

# NESNE YÖNELİMLİ PROGRAMLAMA(Object Oriented Programming/OOP)

**Öğr. Gör. Celil ÖZTÜRK**

Marmara Üniversitesi

Teknik Bilimler Meslek Yüksekokulu

# İçerik

- ✓ Final Kelimesi
- ✓ New Kelimesi
- ✓ Nokta (.) Operatörü
- ✓ Swing Kütüphanesi

# Final Kelimesi

Final anahtar kelimesinin kalıtımda **iki farklı** kullanımı var.

- Final kelimesi sınıf tanımlamasında kullanılır ise, sınıf üzerinden kalıtıma izin verilmez.
- Final ile tanımlanmış sınıflardan yeni sınıf türetilemez.
- **Extends** anahtar kelimesi kullanılamaz.

# Final Kelimesesi

```
public final class Ogrenci {  
    // içerik  
}
```

```
package javaapplication12;
```

```
/**
```

```
 *
```

```
 * @author c3111
```

```
 */
```

```
public class OgrenciBilgi extends Ogrenci {
```

```
}
```

cannot inherit from final Ogrenci

----

(Alt-Enter shows hints)

# Final Kelimesi

- Diğer bir kullanım, bir metot final ile oluşturulmuş ise, @Override ile ezilemez, devre dışı bırakılamaz.

```
public class Kisi {  
    public final void KisiBilgisi()  
    {  
        System.out.println("bu metot final kelimesi ile tanımlanmıştır.");  
    }  
}
```

```
/**  
 *  
 * @author c3ll  
 */  
public class OgrenciBilgi {  
    @Override  
    public final void KisiBilgisi()  
    {  
        System.out.println("bu metot final kelimesi ile tanımlanmıştır.");  
    }  
}
```

KisiBilgisi() in OgrenciBilgi cannot override KisiBilgisi() in Kisi  
overridden method is final  
(Alt-Enter shows hints)

# New kelimesi

- New operatörü sınıfı adıyla beraber kullanılır.
- Bellekte ilgili sınıfa ait yeni bir nesne oluşturur.
- Öğrenci öğrenci1 = **new** Öğrenci();
- Öğrenci1 nesnesi için bellekte yer ayrılması sağlandı.
- Bir nesne, new operatörü ile oluşturulmamış ise varsayılan değeri null olur.

# Nokta (.) Operatörü

- Bir nesne oluşturulduktan sonra, public ve protected erişim belirteci kullanılan üye değişken veya metotlara dışarıdan erişim sağlayabilmek için kullanılır.
- Öğrenci öğrenci1 = new Öğrenci();
- Öğrenci1.kisiBilgisi();
- Öğrenci1.ad="Ali";

# Swing Kütüphanesi

- javax.Swing paketi...
- JFC(Java Foundation Classes) altında tanımlanmıştır.
- Java.AWT (Java Abstract Window Toolkit) üzerine kuruludur.(AWT çalışma anında işletim sistemi bileşenlerini kullanır.)
- GUI(Grafiksel Kullanıcı Arayüzü) oluşturmak için tercih edilir.
- Java masaüstü uygulama geliştirmeye olanak verir.



# Swing Kütüphanesi

- Bir Görsel arayüz oluşturulurken, ilk olarak bir çerçeveye ihtiyaç vardır.
- JFrame ile GUI için bir çerçeve oluşturulur.
- Diğer bileşenler JFrame çerçevesi üzerine yetleştirilir.
- `import java.awt.*;`
- `import javax.swing.*;`

<https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>



# Swing Bileşenleri(JFrame)

- JFrame – Frame çerçeve anlamına gelir. Boyutları değişebilen bir bileşendir.
- Genellikle Panelleri ve diğer bileşenleri içerir.
- Her zaman en altta yer alır.

```
JFrame cerceve = new JFrame("JFrame Çerçevesi");
```

# Swing Bileşenleri(JFrame)

- Yeni bir JFrame nesnesi oluşturma.
- `JFrame cerceve = new JFrame("JFrame Çerçevesi");`
- `setSize()` oluşturulan `cerceve` nesnesinin boyutunu ayarlar.
- `cerceve.setSize(200, 200);`
- `setLocation()` oluşturulan `cerceve` nesnesinin koordinat olarak konumu için kullanılır. `Default(0,0)` olarak konumlanır.

# Swing Bileşenleri(JFrame)

- setVisible(boolean) metodu ise cerceve nesnesinin görünürlüğünü belirler. True değerini alır ise nesne görünür olur.
- `cerceve.setVisible(true);`
- cerceve.setTitle()metodu uygulama başlık bilgisi verilir.
- `cerceve.setTitle("Uygulama başlık bilgisi")`

# Swing Bileşenleri(JFrame) Örnek

```
public class Gui1 {  
    public static void main(String[] args) {  
        JFrame f = new JFrame("JFrame Çerçevesi");  
        f.setSize(400, 400);  
        f.setLocation(200, 200); //X Y eksenini olarak konumu gibi düşünülebilir..  
        f.setTitle("Test başlık bilgisi");  
        f.setVisible(true);  
    }  
}
```

# Swing Bileşenleri(JButton)

- JButton ile Frame üzerinde eklenebilen buton nesneleri oluşturulabilir.
- `JButton buton1 = new JButton("Buton1");`
- Oluşturulan buton nesnesi daha sonra Frame(çerçeve) nesnesine eklenmelidir.
- `Cerceve.add(buton1)`
- Daha sonra mutlaka eklenen bileşenler `setVisible (true)` yapılmalıdır.

# Swing Bileşenleri(JButton)

```
public class Gui1 {  
    public static void main(String[] args) {  
        JFrame f = new JFrame("JFrame Çerçevesi");  
        f.setSize(400, 400);  
        f.setLayout(new FlowLayout());  
        JButton buton1 = new JButton("Test1");  
        JButton buton2=new JButton("Test2");  
        f.add(buton1);  
        f.add(buton2);  
        f.setVisible(true);  
    }  
}
```

# Swing Bileşenleri(JButton ActionListener)

- addActionListener() metodu ile bir butona actionListener eklenebilir.
- ActionListener sayesinde o buton tıklanma özelliğini almış olur.

```
buton1.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent ae) {  
        buton2.setVisible(false); //buton2 nesnesi gizlendi.  
    }  
});
```



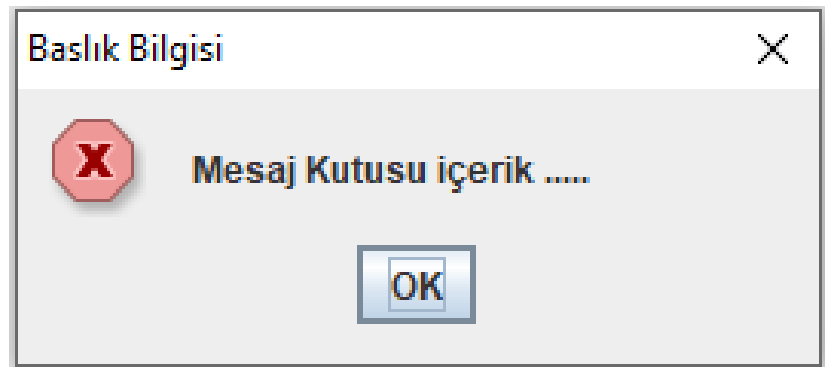
# Swing Bileşenleri(JButton ActionListener)

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Gui1 {
    public static void main(String[] args) {
        JFrame f = new JFrame("JFrame Çerçevesi");
        f.setSize(400, 400);
        f.setLayout(new FlowLayout());
        JButton buton1 = new JButton("Test1");
        JButton buton2=new JButton("Test2");
        f.add(buton1);
        f.add(buton2);
        f.setVisible(true);
        buton1.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) { buton2.setVisible(false); }
        });
    }
}
```

# Swing Bileşenleri(MessageBox/MessageDialog)

- JOptionPane.showMessageDialog(Jframe\_nesnesi,"text içeriği")
- JOptionPane.showMessageDialog(f,"Mesaj Kutusu içeriği .....", "Baslık Bilgisi",JOptionPane.ERROR\_MESSAGE);}



# Swing Bileşenleri(JTextField)

- Text alanı olarak nesne oluşturulmak için kullanılır.
- `JTextField textfield= new JTextField();`
- `setBounds(x, y, width, height)`

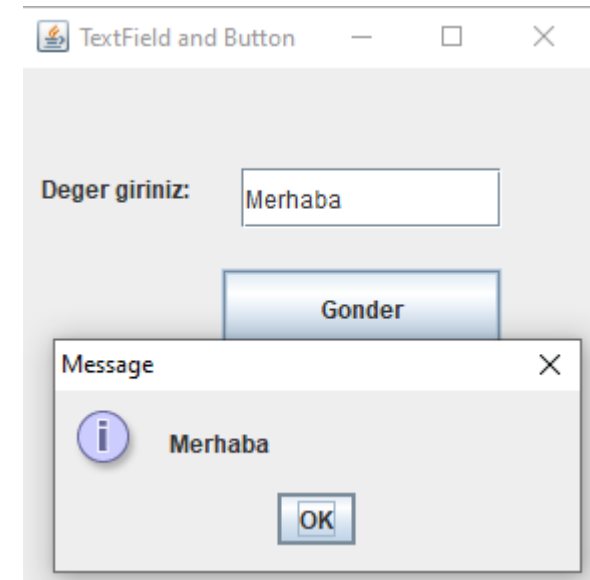
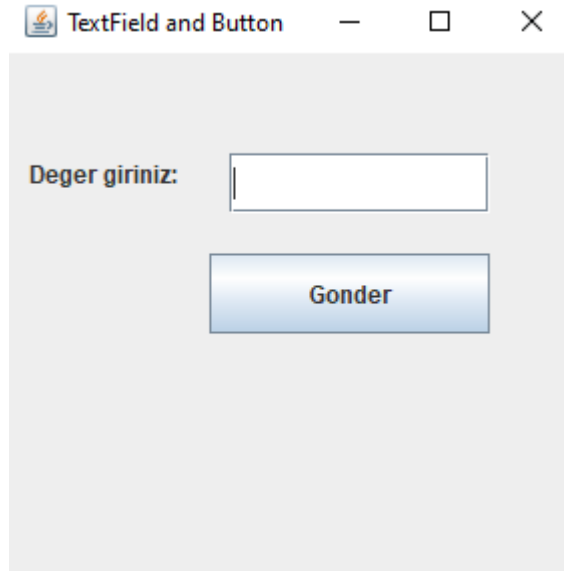
# Swing Bileşenleri(JLabel)

- JLabel label1=new JLabel()
- setBounds(x, y, width, height)
- \*\*\*label1.setText("Name has been submitted.");

# JLabel, JTextField Örnek

```
package gui1;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
public class Gui1 {
    public static void main(String[] args) {
        JFrame f=new JFrame("TextField and Button");           //submit button
        JButton b=new JButton("Gonder");
        b.setBounds(100,100,140, 40);
        JTextField textfield= new JTextField();
        textfield.setBounds(110, 50, 130, 30);
        JLabel label1=new JLabel("Deger giriniz:");
        label1.setBounds(10, 10, 200, 100);
        f.add(label1);
        f.add(textfield);
        f.add(b);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

# Örnek Uygulama



## JLabel, JTextField Örnek

```
buton1.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent ae)  
    {  
        JOptionPane.showMessageDialog(f, textfield.getText());  
    }  
});
```

# 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>