

Java Appletlerine Giriş

- Applet
 - Program aşağıdaki programların birisiyle çalışır.
 - `appletviewer` (appletler için test aracı)
 - Web browser (IE, Communicator)
 - Applet içeren HTML (Hypertext Markup Language) dökümanlarını çalıştırır.

Java 2 Software Development Kit'den Birkaç Applet Örneği

- Örnek Appletler
 - Java 2 Software Development Kit (J2SDK) içinde gelen örnekelere bakacağız.
 - Kaynak kodlar (`.java` dosyası)
 - Yeni fikirlerin akla gelmesi açısından önemli
 - J2SDK demo kalsörü içinde birçok örnek görülebilir.
 - J2SDK aşağıdaki adresten indirilebilir.
java.sun.com/j2se/1.4.1/

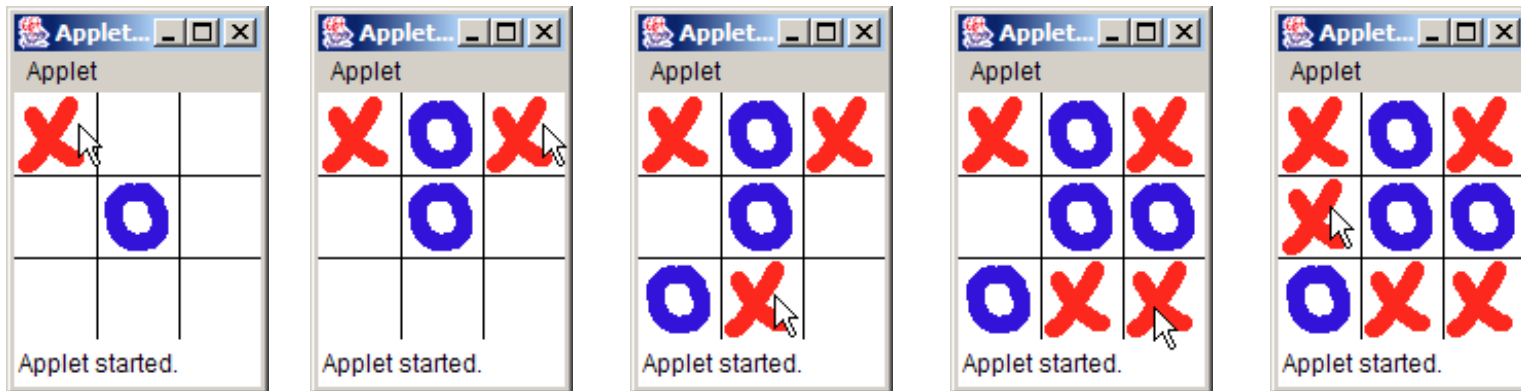
Java 2 Software Development Kit'den Birkaç Applet Örneği

- Appletlerin çalışması
 - Komut isteminden appletinizin bulunduğu yere gelin.
`c:\j2sdk1.4.1\demo\applets`
`cd appletDirectoryName`
 - Appleti çalıştıracak bir HTML dosyası olacak.
 - Yazın “`appletviewer example1.html`”
 - `appletviewer` html dosyayı yükler.
 - HTML dosyasından hangi appletin çalışacağına karar verilir.
 - Applet çalışır, **Reload** ve **Quit** komutları **Applet** menüsünün altında bulunur.

Java 2 Software Development Kit'den Birkaç Applet Örneği

- player "X" olarak oyuna başlıyorsunuz.

Fig. 3.2 Sample execution of applet TicTacToe.



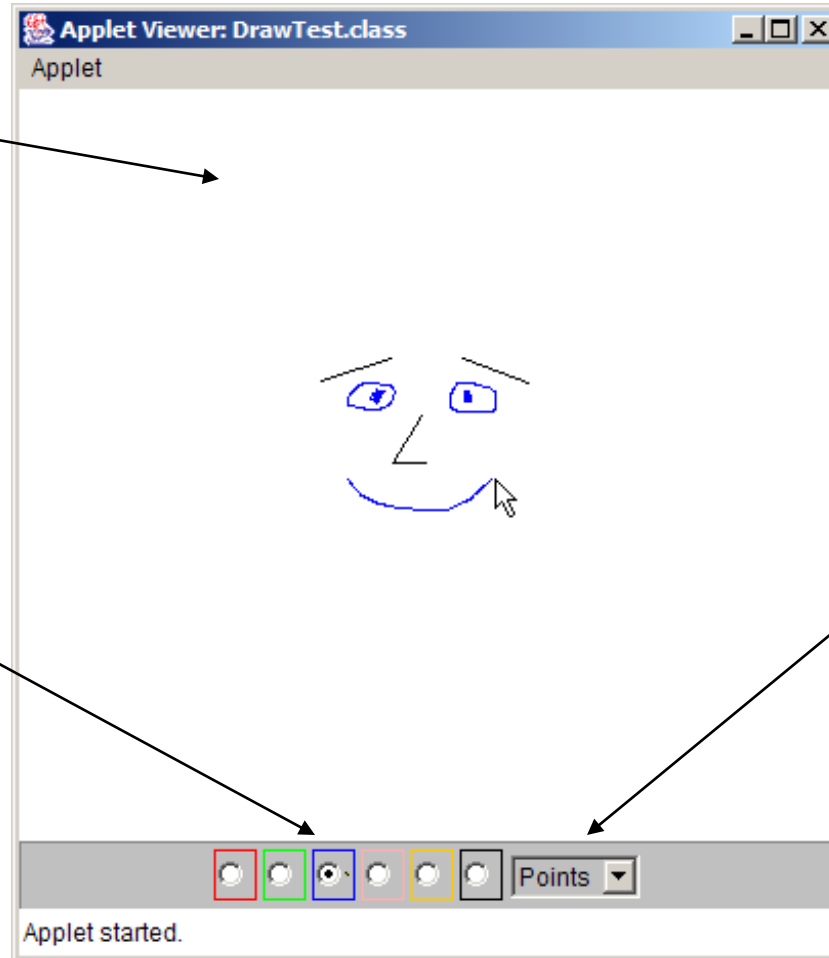
Java 2 Software Development Kit'den Birkaç Applet Örneği

Fig. 3.4 Sample execution of applet DrawTest.

Mouse hareketleri ile
şekillerin çizildiği alan.

Tıklanılarak istediğiniz
renk seçilir. Bu GUI
componentleri *radio*
buttons olarak bilinir.

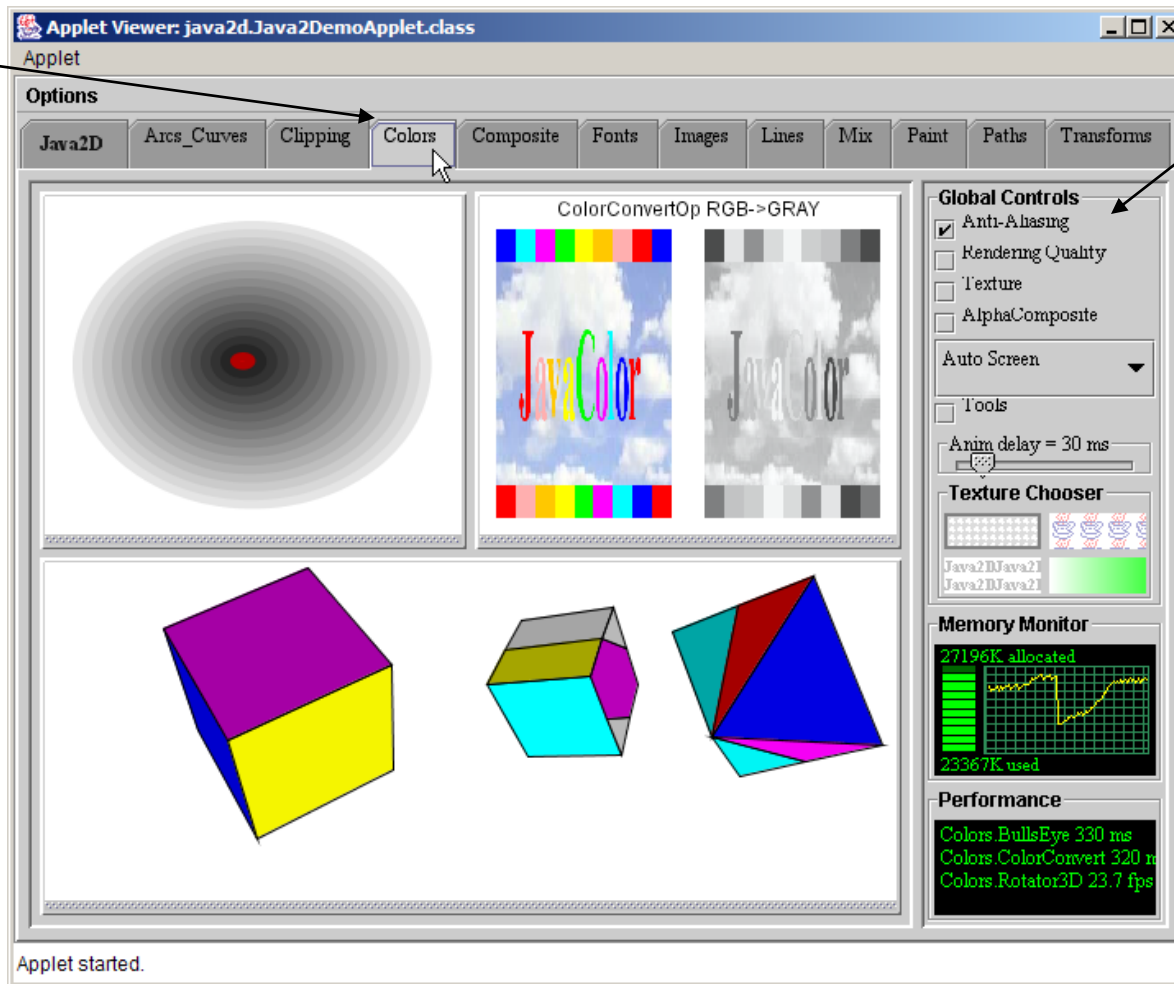
Component tıklanır
Lines veya **Points** seçilir.
Bu GUI component
combo box, *choice* veya
drop-down list olarak
bilinir.



Java 2 Software Development Kit'den Birkaç Applet Örneği

- Demonstrates 2D drawing capabilities built into Java2

Click a tab to select a two-dimensional graphics demo.



Try changing the options to see their effect on the demonstration.

Appletler

- Bir Applet bir Java programı ile etkileşime izin veren Panel'dir.
- Bir Applet genellikle bir Web sayfasında gömülür ve tarayıcıdan çalıştırılabilir.
- Appleti web sayfasında yerleştirmek için özel HTML komutlarına gerek vardır.
- Güvenlik sebeplerinden dolayı özel bir kutu içinde çalıştırılırlar.
- İstemci dosya sistemi ile erişim olmaz.



Applet'in hiyerarşisi

java.lang.Object

|

+----java.awt.Component

|

+----java.awt.Container

|

+----java.awt.Panel

|

+----java.applet.Applet



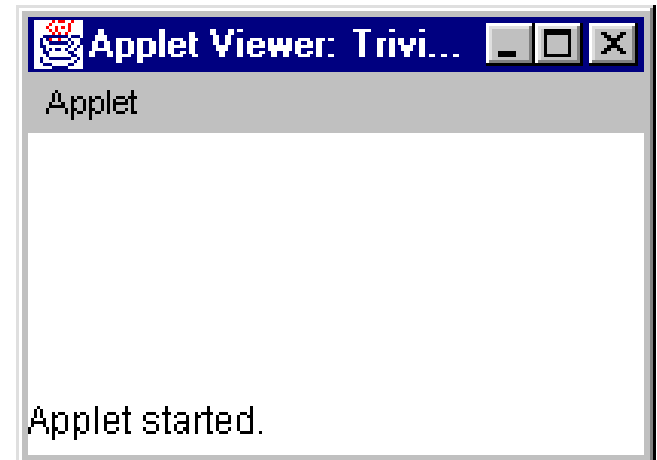
En basit applet örneği

TrivialApplet.java

```
import java.applet.Applet;  
  
public class TrivialApplet extends Applet { }
```

TrivialApplet.html

```
<applet  
  code="TrivialApplet.class"  
  width=150 height=100>  
  
</applet>
```



En basit applet örneği

```
import java.awt.*;  
import java.applet.Applet;  
  
public class HelloWorld extends Applet {  
    public void paint( Graphics g ) {  
        g.drawString( "Hello World!", 30, 30 );  
    }  
}
```



Applet Metotları

`public void init ()`

`public void start ()`

`public void stop ()`

`public void destroy ()`

`public void paint (Graphics)`

Also:

`public void repaint()`

`public void update (Graphics)`

`public void showStatus(String)`

`public String getParameter(String)`



Applet metotları

- Bir applet oluşturmak için JApplet(swing) veya Applet (Awt) sınıfından extend etmek gerekir.
- Applet init(), start(), stop(), paint(Graphics), destroy() metotlarını tanımlar.
- Bu metotlar bir şey yapmaz.
- Yapmaları gereken işi sizin doldurmanız gerekir.



public void init ()

- Çalıştırılacak ilk metottur.
- Değişkenleri başlatmak için ideal bir yerdir.
- GUI bileşenlerini tanımlamak, yerleştirmek ve onların yapacağı işi belirleyen listener'ları tanımlamak için idealdir.
- Hemen hemen her yazdığınız Applet init() metoduna sahiptir.



public void start ()

- Her zaman gerekli değildir.
- Init()’ten sonra çağrılır.
- Her sayfa yüklendiğinde ve yeniden başlatıldığında çağrılır.
- stop() ile birleştirilerek kullanılır.
- start() ve stop() Applet zaman tüketen hesaplamalar yaparken kullanılır.



public void stop()

- Her zaman gerekli değildir.
- Tarayıcı sayfasından ayrılırken çağrılır
- Destroy()'dan hemen önce çağrılır.
- Eğer devam etmesini istemediğiniz ağır hesaplamalar var ise stop() metodunu kullanabilirsiniz.

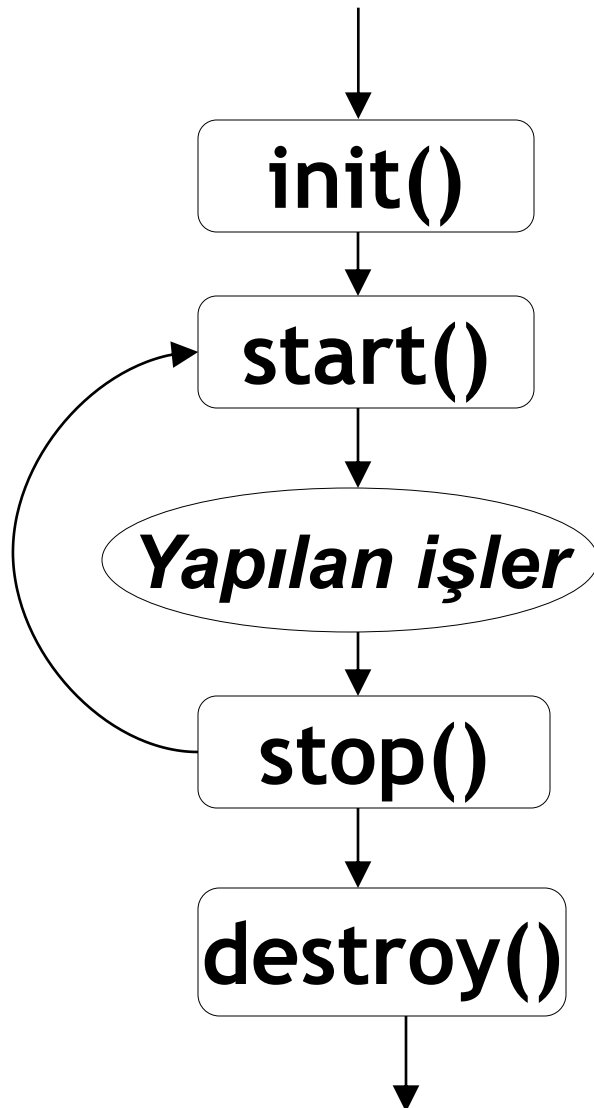


public void destroy()

- Nadiren gereklidir.
- Stop()’tan sonra çağrılır.
- Sistem kaynaklarını(thread) serbest bırakmak için kullanılır.
- Sistem kaynakları genellikle otomatik bırakılır



Metotlar bu sırada çağrılır.



- **init** ve **destroy** sadece bir kez çağrılır.
- **start** ve **stop** tarayıcıya her girildiğinde ve çıkıldığında çağrılır.
- *Yapılan işler listener'larınız tarafından çağrılan kodlardır.*
- Applet yeniden çizilecekse **paint** çağrılır.



public void paint(Graphics g)

- Standard GUI bileşenlerini kullanmadan bir şeyler çizmek için kullanılır.
- Components
- Yapmak istediğini herhangi bir çizim veya çağırmak istediğiniz bir metot burada yapılır
- *paint() metodunu başka yerden çağırmayınız, repaint() olarak çağırınız*



repaint()

- Bir şeyler değiştirdiğinizde veya ekrandaki değişimleri göstermek istediğinizde repaint() metodunu çağırabilirsiniz.
- repaint() bir istektir, oluşmayabilir.
- repaint()'i çağırdığınızda, Java update (Graphics g) 'ti çağırmak için düzenlenir.



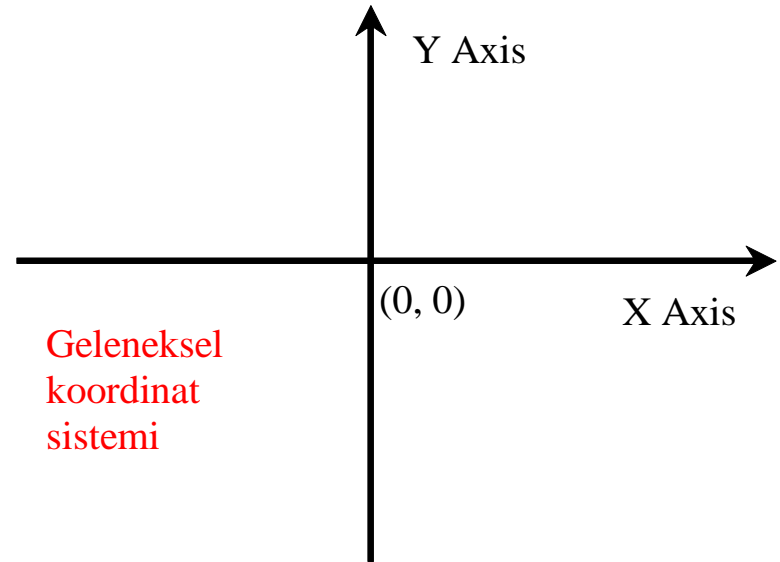
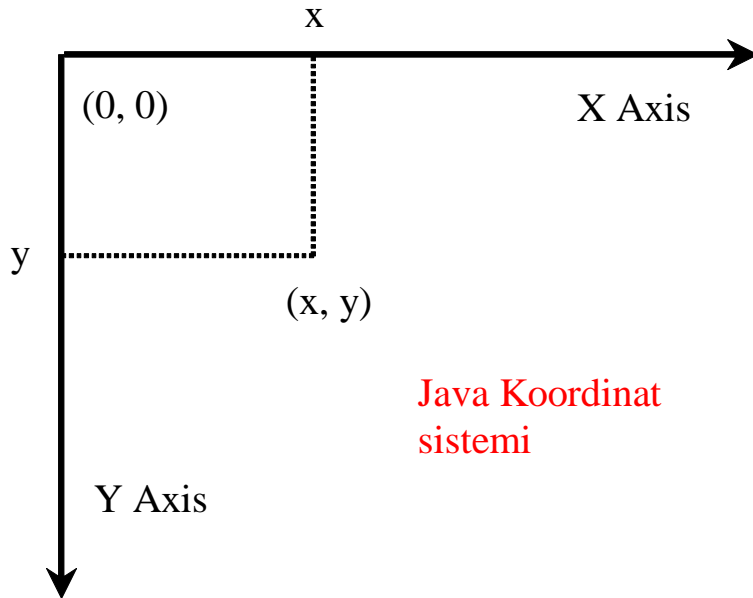
update()

- repaint()'i çağırdığınızda, Java update (Graphics g) 'ti çağırmak için düzenlenir.
- update'in yaptığı şey:

```
public void update(Graphics g) {  
    // arkaplan rengi ile applet'i doldur ve  
    paint(g);  
}
```



Java Koordinat Sistemi



Basit Java Appleti: Bir String ifadeyi Applete Yazmak

- Şimdi, kendi appletimizi oluşturacağız.
 - Demodakiler gibi örnek yapmak için biraz zamanımız var.
 - Birçok farklı teknikleri bilmemiz gerekir.
- Programımız
 - Yeni bir applet oluşturacağız ve bu applet aşağıdakini yazacak;
“java dünyası yeniden Merhaba!”
 - Applet ve HTML dosyaları gösterilecek, sonra satır satır inceleme yapacağız.

```
1  // merhabaApplet.java
2  // ilk applet programımız.
3
4  // Java packages
5  import java.awt.Graphics;    // import class Graphics
6  import javax.swing.JApplet;  // import class JApplet
7
8  public class merhabaApplet extends JApplet {
9
10     // applet arka planına yazı yazma
11     public void paint( Graphics g )
12     {
13         // her aplette olması gereken satır
14         super.paint( g );
15
16         // x,y koordinatı 25,25 olan noktadan itibaren yazıyı yaz
17         g.drawString( "Welcome to Java Programming!", 25, 25 );
18
19     } // end method paint
20
21 } // end class
```



Basit Java Appleti: Bir String ifadeyi Applete Yazmak

```
1 // merhabaApplet.java
```

```
2 // ilk applet programımız.
```

– Yorumlar

- Kaynak kodun ismi ve appletin tanıtımı

```
5 import java.awt.Graphics; // import class Graphics
```

```
6 import javax.swing.JApplet; // import class JApplet
```

– Önceden tanımlı sınıfları programımıza dahil etme

- Bir applet oluşturacağın zaman, `JApplet` sınıfını programına dahil etmelisin. (package `javax.swing`)
- Grafik çizimi yapabilmek için `Graphics` sınıfını (package `java.awt`) programına dahil etmelisin
 - Çizgi, dörtgen, oval çizip yazı yazabilmek için

Basit Java Appleti: Bir String ifadeyi Applete Yazmak

- Appletde de en az bir tane sınıf tanımlanmalı (uygulama örneğinde olduğu gibi)

```
8 public class merhabaApplet extends JApplet {
```

- `class merhabaApplet` için class tanımlaması
 - Anahtar kelime **class** dan sonra class ismi
- class isminden sonra **extends**
 - Yeni classı türeteceğin ana class (`JApplet`)
 - `JApplet` : superclass (ana class)
 - `merhabaApplet` : altclass (türemiş class)
 - `merhabaApplet` classı `JApplet` in method ve datalarına sahip

Basit Java Appleti: Bir String ifadeyi Applete Yazmak

```
8 public class merhabaApplet extends JApplet {
```

- Sınıf `JApplet` bizim yerimize tanımlanmış
 - Birisi kalkmış “Bir şeyin applet olması için neler lazım” tanımlamış
 - Applets 200 fazla metoda sahip!
 - `extends JApplet`
 - Miras yolu ile metodalarını yeniden yazmaya gerek olmadan programıma dahil ediyorum.
 - `JApplet` sınıfının tüm detayını bilmeme gerek yok.

Basit Java Appleti: Bir String ifadeyi Applete Yazmak

```
8 public class merhabaApplet extends JApplet {
```

- Class merhabaApplet
appletviewer veya browser WelcomeApplet
sınıfının objesini oluşturur.
 - Anahtar kelime public gerekli.
 - Bir dosyada sadece bir tane public class olur.
 - public class ismi dosya ismi ile aynı olmalıdır.

Basit Java Appleti: Bir String ifadeyi Applete Yazmak

11 `public void paint(Graphics g)`

- Sınıfımız `JApplet` sınıfından `paint` metodunu miras alır.
 - `paint` metodunun default olarak gövde kodu yok.
 - Bizim sınıfımızdaki `paint` metodu yeniden tanımlanur.(override)
- Methods `paint`, `init`, ve `start`
 - Her appletin bu üç metoda ihtiyacı olamayabilir.
 - İhtiyacı olduklarını yeniden tanımlar.

Basit Java Appleti: Bir String ifadeyi Applete Yazmak

11 `public void paint(Graphics g)`

– Method `paint`

- Satır 11-19 `paint` metoduna aittir.
- Ekranı grafik çizer.
- `void` metodun işi bittiğinde hiçbirşey geri döndermeyeceği anlamındadır.
- Parantezler parametre listesini tanımlamak içindir.
 - Normalde, programcı metoda verilerini parametre yolu ile geçirir `JOptionPane.showMessageDialog` olduğu gibi
- `paint` parametrelerini otomatik olarak alır.
 - `Graphics` objesi `paint` tarafından kullanılır.

Basit Java Appleti: Bir String ifadeyi Applete Yazmak

```
14      super.paint( g );
```

- superclass `JApplet` tarafından `paint` metodu çağrılır.
- Her applet'in `paint` metodunun ilk satırında olmalıdır.

```
17      g.drawString( "Welcome to Java Programming!", 25, 25 );
```

- `Paint` in gövde kodları
 - Method `drawString` (`Graphics` sınıfının)
 - `Graphics` türünde yaratılmış `g` objesi
 - Method ismi, parametreleri için parantez
 - Birinci parametre: yazılacak `String`
 - İkincisi: x koordinatı (in pixels)
 - Üçüncüsü: y koordinatı (in pixels)
- Java koordinat sistemi
 - Pixel olarak ölçeklenir.
 - Üst sol (0,0)



Basit Java Appleti: Bir String ifadeyi Applete Yazmak

- Appleti çalıştırma
 - Derleme
 - `javac merhabaApplet.java`
 - Hata yoksa, bytecode lar `WelcomeApplet.class` içinde saklanır.
 - HTML dosyası yüklemek
 - Dosyayı `appletviewer` veya browsera yüklemek
 - `.htm` or `.html` bitebilir.
 - Appleti çalıştırma
 - HTML dosyasında hangi applet varsa o applet çalışır.

Basit Java Appleti: Bir String ifadeyi Applete Yazmak

```
1 <html>
2 <applet code = "merhabaApplet.class" width = "300" height = "45">
3 </applet>
4 </html>
```

- Basit HTML dosyası (merhabaApplet.html)
 - Genellikle **.class** dosyası ile aynı klasörde bulunur.
 - Hatırlayın, **.class** dosyası derlemeden sonra oluşur.
- HTML kodları (tagler)
 - Genellikle çiftler halinde yazılır.
 - < ile başlar . > ile biter.
- Satır 1 ve 4 – başlangıç bitiş HTML tagleri
- Satır 2 - <applet> tagine başlayış
 - Applet için özel bir tag
 - Appleti görüntülemek için **width** ve **height** parametreleri ister.
- Satır 3 - </applet> taginin bitişi

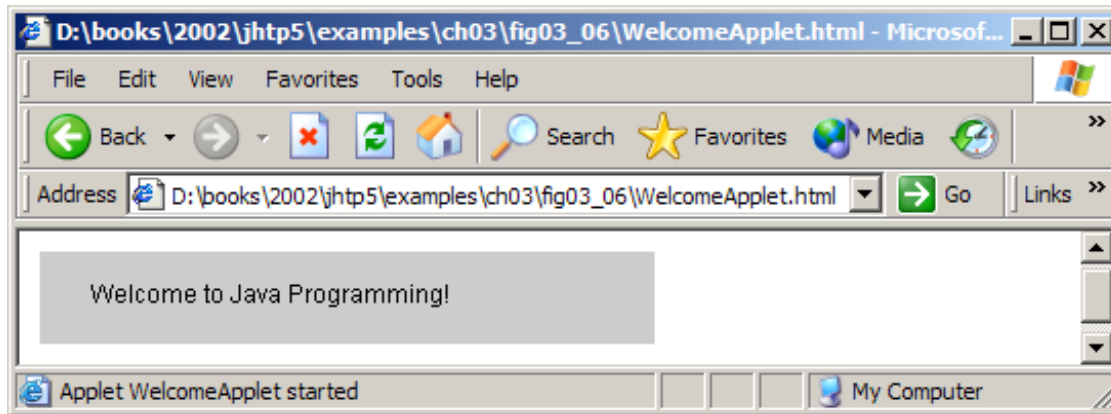
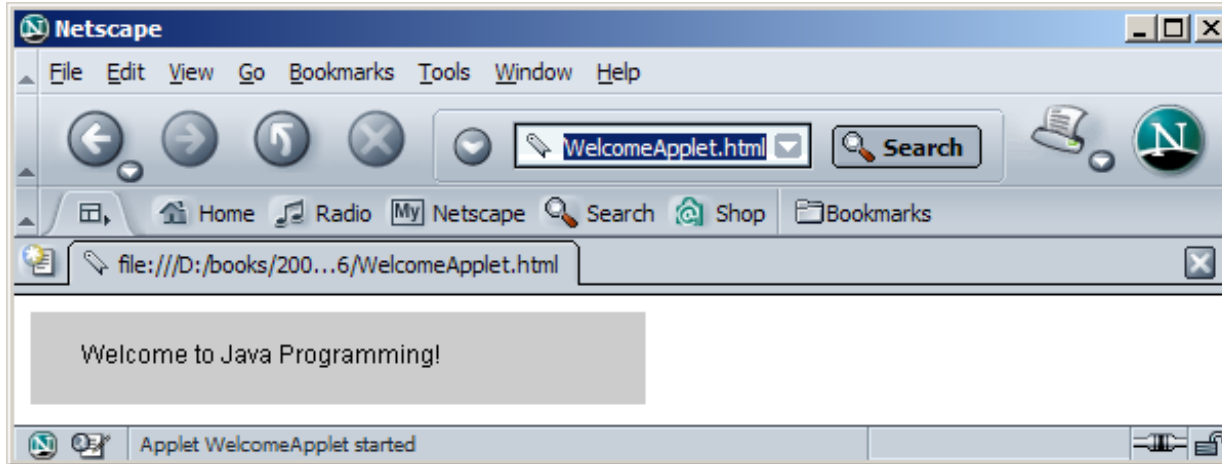
Basit Java Appleti: Bir String ifadeyi Applete Yazmak

```
1 <html>
2 <applet code = "merhabaApplet.class" width = "300" height = "45">
3 </applet>
4 </html>
```

- appletviewer sadece <applet> taglerinden anlar.
 - Gerisini iptal eder.
 - Küçük tarayıcı
- Appleti çalıştırma
 - appletviewer merhabaApplet.html
 - .class dosyası çalışır.

Basit Java Appleti: Bir String ifadeyi Applete Yazmak

- Web browser da appletin çalışması

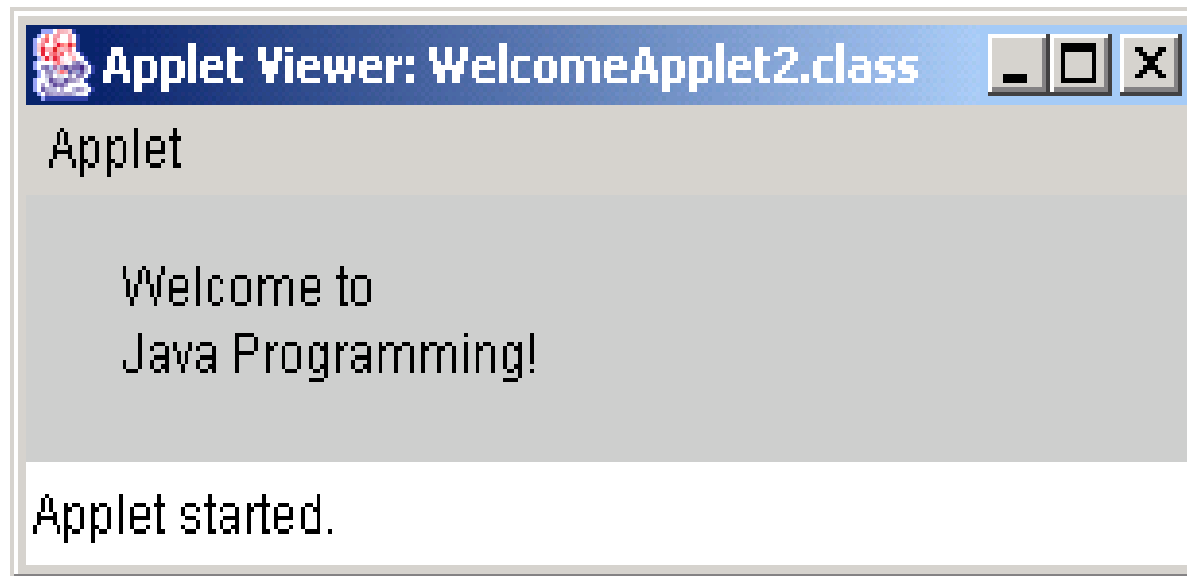


Çizgi Çizmek ve String Yazmak

- Daha fazla applet
 - Birinci örnek
 - İki satırlık text yazdırmak
 - `drawString` kullanarak iki satırlık text yazdırmak.
 - İkinci örnek
 - Method `g.drawLine(x1, y1, x2, y2)`
 - `(x1, y1)`'den `(x2, y2)`'ye kadar çizgi çizmek
 - Hatırla `(0, 0)` appletin sol üst kısmını işaret eder.
 - `drawLine` kullanarak text in altını ve üstünü çerçeveleyelim.

```
1 // welcomeApplet2.java
2 // 2 satır yazı yazdırma.
3
4 // Java packages
5 import java.awt.Graphics; // import class Graphics
6 import javax.swing.JApplet; // import class JApplet
7
8 public class welcomeApplet2 extends JApplet {
9
10     // draw text on applet's background
11     public void paint( Graphics g )
12     {
13         // call superclass version of method paint
14         super.paint( g );
15
16         // draw two Strings at different locations
17         g.drawString( "java dunyasına", 25, 25 );
18         g.drawString( "merhaba!", 25, 40 );
19
20     } // end method paint
21
22 } // end class welcomeApplet2
```

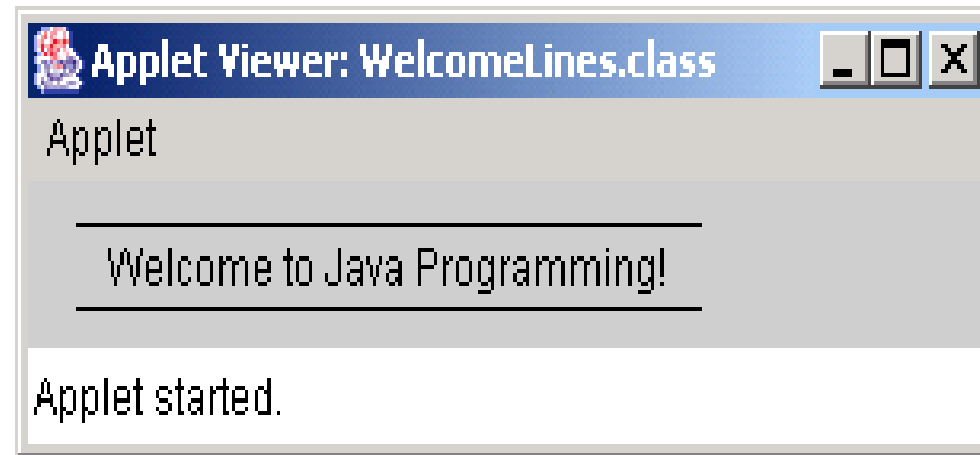
```
1 <html>
2 <applet code = "WelcomeApplet2.class" width = "300" height = "60">
3 </applet>
4 </html>
```



```
1 // welcomeLines.java
2 // yazı yazma ve çizgi çizme
3
4 // Java packages
5 import java.awt.Graphics; // import class Graphics
6 import javax.swing.JApplet; // import class JApplet
7
8 public class welcomeLines extends JApplet {
9
10     public void paint( Graphics g )
11     {
12         // paint metodunun super classını çağır
13         super.paint( g );
14
15
16         // (15, 10) den (210, 10) ye kadar yatay çizgi
17         g.drawLine( 15, 10, 210, 10 );
18
19         // (15, 30) den (210, 30) ye kadar yatay çizgi
20         g.drawLine( 15, 30, 210, 30 );
21
22         // yazıyı 25,25 noktasından başlayarak yaz
23         g.drawString( "Welcome to Java Programming!", 25, 25 );
24
25     } // end method paint
26
27 } // end class welcomeLines
```

```
1 <html>
2 <applet code = "WelcomeLines.class" width = "300" height = "40">
3 </applet>
4 </html>
```

HTML file



Çizgi Çizmek ve String Yazmak

- Graphics sınıfının drawLine metodu
 - Başlangıç x ve y koordinatı
 - Bitiş x ve y koordinatı

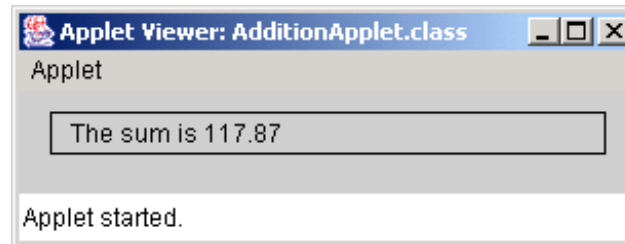
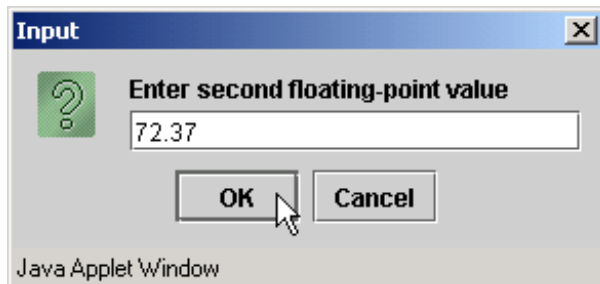
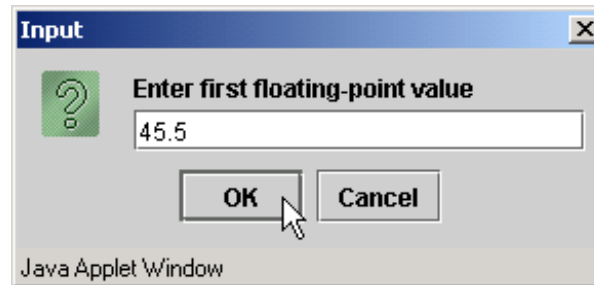
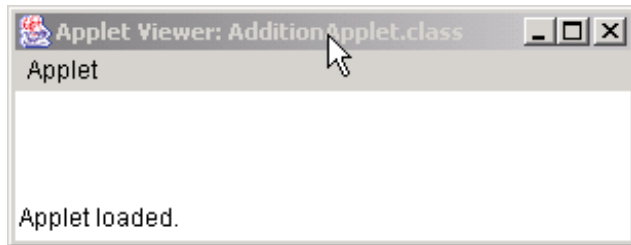
İki Sayıyı Ekleme

- Bir sonraki applet
 - Bu seferki küçük uygulamamız; iki sayının toplanması
 - Sayıları ondalıklı kullanacağız.
 - İlkel tipler
 - `double`
 - `float`
 - Programı görelim, sonra satır satır inceleyelim...

```
1 // AdditionApplet.java
2 // iki ondalıklı sayıyı toplama.
3
4 // Java packages
5 import java.awt.Graphics; // import class Graphics
6 import javax.swing.*;     // import package javax.swing
7
8 public class AdditionApplet extends JApplet {
9     double sum; ;
10
11     // initialize applet by obtaining values from user
12     public void init()
13     {
14         String firstNumber;
15         String secondNumber;
16
17         double number1;
18         double number2;
19
20         firstNumber = JOptionPane.showInputDialog(
21             "Enter first floating-point value" );
22
23         secondNumber = JOptionPane.showInputDialog(
24             "Enter second floating-point value" );
25
26         // convert numbers from type String to type double
27         number1 = Double.parseDouble( firstNumber );
28         number2 = Double.parseDouble( secondNumber );
29
30
31
```

```
32
33     sum = number1 + number2;
34
35 } // end method init
36
37 public void paint( Graphics g )
38 {
39     // call superclass version of method paint
40     super.paint( g );
41
42     // 15,10 noktasından başlayarak dörtgen çiz.
43     // boyu 270,eni 20 piksel olsun
44     g.drawRect( 15, 10, 270, 20 );
45
46     // draw results as a String at (25, 25)
47     g.drawString( "The sum is " + sum, 25, 25 );
48
49 } // end method paint
50
51 } // end class AdditionApplet
52
```

```
1 <html>
2 <applet code = "AdditionApplet.class" width = "300" height = "65">
3 </applet>
4 </html>
```



İki Sayıyı Eklemek

```
5  import java.awt.Graphics;    // import class Graphics
```

– Satır 5: `imports class Graphics`

- Eğer kodlarda paket ve sınıf ismi tam kullanılırsa `import` ifadesine gerek yoktur.

```
public void paint ( java.awt.Graphics g )
```

```
12 public void init()
```

```
6  import javax.swing.*;        // import package javax.swing
```

- * `javax.swing` içindeki tüm sınıflar erişilebilir.
 - `JApplet` ve `JOptionPane` kullanılmış
- * bütün sınıfları yüklemeyiz.
 - Derleyici sadece programda kullanılan sınıfları yükler.

İki Sayıyı Eklemek

```
8 public class AdditionApplet extends JApplet {
```

- Sınıf tanıtımı
 - `javax.swing` paketindeki `JApplet` ten miras alınmış.

```
9 double sum;
```

- Sınıf üye değişkeni tanımlama
 - Sınıf gövdesinde tanıtılmalı, metod içinde tanıtılmamalı
 - Metodda tanıtılan değişkenler yerel (lokal) değişkenlerdir.
 - Metodda tanıtılan değişken sadece metod içinde kullanılabilir.
 - Sınıf üyeleri sınıfın istenilen yerinde kullanılırlar.
 - Varsayılan değeri (`0.0` in this case)

İki Sayıyı Eklemek

```
9      double sum;
```

- İlkel tip `double`

```
12     public void init()
```

- Method `init`
 - Appletteki sınıfının ve üyelerinin ilk değer aldığı metod.
 - Applet çalışırken ilk çağrılan metod.
 - İlk satır her zaman yukarıdaki gibi gözükür.
 - Geriye bir şey döndürmez (`void`) ve herhangi bir argümanda almaz

```
13     {
```

`init` metoduna başlangıç

İki Sayıyı Eklemek

```
14      String firstNumber;  
15      String secondNumber;  
16  
17      double number1;  
18      double number2;
```

- Değişken tanımlama
- İki tür değişken vardır
 - Referans değişkenler (referanslar)
 - Objeyi referans eder. (bellekte konum içerir)
 - Objeler sınıf tanımlamasında tanımlanırlar.
 - Birçok data ve metodlar içerirler.
 - `paint` metodu `Graphics` objesinden `g` isminde referans alır.
 - Bu referans `Graphics` objesinin metodlarını çağırır.
 - ilkel tipler (değişkenler)
 - Sadece bir tane veriyi tutarlar.

İki Sayıyı Ekleme

```
14      String firstNumber;  
15      String secondNumber;  
16  
17      double number1;  
18      double number2;
```

- Referans ve değişkenler arasındaki farklılıklar
 - Eğer veri tipi sınıf ismi ise, bu referanstır.
 - `String` bir sınıftır.
 - `firstNumber`, `secondNumber`
 - Eğer veri tipi ilkel tip ise, bu değişkendir.
 - `double` bir ilkel tiptir.
 - `number1`, `number2`

İki Sayıyı Ekleme

```
21     firstNumber = JOptionPane.showInputDialog(  
22         "Enter first floating-point value" );
```

- Method `JOptionPane.showInputDialog`
 - Kullanıcıdan string bir girdi dönderir.
 - Text alan yazılır ardından **OK** basılır.
 - Eğer yanlış tip girdi ise hata verir.
 - Kullanıcı girişini string olarak geri dönderir.
- Satır 25-26: Yukardaki aynı işi `secondNumber` için yapar.

İki Sayıyı Ekleme

```
29     number1 = Double.parseDouble( firstNumber );  
30     number2 = Double.parseDouble( secondNumber );
```

- static method `Double.parseDouble`
 - String ifadeyi double çevirir.
 - Static method söz dizimi(syntax) hatırlayın.
 - `Classİsmi.methodİsmi(argumanlar)`

İki Sayıyı Eklemek

```
35      } // end method init
```

– Init metodunun bitişi

- `appletviewer` (veya `browser`) `start` metodunu çağırır.
- `start` ugenellikle çoklu kanallama (`multithreading`) de kullanılır.
 - Biz bu programımızda onu ilan etmediğimizden (`declare`) kullanılmıyor.
- Sonra, method `paint` çağrılır.

```
45      g.drawRect( 15, 10, 270, 20 );
```

– Method `drawRect(x1, y1, width, height)`

- Sol üst koordinatı (`x1`, `y1`) olan ve genişliği (`width`) ve yüksekliği (`height`)
- Satır 45 (15, 10) noktasından başlayıp genişliği 270 pixel , yüksekliği 20 pixel olan bir dikdörtgen çizer.

İki Sayıyı Eklemek

48

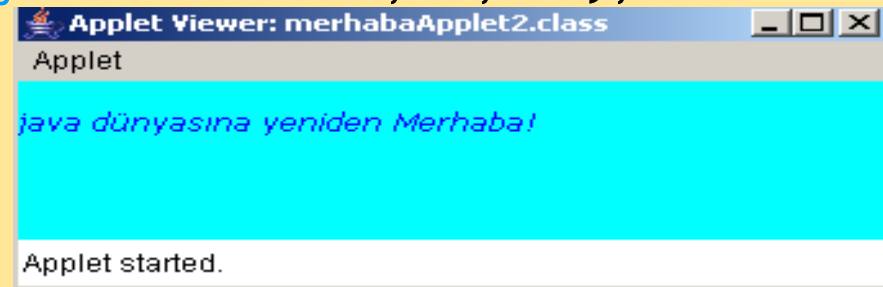
```
g.drawString( "The sum is " + sum, 25, 25 );
```

- g referansını kullanarak Graphics objesinin drawString metoduna "The sum is" + sum yollanıyor.
 - sum stringe çevrildi.
- sum paint metodunun içinde tanımlanmadığı halde kullanıldı.
 - Local değişken değil

```

1 // merhabaApplet.java
2 // ilk applet programımızda renkler ve yazılar.
3
4 // Java packages
5 import java.awt.*; // tüm çizim sınıflarını dahil ediyoruz
6 import javax.swing.JApplet; // import class JApplet
7
8 public class merhabaApplet extends JApplet {
12 // yeni yazı stili tanımlar
13 Font f=new Font("verdana",Font.ITALIC,12);
14
11 public void paint( Graphics g )
12 {
14     super.paint( g );
15
12     //arka plan rengini değiştirir
13     setBackground(Color.cyan);
14     // global yazı stilin f yi, g grafik objesine uyarlar
15     g.setFont(f);
16     // g nesnesinin renk özelliğini değiştirir.
17     g.setColor(Color.BLUE);
18
16     // x,y koordinatı 25,25 olan noktadan itibaren yazıyı yaz
17     g.drawString( "java dünyasına yeniden merhaba!", 25, 25 );
18
19 } // end method paint
20 } // end class

```



```

import java.awt.*;
import javax.swing.*;
public class parametre extends JApplet {

Font f=new Font("verdana",Font.ITALIC,12);
public void paint( Graphics g )
{
    setBackground(Color.cyan);
    g.setFont(f);
    g.setColor(Color.MAGENTA);

    String paramAd,paramSad;
    //html kodundaki parametrelerle verilen bilgiyi alır
    paramAd=getParameter("Ad");
    paramSad=getParameter("Soyad");

    // önemli: getParameter("Soyad") gönderilen string parametresi ile,
    // html kodundaki parametre adı aynı olmak zorunda.
    // <PARAM NAME="Soyad" VALUE="GÖRMEZ">

    g.drawString( "adınız: "+paramAd,25, 25 );
    g.drawString( "soyadınız: "+paramSad,25, 35 );

} //
}

```

adınız: ZELİHA
soyadınız: GÖRMEZ

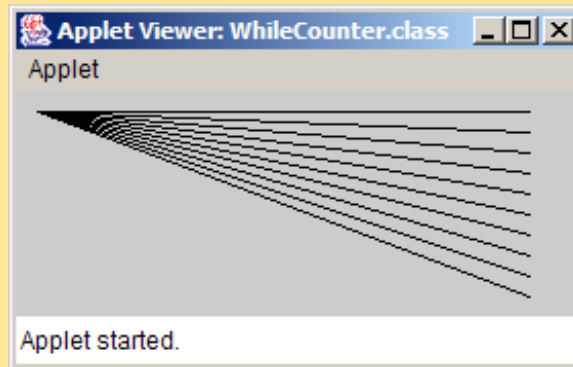
adınız: Fatih sultan
soyadınız: Mehmet

Örnek Programlar

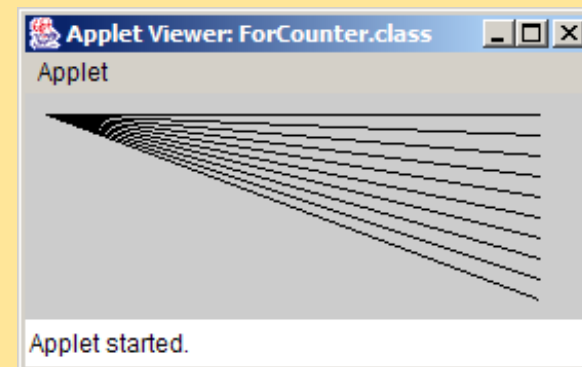
- for
- while
- do while



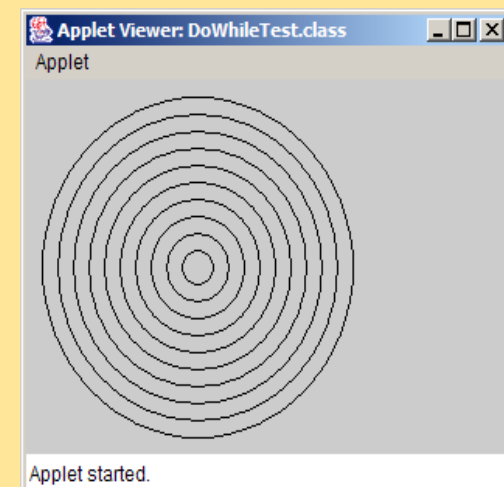

```
1 // Fig. 5.1: WhileCounter.java
2 // Counter-controlled repetition.
3 import java.awt.Graphics;
4
5 import javax.swing.JApplet;
6
7 public class WhileCounter extends JApplet {
8
9     // draw lines on applet's background
10    public void paint( Graphics g )
11    {
12        super.paint( g ); // call paint method inherited from JApplet
13
14        int counter = 1; // initialization
15
16        while ( counter <= 10 ) { // repetition condition
17            g.drawLine( 10, 10, 250, counter * 10 );
18            ++counter; // increment
19
20        } // end while
21
22    } // end method paint
23
24 } // end class WhileCounter
```



```
1 // ForCounter.java
2 // Counter-controlled repetition with the for statement.
3 import java.awt.Graphics;
4
5 import javax.swing.JApplet;
6
7 public class ForCounter extends JApplet {
8
9     // draw lines on applet's background
10    public void paint( Graphics g )
11    {
12        super.paint( g ); // call paint method inherited from JApplet
13
14        // for statement header includes initialization,
15        // repetition condition and increment
16        for ( int counter = 1; counter <= 10; counter++ )
17            g.drawLine( 10, 10, 250, counter * 10 );
18
19    } // end method paint
20
21 } // end class ForCounter
```



```
1 // Fig. 5.7: DoWhileTest.java
2 // Using the do...while statement.
3 import java.awt.Graphics;
4
5 import javax.swing.JApplet;
6
7 public class DoWhileTest extends JApplet {
8
9     // draw lines on applet
10    public void paint( Graphics g )
11    {
12        super.paint( g ); // call paint method inherited from JApplet
13
14        int counter = 1; // initialize counter
15
16        do {
17            g.drawOval( 110 - counter * 10, 110 - counter * 10,
18                      counter * 20, counter * 20 );
19            ++counter;
20        } while ( counter <= 10 ); // end do...while
21
22    } // end method paint
23
24 } // end class DoWhileTest
```

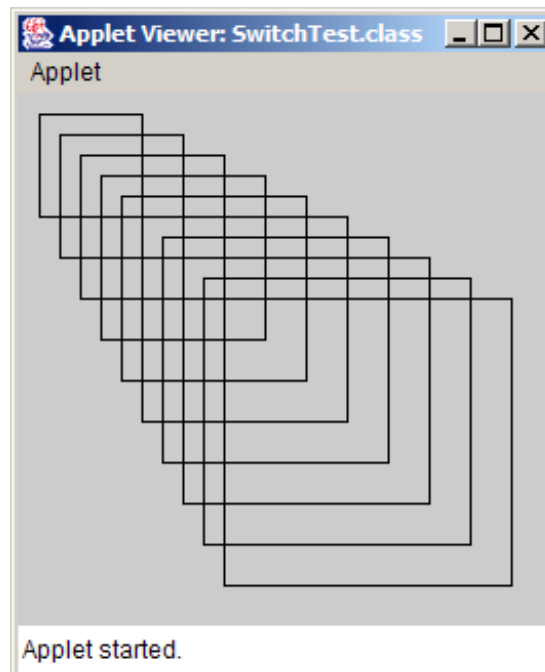
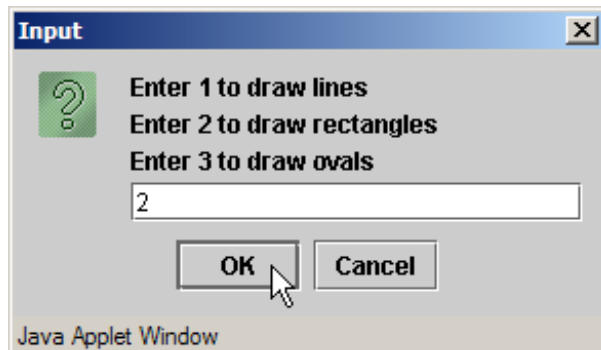
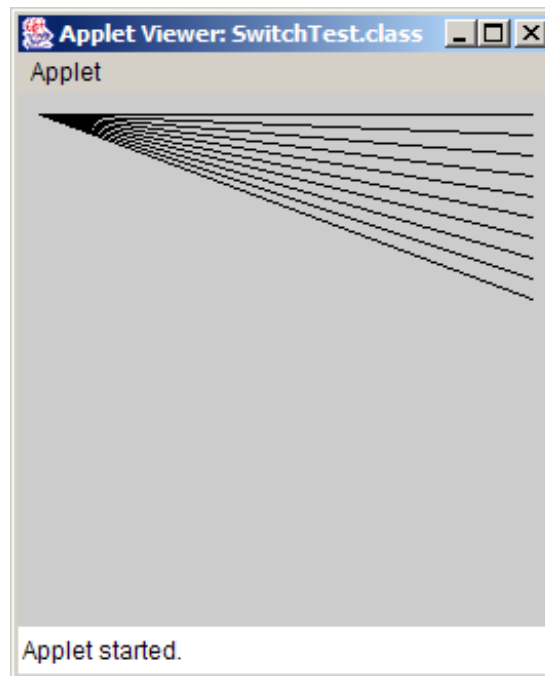
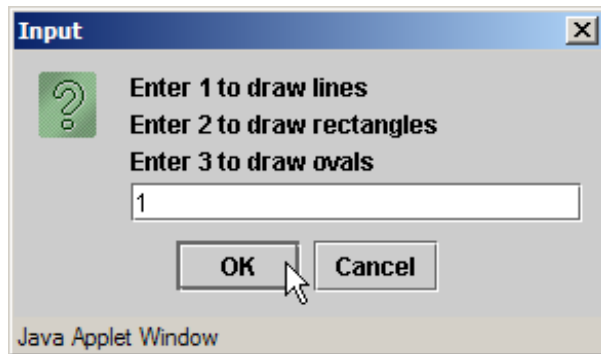


```
1 // Fig. 5.9: SwitchTest.java
2 // Drawing lines, rectangles or ovals based on user input.
3 import java.awt.Graphics;
4
5 import javax.swing.*;
6
7 public class SwitchTest extends JApplet {
8     int choice; // user's choice of which shape to draw
9
10    // initialize applet by obtaining user's choice
11    public void init()
12    {
13        String input; // user's input
14
15        // obtain user's choice
16        input = JOptionPane.showInputDialog(
17            "Enter 1 to draw lines\n" +
18            "Enter 2 to draw rectangles\n" +
19            "Enter 3 to draw ovals\n" );
20
21        choice = Integer.parseInt( input ); // convert input to int
22
23    } // end method init
24
25    // draw shapes on applet's background
26    public void paint( Graphics g )
27    {
28        super.paint( g ); // call paint method inherited from JApplet
29
30        for ( int i = 0; i < 10; i++ ) { // loop 10 times (0-9)
31
```

```
32     switch ( choice ) { // determine shape to draw
33
34         case 1: // draw a line
35             g.drawLine( 10, 10, 250, 10 + i * 10 );
36             break; // done processing case
37
38         case 2: // draw a rectangle
39             g.drawRect( 10 + i * 10, 10 + i * 10,
40                 50 + i * 10, 50 + i * 10 );
41             break; // done processing case
42
43         case 3: // draw an oval
44             g.drawOval( 10 + i * 10, 10 + i * 10,
45                 50 + i * 10, 50 + i * 10 );
46             break; // done processing case
47
48         default: // draw string indicating invalid value entered
49             g.drawString( "Invalid value entered",
50                 10, 20 + i * 15 );
51
52     } // end switch
53
54     } // end for
55
56     } // end method paint
57
58 } // end class SwitchTest
```

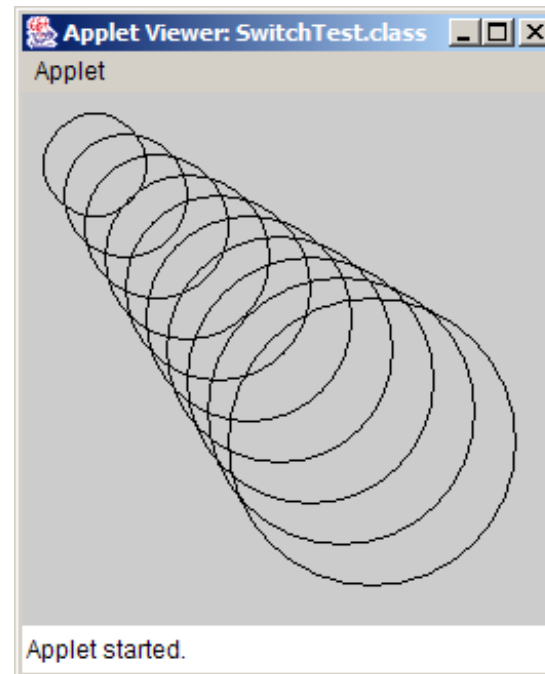
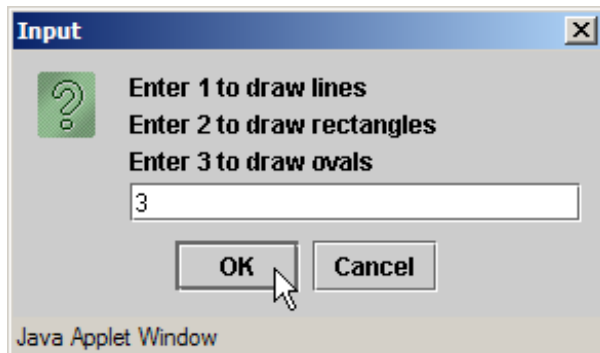
Outline

switchTest.java





SwitchTest.java



Java Applet Internet ve World Wide Web Kaynakları

- Birçok Java appletlerine ulaşılabilir
 - java.sun.com/applets/
 - Birçok kaynak kod ve free appletler indirilebilir.
 - Sun site developer.java.sun.com/developer
 - Tartışma oturumları, eğitici slaytlar, makaleler..vb.
 - Kayıt gerekli
 - www.jars.com
 - Appletler oylanıyor., top 1, 5 ve yüzde 25
 - Web deki en iyi appletleri bu sayfadan görebilirsiniz.