı bilgileri listelemek için kullanılan komponent.

Ağaç verileri TreeModel kullanan bir modelde saklıdır

Ağaçtaki düğümleri göstere bilmek için TreeNode arayüzü kullanılır

DefaultMutableTreeNode genel olarak kullanılan bir ağaç düğümü.

setRootVisible Ana kökün görünüp görünmeyeceği

getRowCount Kök dahil kaç adet ana klasör başlığı old. verir.

isRoot Seçilen düğüm root ise true değilse false döner.

isLeaf " " Yaprak " " " " "

getSelectionCount Ağaçta kaç adet elemanın seçili old. verir.

getNewLeadSelectionPath Ağaçta Kökten seçilen düğüme kadarki yol sırasını alır

getLastSelectedPathComponent Ağaçta son seçilen elemanın yol bileşenini döndürür

insertNodeInto Ağaca yeni eleman ekleme işlemi yapar.

removeNodeFromParent Seçilen elemanı ağaçtan siler.

Private ValueChanged Seçilmiş değerde değişiklik old. zaman çağrılır

JMenuBar Menü Çubuğu

add Menü çubuğuna yeni menü başlığı ilave etmek için kullanılır.

JMenu Ana menü

setText Menü başlığı ismini ayarlamak için

getItemCount Menü başlığındaki alt elemanların sayısını verir.

JMenuItem Menü alt bileşeni

SetAccelerator Kısayol tuş kombinasyonu için KeyStroke()

SetIcon Menü başlığına ikon koymak için

JSeperator Ayraç

JPopUpMenu

SetComponentPopUpMenu Bileşeni aktif hale getirir.

JOptionPane (javax.swing.JOptionPane)

JOptionPane.showMessageDialog	Konsol mesajı görevi görür.
JOptionPane.showInputDialog	Girdi istenir ve mesaj gösterilir.
JOptionPane.showConfirmDialog	Soru için onay gerekir (Yes / No / Cansel)
JOptionPane.showOptionDialog	Seçeneklerden birini seçmesi istenir.

JDIALOG

JDialog için ekranda görünecek en küçük pencere boyutu belirtilmeli ve Pencerenin kapatılması için ~~yazılır~~ kod yazılmalı.

MinimumSize Açılacak pencerenin min. boyutları belirlenmeli

Bounds Boyut belirlemek için. (x y w H)

JFILECHOOSER

JFileChooser.CANCEL_OPTION → Kullanıcı iptal tuşuna basarsa

JFileChooser.APPROVE_OPTION → " OK AS " "

JFileChooser.ERROR_OPTION → " Komponent penc. kapatırsa

dialogType → Dosya penceresi türü (Open, Save, Custom)

fileSelectionMode → Açılan pencereden hangi tür eleman seçileceği belirlenir

MultiSelectionEnabled → Çoklu dosya seçilip seçilemeyeceği

currentDirectory → Başlangıçta geçerli klasör tanımı

JTextArea

Çok satırlı yazı alanı

append (String str) JTextArea'daki yazıya str nesnesindeki yazıyı ekler.

insert (String str, int pos) Pos değerinin gösterdiği konuma str nesnesini ekler

getLineCount Yazının satır sayısını döndürür.

lineWrap içerisindeki yazının satır sonuna geldiğinde alt satıra geçmesi

Dosyadan Karakter Okuma

```
FileReader fr = new FileReader("deneme.txt");
int karakter = fr.read();
BufferedReader br = new BufferedReader(fr);
String line = br.readLine();
```

Dosyaya Karakter Yazma

```
FileWriter fw = new FileWriter("deneme.txt", true);
fw.write("_____");
```

Dosyadan Byte Okuma

```
FileInputStream fs = new FileInputstream("deneme.txt");
InputStreamReader isr = new InputStreamReader(fs, Charset.forName("iso 8859
int okunanByte = isr.read();
BufferedReader br = new BufferedReader(isr);
String line = br.readLine();
```

Dosyaya Byte Yazma

```
FileOutputStream fos = new FileOutputStream("deneme.txt", true);
OutputStreamWriter wr = new OutputStreamWriter(fos, Charset.forName("iso-
wr.write("____");
```

Dosyaya Yeniden Karakter Yazma

```
FileWriter fw = new FileWriter("deneme.txt", false);
```

↳ Dosya daha önceden varsa, dosya içindekiler silinir ve üstüne yazılır.

Dosyaya Yeniden Byte Yazma

```
FileOutputStream fos = new FileOutputStream("deneme.txt", false);
```

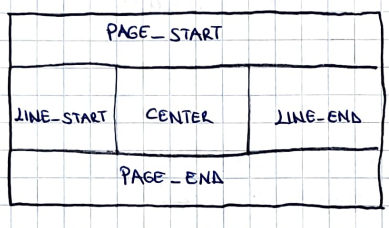
Dosyada Sonraki Satırı Yeni Satır Olarak Atarkana

```
FileWriter fw = new FileWriter("deneme.txt");
BufferedWriter bw = new (BufferedWriter(wr);
bw.newLine();
```

Görsel ①

revalidate → Düzenin yeniden hesaplanması repaint → Paneli yeniden çizer

BorderLayout



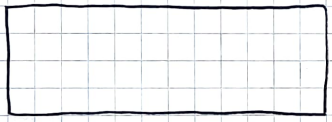
Oluşturulan Bileşen

```
JPanel panel = new JPanel();
panel.setLayout(new BorderLayout());
panel.add(button, BorderLayout.PAGE_START);
button.addActionListener(this);
add(panel);
```

Türetilen Frame

```
setLayout(new BorderLayout());
add(button, BorderLayout.PAGE_START);
button.addActionListener(this);
```

Flow Layout



içerdiği bileşenleri belirli bir yön boyunca konumlandırır.

- → Left
- → Right
- → Center
- → LEADING
- → Trailing

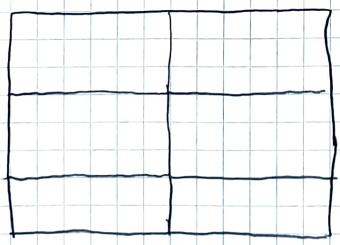
Oluşturulan Bileşen

```
FlowLayout myLayout = new FlowLayout();
    myLayout.setAlignment(FlowLayout.CENTER);
JPanel panel = new JPanel();
Panel.setLayout(myLayout);
Panel.add(button);
button.addActionListener(this);
add(panel);
```

Türetilen Frame

```
FlowLayout myLayout = new FlowLayout();
myLayout.setAlignment(FlowLayout.Center);
setLayout(myLayout);
add(button);
button.addActionListener(this);
```

Grid Layout



İçerdiği bileşenleri Satır ve Sütun sayısına göre hücrelere konumlandırır.

0 Verilirse değişebilir.

Oluşturulan Bileşen

```
JPanel panel = new JPanel();
panel.setLayout(new GridLayout(3,2));
panel.add(button);
button.addActionListener(this);
add(panel);
```

Türetilen Frame

```
setLayout(new GridLayout(3,2));
add(button);
button.addActionListener(this);
```

Dinamik Bileşen

→ Frame üzerindeki Label ve Checkbox ları değiştir

```
Private Void JButton ActionPerformed (Java.awt.event. ActionEvent evt){
    Component [] C = this. get ContentPane (). get Components ();
    for(int i=0 ; i<C.length ; i++) {
        if (c [i] instanceof JLabel) {      JLabel tmp = (JLabel) C [i];
                                            tmp . setText (" Label ") ;
                                        }
        else if ( C [i]. getClass () == JCheckBox. getClass()) { JCheckBox cb = (JCheckBox)C[i];
                                                                 cb. SetText (" Check box");
    }                                                          }
}
```

→ name özelliğindeki textfield içeriğini değiştirin

```
private Void JButton2 Action Performed (---) {
    Component [] c  = JPanel2. get Components ();
    for( int i=0;  i<c.length; i++) {
        if( c[i]. getName != NULL)
        if( c [i] instanceof JTextField && c[i]. getName (). equals ("ad")) {
            JTextField tmp = (JTextField) c [i];
            tmp. setText (" ---- ");
        }
    }
}
```

→ Panel Temizleyip üstüne Dinamik Label Ekler

```
Private Void JButton3 Action Performed (--) {
    JPanel3 Remove All();      JPanel3 Set PreferredSize (JPanel3. getSize); (Mevcut boyut Korunuyor
    JPanel3. set Layout (new Grid Layout (2,1));
    for(int sayi=0; sayi<2; sayi ++) {
        JLabel tmp = new JLabel ("Dinamik");
        tmp. setName ("Dnmk" + sayi);
        JPanel3. add (tmp);
    }
    JPanel3. Validate ;   JPanel3 repaint ;
    this repaint ; (frame için);
```

Görsel ②

```
Public Connection GetConnection (){

    Connection con ;
    Con = null;
try {
        Class. forName ("net. Ucanaccess. jdbc. UcanaccessDriver");
        String strdatabase = " Veritabanı ismi

        Con = DriverManager. getConnection (strdatabase);
        System. out. println (" Con );
    }
 Catch (Exception e)  System. out. println (" bağlantı hatası ");

    return con;
}

Private Void ekle ActionPerformed (__) {

  Connection Con = GetConnection();

  String ad , yas ;

  ad = JTextField. GetText ;    yas = JTextfield GetText ;

  String sorgu = " INSERT INTO KISI (AD, YAS) VALUES (' " + ad + " ', " + yas + ")";

try {
      Statement stmt = Con. createStatement ():

      Stmt . execute (Sorgu);

      Stmt. close();   Con. close ();
   }
Catch (Exception e)  System.out. println(" Eklenmedi ");

private Void Sil Actton Performed (__) {

  Connection Con = GetConnection ();
  String sorgu = " DELETE FROM KISI WHERE AD = ' " + ad + " ' " ;

try {
      Statement stmt = Con. createStatement ();
      Stmt . execute (sorgu);
```