



Java'da Applet'ler (...devam) ve AWT ve SWING Kütüphaneleri

BBS-515 Nesneye Yönelik Programlama

Ders #8 (9 Aralık 2009)

■ Geçen ders:

- ▶ Java'da İstemci Tarafı Uygulamalar ("Applets")
- ▶ Örnekler

■ Bu ders:

- ▶ Java Applet'lerde bileşen yerleştirme türleri ("applet layouts")
- ▶ Java'da "Awt" ve "Swing" kütüphane bileşenleri
- ▶ Örnekler

Java Applet'lerde Bileşen Yerleştirme Türleri - 1

■ “Flow Layout” – varsayılan

- ▶ Bileşenleri, metin editörde sözcüklerin dizilmesi gibi yerleştirir.
- ▶ Bileşenler soldan sağa ve yukarıdan aşağıya yerleşir.
- ▶ Her satır; sol dayalı (“LEFT”), merkezde (“CENTER”) veya sağa dayalı (“RIGHT”) olarak yerleşebilir. – varsayılan: “CENTER”
- ▶ Örnek:
 - ◆ FlowYerlesDemo.java (Applet)

Java Applet'lerde Bileşen Yerleştirme Türleri - 2

■ “Border Layout”

- ▶ Genel pencere tasarımı için 5 temel alan tanımlar.
 - ◆ “NORTH” – kuzey (en üst) alan
 - ◆ “SOUTH” – güney (en alt) alan
 - ◆ “WEST” – batı (en sol) alanı
 - ◆ “EAST” – doğu (en sağ) alanı
 - ◆ “CENTER” – merkezdeki alan

- ▶ Örnek:
 - ◆ BorderYerlesDemo.java (Applet)

Java Applet'lerde Bileşen Yerleştirme Türleri - 3

■ “Grid Layout”

- ▶ Bileşenlerin grid (tablo) şeklinde yerleştirilmesini sağlar.
- ▶ Tablonun satır ve sütun sayısı önceden belirlenir.
- ▶ Örnek:
 - ◆ GridYerlesDemo.java (Applet)

Java Applet'lerde Bileşen Yerleştirme Türleri - 4

■ “Card Layout”

- ▶ Farklı yerleştirme türlerini, bir anda bir tanesi görünmek üzere, kombine eder.
- ▶ Örnek:
 - ◆ KartYerles.java (Applet)

■ “Mixed Layout”

- ▶ Farklı yerleştirme türlerini, aynı anda görünmek üzere, kombine eder.
- ▶ Örnek:
 - ◆ KarisikYerles.java (Applet)

Java'da “AWT” ve “Swing” Kütüphaneleri

- **AWT** (“Abstract Window Toolkit”), Java’nın platform-bağımsız pencere, grafik ve kullanıcı-arayüzü araç-setidir.
 - ▶ AWT, J2SE 1.2 ile birlikte, **Swing** araç-seti altına taşınmıştır.
 - ▶ Paket: java.awt

- **Swing**, Java SE platformu için grafik kullanıcı arayüzü kütüphanesi olarak tanımlanmaktadır.
 - ▶ AWT’ye göre daha gelişmiş grafik kullanıcı arayüzü bileşenlerini sağlar.
 - ▶ Paket: javax.swing

Java “Awt” Bileşenleri

■ Component

- ▶ Button
- ▶ CheckBox
- ▶ Choice
- ▶ List
- ▶ Container ~ JComponent (“swing”de)
 - ◆ Panel
 - Applet ~ JApplet (“swing”de)
 - ◆ ScrollPane
 - ◆ Window ~ JWindow (“swing”de)
 - Dialog ~ JDialog (“swing”de)
 - Frame ~ JFrame (“swing”de)
- ▶ Canvas
- ▶ Label
- ▶ TextComponent
 - ◆ TextField
 - ◆ TextArea
- ▶ ScrollBar

Java “Swing” Bileşenleri

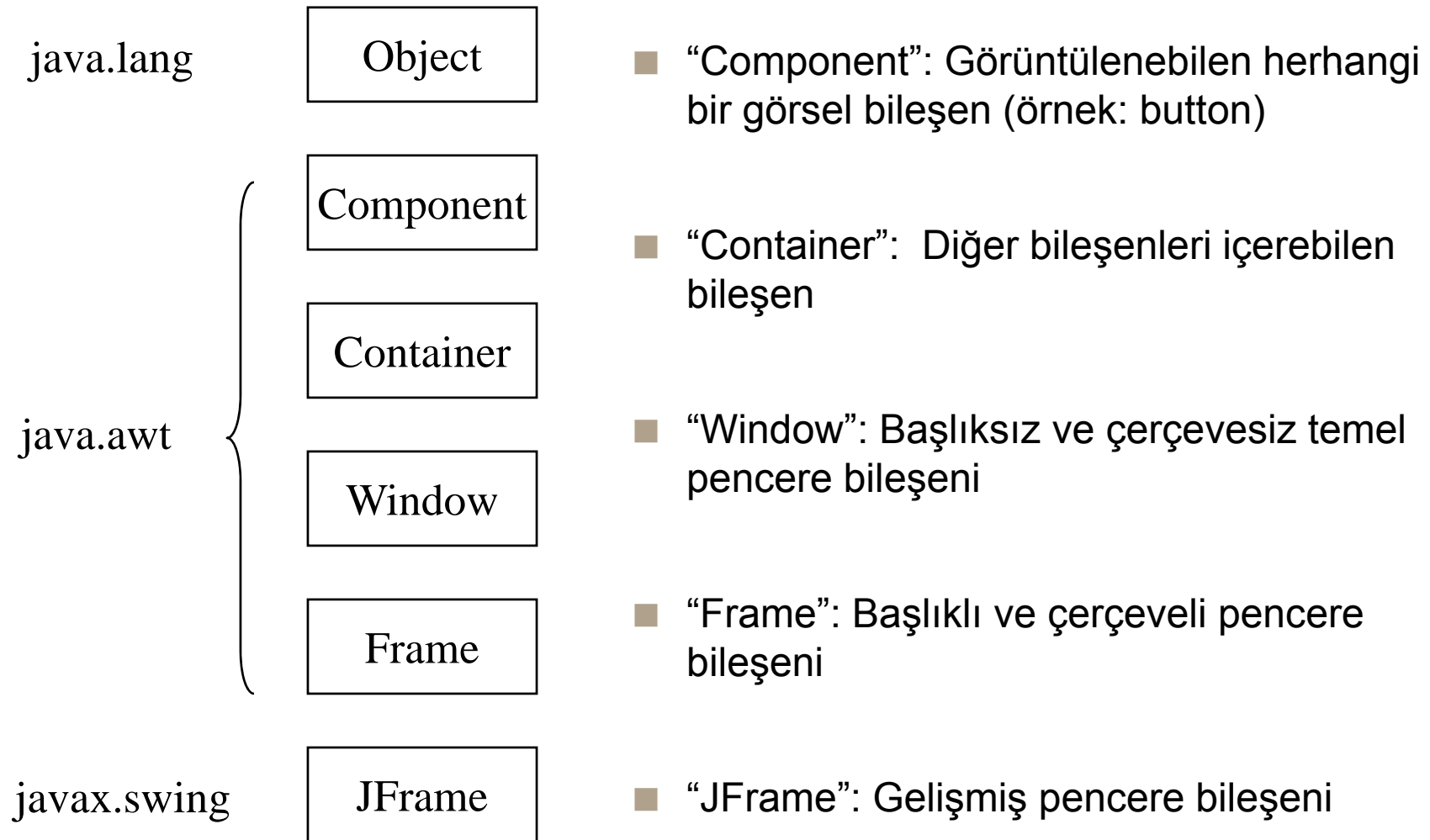
■ JComponent

- ▶ JLabel
- ▶ JPanel
- ▶ JMenuBar
- ▶ AbstractButton
 - ◆ JMenuItem
 - JMenu
 - JRadioButtonMenuItem
 - JCheckBoxMenuItem
 - ◆ JButton
 - ◆ JToggleButton
 - JCheckBox
 - JRadioButton
- ▶ JPopupMenu
- ▶ JSeparator
- ▶ JComboBox
- ▶ JList
- ▶ JTextComponent
 - ◆ JTextField – JPasswordField
 - ◆ JTextArea
 - ◆ JEditorPane – JTextPane

■ JComponent (devamı ...)

- ▶ JOptionPane
- ▶ JScrollPane
- ▶ JSplitPane
- ▶ JTabbedPane
- ▶ JInternalFrame
- ▶ JRootPane
- ▶ JLayeredPane – JDesktopPane
- ▶ JScrollBar
- ▶ JSlider
- ▶ JProgressBar
- ▶ JToolBar
- ▶ JToolTip
- ▶ JFileChooser
- ▶ JColorChooser
- ▶ JTable
- ▶ JTree

En Sık Kullanılan Bileşenler – Sınıf Hiyerarşisi



Pencere Oluşturmak

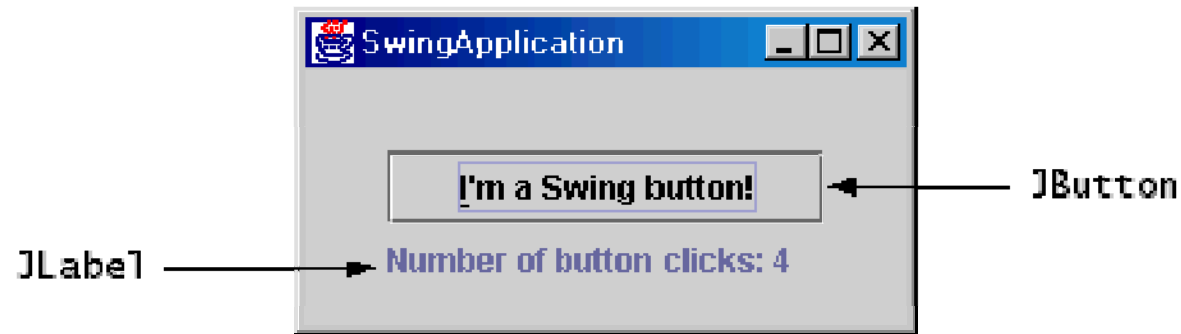
- JFrame sınıfından bir pencere nesnesi yarat.
- Pencerenin boyutlarını ayarla.
- Pencereyi görüntüle.
- Örnek:

```
JFrame birPencere = new JFrame("Başlık");  
birPencere.setBounds(50,100,400,150); // (x,y,w,h)  
birPencere.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
//DO_NOTHING_ON_CLOSE, DISPOSE_ON_CLOSE, HIDE_ON_CLOSE  
birPencere.setVisible(true);
```

“Component” ve “Container”

- “Component”, ekranda görüntülenebilecek grafik bir bileşendir.
- “getContentPane()” yöntemi, tipi “Container” olan bir çerçeve döndürür.
- JFrame sınıfından oluşturulmuş bir çerçeveye (“container”) bir bileşen (“component”) eklemek için çerçevenin “add(...)” yöntemi kullanılır.

Bileşen Ekleme: Örnek - 1



```
public class JCompDemo extends JFrame implements ActionListener {  
    public JCompDemo () {  
        Container c = this.getContentPane();  
        ...  
        JButton btn = new JButton("I'm a Swing button!");  
        btn.addActionListener ( this );  
        c.add ( btn );  
        ...  
    }  
}
```

Olayları İşlemek (“Event Handling”) - 1

- Kullanıcı bir karakter yazdığı zaman veya bir düğmeye bastığında, sistem bunlar için olaylar (“events”) üretir.
- Nesneler bu olaylardan haberdar edilirler.
- Her nesne kendi olayına göre olay dinleyici olarak kendini tanıtmakla ve ilgili arayüzü tanımlamakla yükümlüdür.

Olayları İşlemek (“Event Handling”) - 2

■ Her olay işleyici (“event handler”) üç parça kod içerir:

1. Sınıf bir dinleyici (“Listener”) arayüzünü gerçekleştirir.

◆ `public class SinifAdi implements ActionListener {`

2. Olay işleyici sınıfının bir örneği, bir veya daha fazla bileşen üzerinde dinleyici (“listener”) olarak kaydedilir.

◆ `bilesenAdi.addActionListener (this) // “this” veya “SinifAdi” sınıfının bir örneği`

3. Olay işleyici sınıfın içinde, dinleyici (“Listener”) arayüzünün yöntemleri gerçekleştirilir.

◆ `public void actionPerformed (ActionEvent e) { ... }`

Olayları İşlemek (“Event Handling”): Örnek

- Uzun uzun tanımlanabileceği gibi, anonim nesneler yaratarak birkaç satırda da tanımlanabilir.

- Örnek:

```
button.addActionListener(  
    new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            numClicks++;  
            label.setText("Number of clicks is:" + numClicks); }  
    }  
);
```



- JButtonDemo.java
- JCompDemo.java
- AWTOrnek.java

Olay İşleyiciler (“Event Handlers”)

Olaylar ve İlişkili Olay Dinleyiciler: Örnekler	
Olaya sebep olan davranış	Dinleyici tipi
Kullanıcı bir düğmeye basar, metin yazarken “enter” a basar veya bir menü öğesini seçerse	ActionListener
Kullanıcı ana pencereyi kapatırsa	WindowListener
Kursör bir bileşenin üstünde iken kullanıcı farenin bir tuşuna basarsa	MouseListener
Kullanıcı fareyi bileşenin üstünde gezdirirse	MouseMotionListener
Bileşen görünür yapılırsa	ComponentListener
Bileşen klavyenin odağına gelirse	FocusListener
Tablo veya liste seçim değişiklikleri yapıldıysa	ListSelectionListener

Dinleyici (“Listener”) Örnekleri - 1

■ “MouseListener”

- ▶ Fareyi dinlemeyi ve farenin ürettiği olayları yakalamayı sağlar.
- ▶ Bu arayüzü gerçekleştiren sınıfların, aşağıdaki yöntemleri tanımlaması zorunludur:
 - ◆ `public void mouseClicked (MouseEvent me) {}`
 - ◆ `public void mouseEntered (MouseEvent me) {}`
 - ◆ `public void mousePressed (MouseEvent me) {}`
 - ◆ `public void mouseReleased (MouseEvent me) {}`
 - ◆ `public void mouseExited (MouseEvent me) {}`
- ▶ Örnek:
 - ◆ `FareTiklaOrnek.java`

Dinleyici (“Listener”) Örnekleri - 2

■ “MouseMotionListener”

- ▶ Farenin hareketlerini dinlemeyi sağlar.
- ▶ Bu arayüzü gerçekleştiren sınıfların, aşağıdaki yöntemleri tanımlaması zorunludur:
 - ◆ `public void mouseMoved (MouseEvent me) {}`
 - ◆ `public void mouseDragged (MouseEvent me) {}`
- ▶ Örnek:
 - ◆ `FareHareketOrnek.java`

Ödev-4

- Basit bir kullanıcı arayüzü içeren bir uygulama programı yazın. Kullanıcı arayüzü aşağıdaki öğeleri içermelidir:
 - ▶ Bir JTextField ögesi
 - ▶ Bir JTextArea ögesi
 - ▶ İki tane JButton ögesi; birinin adı KUCUKHARF, ikincisinin adı BUYUKHARF olsun.
- Programınız JTextField alanından girilecek bir satırı (o bölgede **enter** tuşuna basıldığında) almalı ve bu satırı JTextArea içinde bulunan karakter dizisinin sonuna yeni bir satır olarak eklemelidir. Böylece JTextField bölgesinden gireceğiniz bütün satırlar JTextArea bölgesinde gözükecektir.
- JTextField bölgesinden bir satır okunduktan sonra o bölge temizlenmelidir.
- Kullanıcı BUYUKHARF düğmesine bastığında, programınız JTextArea bölgesindeki karakter dizisindeki bütün harfleri büyük harflere çevirerek o bölgede yeniden göstermelidir. KUCUKHARF düğmesine bastığında ise karakter dizisindeki bütün harfler küçük harflere çevrilerek yeniden JTextArea bölgesinde gösterilmelidir.
- Teslim Tarihi: 22.Aralık.2009 24:00, atarhan@hacettepe.edu.tr adresine e-posta ile.