

APPLET

- ◆ ‘APPLET’, bir web tarayıcı aracılığıyla çalışan JAVA programıdır.
- ◆ ‘Applet’ sınıfı “Java.applet” paketinde bulunur.

APPLET

- ◆ Bir applet, yapısını çalışması sırasında gerçekleşen 4 olayla oluşturur.

Metot	Özellik
init()	Başlangıçta oluşturulur, başlangıç sürecinde appletin ihtiyaç duyduğu nesneler yaratılır. Bu metot grafiği yüklemek, değişkenlerin başlangıç değerlerini atamak ve nesneleri oluşturmak için kullanılır.
start()	Applet başladığında çağrılır. Başlangıç süreci bittiği zaman applet çalıştırılır. Bu metot, applet durduktan sonra yeniden başlatmak için kullanılır.
stop()	Appletin çalışması geçici olarak durdurulacaksa, çağrılır.
destroy()	Appleti yok etmek için kullanılır. Applet yok edildiği zaman bellek işlemci zamanı ve diske taşıma alanı sisteme geri verilir.

APPLET

◆ paint()

- Bu metot ekran üzerinde satır, metin veya resim gösterir. Türü graphics sınıfı olan bir argüman alır.

◆ repaint()

- Bir pencere güncelleneceği zaman kullanılır. Kullanılan parametre graphics sınıfının bir örneklemesidir.

◆ showStatus()

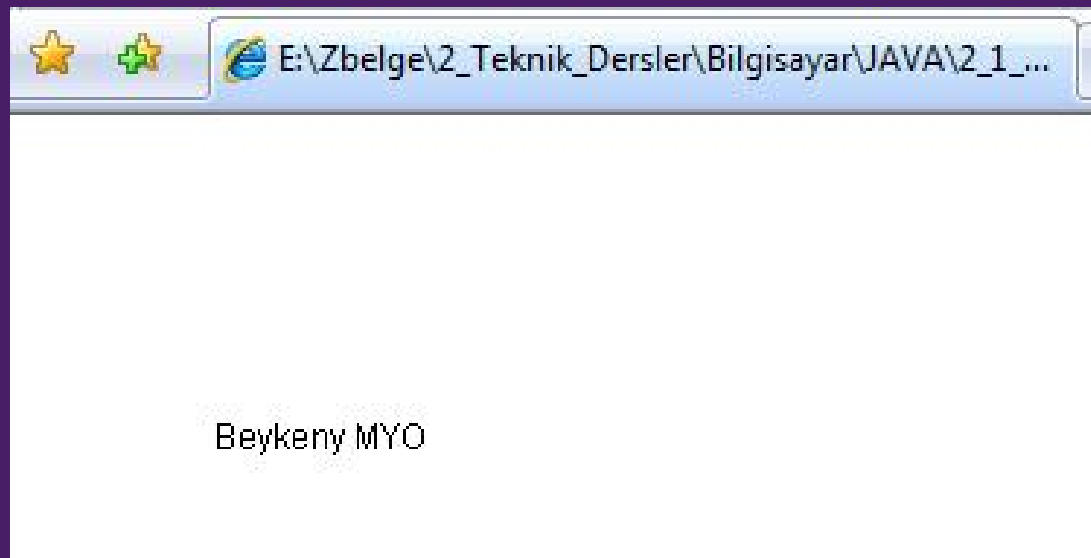
- Appletler bu metodu kullanarak durum çubuğunda bilgi gösterir. Bu metodun gizli parametresi vardır. Applet hakkında herhangi bir bilgi döndürmek için kullanıcı applet sınıfının getAppletInfo() metodunun üzerine yazabilir.

APPLET

applet1.java *

```
1 import java.applet.*;
2 import java.awt.*;
3 public class applet1 extends Applet{
4     int x,y;
5     public void init(){
6         x=60;
7         y=100;
8     }
9     public void paint(Graphics g){
10         g.drawString("Beykent MYO",x,y);
11         showStatus(getAppletInfo());
12     }
13     public String getAppletInfo(){
14         String info="Eğitimin doğru adresi";
15         return info;
16     }
17 }
```

```
<html>  
<applet code="applet1.class" width=500 height=400></applet>  
</html>
```



- ◆ Yazdığımız java dosyasını derleyin, derlediğiniz dosyaya ait .class dosyasını bir HTML dosyasına gömmeniz gerekir. Bunun için bir HTML dosyası oluşturun.

APPLET

- ◆ HTML dosyasına applet gömmek.
 - `<applet>` takısını kullanarak bir Appleti web sayfasına gömmeniz gerekir. Bu takıya ait temel 4 parametre vardır.
 - ◆ `width` : appletin sayfa içerisinde görünmesini istediğiniz kısmının genişliğini ayarlamak için kullanılır.
 - ◆ `Height` : appletin sayfa içerisinde görünmesini istediğiniz kısmının yüksekliğini ayarlamak için kullanılır.

APPLET

- ◆ code : eklemek istediğiniz appletin(.class uzantılı dosya) ismi uzantısı ile birlikte yazılır.
- ◆ codebase : Appletin bulunduğu klasörün tam yolu yazılır.

```
<html>
<head>
<title>ornek1</title>
<meta http-equiv="Content-Type" content="text/html;
  charset=iso-8859-9">
</head>

<body>
<applet code="applet1.class" codebase =
  "file:///C:/Beykent_dersleri" width="344"
  height="266">
</applet>
</body>
</html>
```

APPLET

- ◆ JAVA destekleyen herhangi bir web tarayıcı ile hazıladığınız HTML dosyasını kontrol edebilirsiniz, veya JDK ile birlikte gelen appletviewer aracını kullanarak HTML dosyanızı test edebilirsiniz.
- ◆ Appletviewer'ın kullanımı :
 - `apllletviewer applet1.html`

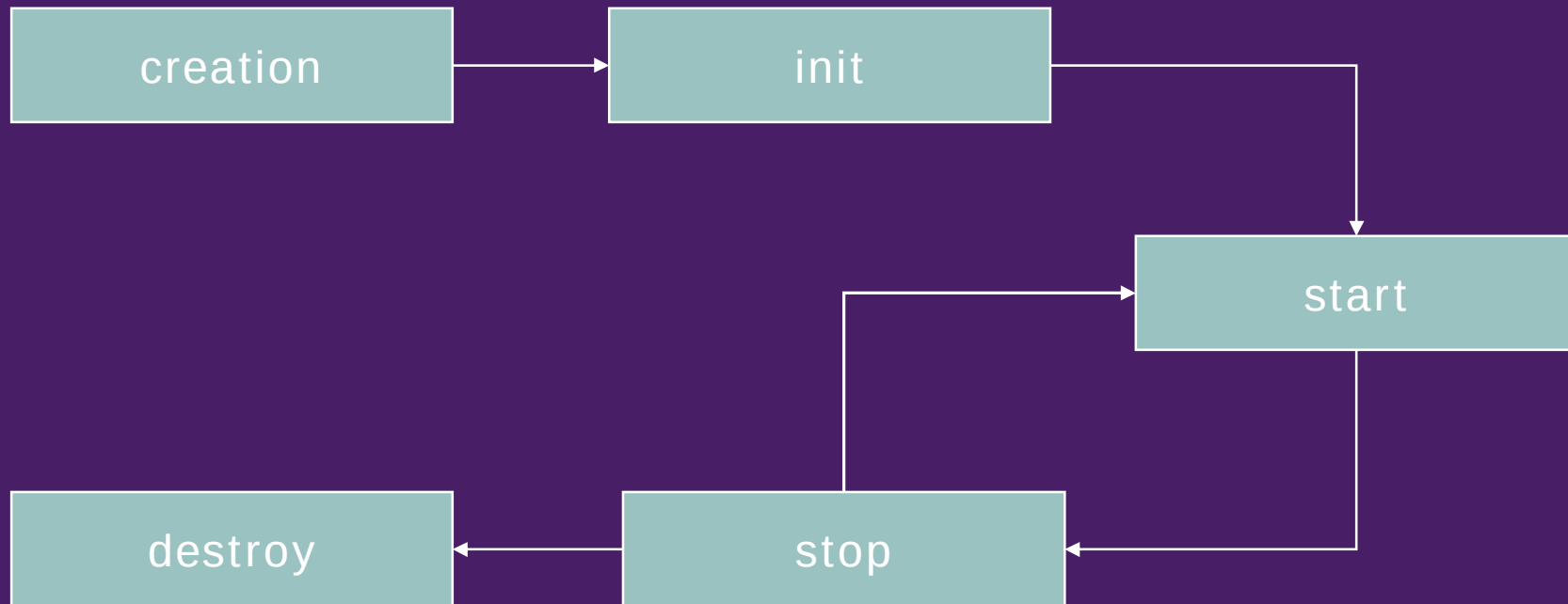
APPLET

◆ Güvenlik kısıtlamaları :

- Bir appletin neler yapabileceği ile ilgili yoğun bir kısıtlama vardır. Bir java dosyası kullanıcının sistemine zarar veren kodlar içerebilir. Bu nedenle aşağıdaki kısıtlamalar gerçekleştirilir.
 - ◆ Kullanıcının bilgisayarındaki dosyaları okuyamaz, değiştiremez ve silemez.
 - ◆ Web sayfasını(applet dahil) sağlayan dışında hiçbir internet sitesi ile iletişim kuramaz.
 - ◆ Kullanıcının sisteminde hiçbir program çalıştıramaz.
 - ◆ Kullanıcının sisteminde tutulan hiçbir programı yükleyemez.

APPLET

◆ Bir Appletin Yaşam döngüsü.



applet2.java *

```
1 import java.applet.*;
2 import java.awt.*;
3 import java.awt.event.*;
4 public class applet2 extends Applet implements MouseListener{
5     static int i;
6     public void mouseExited(MouseEvent e){}
7     public void mousePressed(MouseEvent me){}
8     public void mouseReleased(MouseEvent me){}
9     public void mouseEntered(MouseEvent me){}
10    public void mouseClicked(MouseEvent me){}
11    public void mouseReleased(MouseEvent me){repaint();}
12    public void init(){i=0;addMouseListener(this);}
13    public void paint(Graphics g){
14        if(i>50){this.destroy();}
15        else{g.drawString(String.valueOf(i),20,15);}
16        i++;
17    }
18    public void start(){
19        Frame f=new Frame("pencere");
20        f.setSize(400,400);
21        f.setVisible(true);
22    }
23    public void stop(){
24        Frame f=new Frame("pencere 2");
25        f.setSize(400,400);
26        f.setVisible(true);
27    }}
```

APPLET

- ◆ Dizgi-Karakter ve Byte çizme
 - drawString()
 - drawChars()
 - drawBytes()

APPLET

- ◆ `drawString(String str,int xCoor,int yCoor);`
 - Str, yazılacak dizgidir,
 - xCoor, dizginin yazılacağı çerçevedeki x koordinatıdır.
 - yCoor, dizginin yazılacağı çerçeve üzerindeki y koordinatıdır.
- ◆ `drawChars(char array[],int offset, int length,int xCoor, int yCoor);`
 - Array[], karakter dizisi,
 - Offset, yazılacak karakterlerin başlangıç pozisyonu,
 - Length, diziden yazılacak karakterlerin sayısı.
- ◆ `drawBytes(byte array[], int offset, int length, int xCoor, int yCoor);`
 - Array[], byte dizisi.

APPLET

applet3.java

```
1 import java.awt.*;
2 import java.applet.*;
3 import java.awt.event.*;
4 public class applet3 extends Applet{
5     public void paint(Graphics g){
6         String str="Beykent MYO";
7         char ch[]={'B','e','y','k','e','n','t'};
8         byte byt[]={65,66,67,68,69,70,71,72};
9         g.drawString(str,10,20);
10        g.drawChars(ch,0,7,10,40);
11        g.drawBytes(byt,3,4,10,60);
12    }
13 }
```



E:\Zbelge\2_Teknik_Dersler\Bil

Beykent MYO

Beykent

DEFG

APPLET

◆ Çizimler

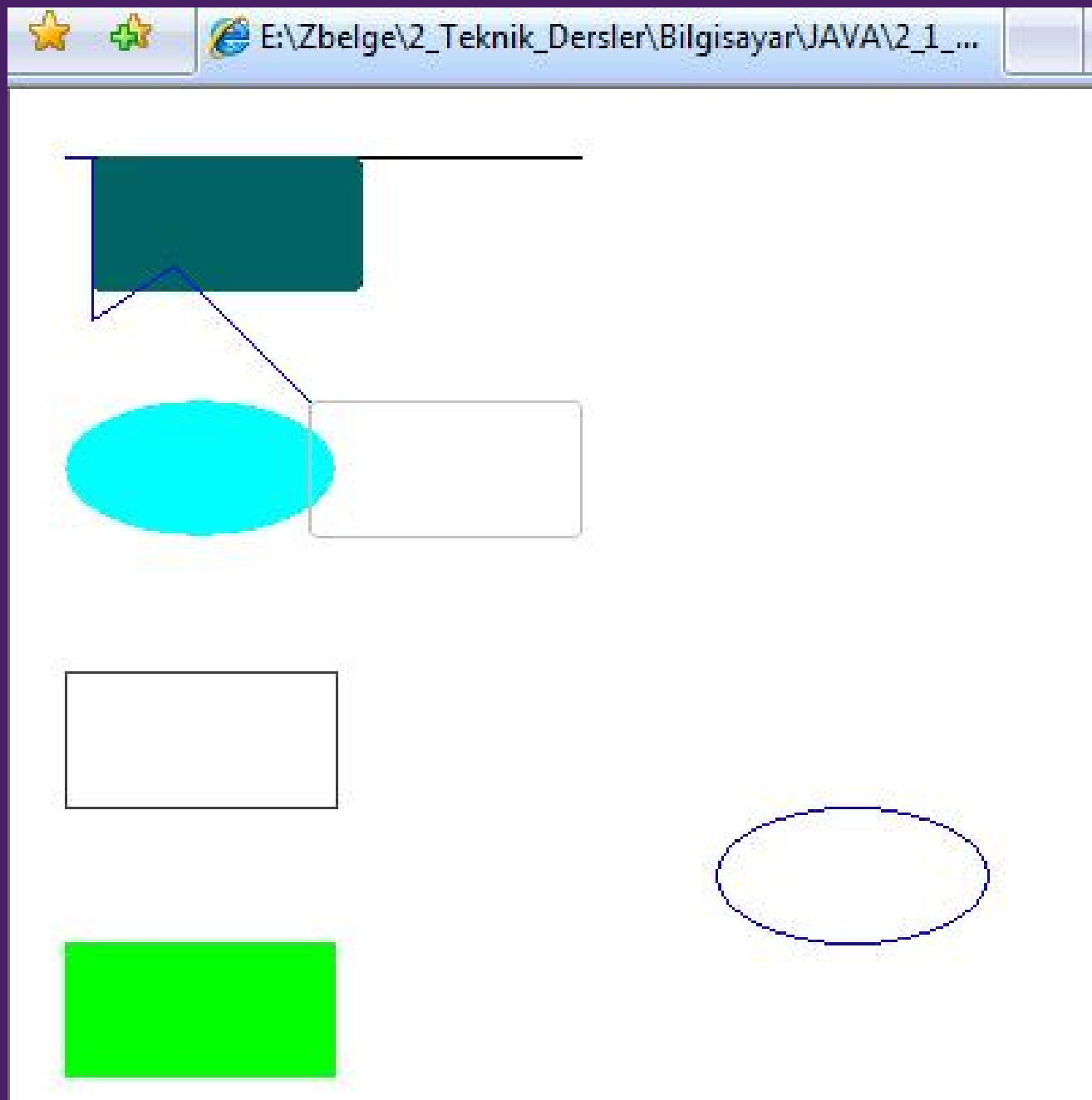
- Drawline
- drawOval
- fillOval
- drawRect
- fillRect
- drawRaundRect
- fillRaundRect
- drawPolyLine

APPLET

- ◆ drawLine
 - DrawLine(int x,int y,int w,int h)
- ◆ drawOval
 - drawOval(int x,int y,int w, int h)
- ◆ fillOval
 - fillOval(int x,int y, int w, int h)
- ◆ drawRect
 - drawRect(int x,int y,int w, int h)
- ◆ fillRect
 - fillRect(int x, int y, int w, int h)
- ◆ drawRaundRect
 - drawRaundRect(int x,int y,int w, int h, int arcW, int arcH)
- ◆ fillRaundRect
 - fillRaundRect(int x,int y, int w, int h, int arcW,int arcH)
- ◆ drawPolyLine
 - drawPolyLine(int xPoints[],int yPoints[],int length)

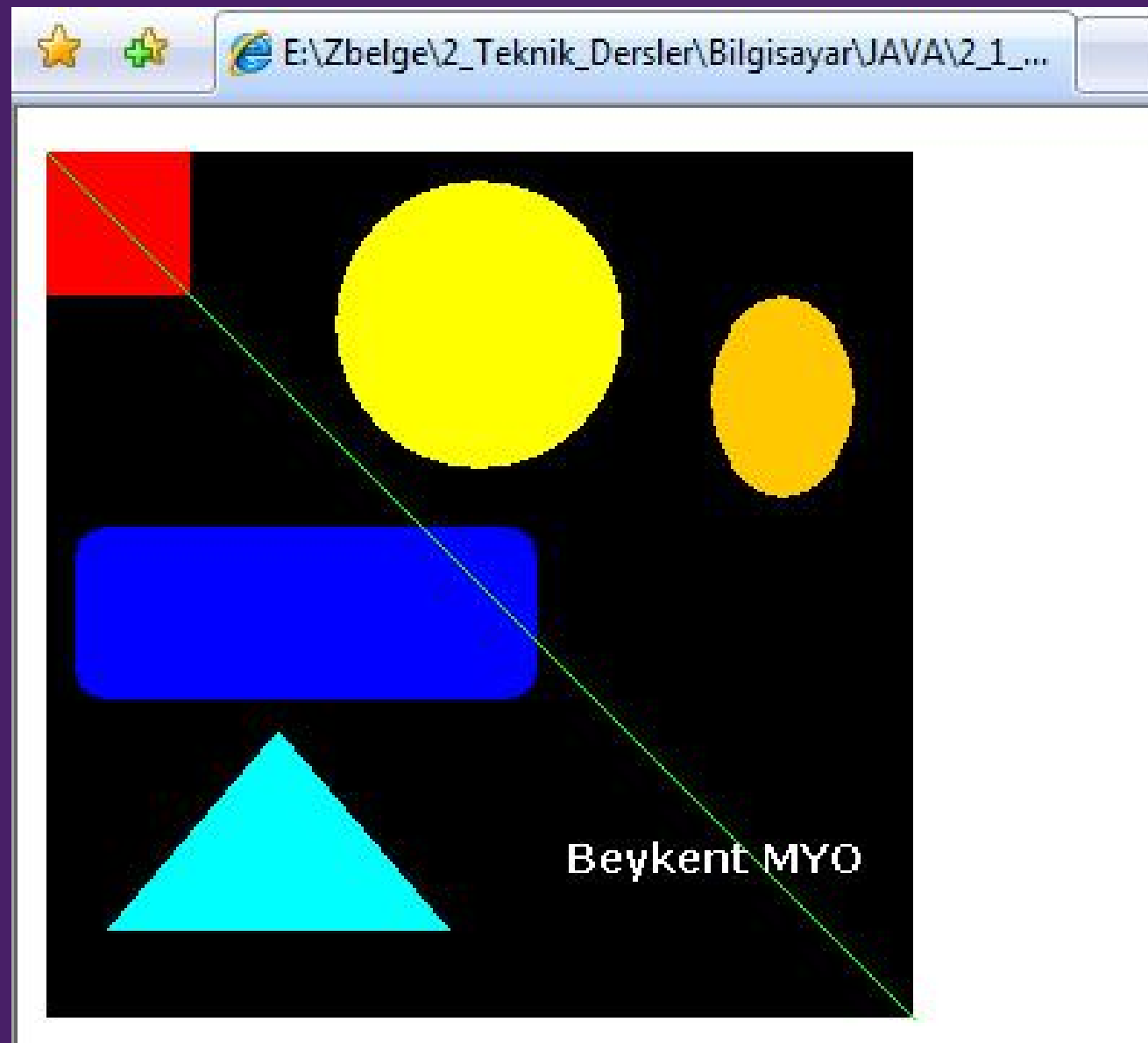
applet4.java *

```
1 import java.awt.*;
2 import java.applet.*;
3 public class applet4 extends Applet{
4     public void paint(Graphics g){
5         //çizgi çizme
6         g.setColor(Color.black);g.drawLine(10,10,200,10);
7         //oval çizme
8         g.setColor(Color.blue);g.drawOval(250,250,100,50);
9         //içi dolu oval çizme
10        g.setColor(Color.cyan);g.fillOval(10,100,100,50);
11        //dikdörtgen çizme
12        g.setColor(Color.DARK_GRAY);g.drawRect(10,200,100,50);
13        //içi dolu dikdörtgen
14        g.setColor(Color.green);g.fillRect(10,300,100,50);
15        //kenarları yuvarlak dikdörtgen çizme
16        g.setColor(Color.lightGray);g.drawRoundRect(100,100,100,50,5,5);
17        //içi dolu, köşeleri yuvarlak dikdörtgen çizme
18        Color c=new Color(0,100,100);
19        g.setColor(c);
20        g.fillRoundRect(20,10,100,50,5,5);
21        //çoklu çizgi çizme
22        int x[]={10,20,20,50,100};
23        int y[]={10,10,70,50,100};
24        g.setColor(Color.BLUE);
25        g.drawPolyline(x,y,5);
26    }}
```



ornek1.java *

```
1 import java.awt.*;
2 import java.applet.*;
3 public class ornek1 extends Applet{
4     public void paint (Graphics g){
5         //Arka zemin siyah (300 x 300 Pixel)kare
6         g.setColor(Color.black);
7         g.fillRect(0,0,300,300);
8         //Kırmızı kare
9         g.setColor(Color.red);g.fillRect(0,0,50,50);
10        //Sarı daire
11        g.setColor(Color.yellow);g.fillOval(100,10,100,100);
12        //turuncu elips
13        g.setColor(Color.orange);g.fillOval(230,50,50,70);
14        //Mavi dikdörtgen - köşeleri yuvarlatılmış
15        g.setColor(Color.blue);g.fillRoundRect(10,130,160,60,20,20);
16        //turkuaz renge ayarlıyor
17        g.setColor(Color.cyan);
18        //noktalar için x koordinatlarını belirliyor
19        int x[] = {20,80,140};
20        //set all y-coordinates of the corners
21        int y[] = {270,200,270};
22        //noktalar için y koordinatlarını belirliyor
23        g.fillPolygon(x,y,3);
24        //köşegen yeşil çizgi çiziyor
25        g.setColor(Color.green);g.drawLine(0,0,300,300);
26        //verilen koordinata metin yazıyor
27        g.setColor(Color.white);
28        g.setFont(new Font("Verdana",Font.BOLD,14));
29        g.drawString("Beykent MYO",180,250);
30    }}
```



Renkler

Eleman	Aralık
Kırmızı	0-255
Yeşil	0-255
Mavi	0-255

RGB Değerleri

Renk	Kırmızı	Yeşil	Mavi
Beyaz	255	255	255
Açık Gri	192	192	192
Gri	128	128	128
Koyu Gri	64	64	64
Siyah	0	0	0
Pembe	255	175	175
Turuncu	255	200	0
Sarı	255	255	0
Mürdüm	255	0	255

Color sınıfının sabitleri

color.white	color.black
color.orange	color.gray
color.lightgray	color.darkgray
color.red	color.green
color.blue	color.pink
color.cyan	color.magenta
color.yellow	

Font Kontrolü

- ◆ Font, metni farklı formatlarda gösterir. Java.awt paketinin sağladığı 'Font' sınıfı, çeşitli fontların kullanımını sağlar
- ◆ Bir fontu kullanmadan önce, sistemin onu destekleyip desteklemeyi kontrol etmemiz gerekir.
- ◆ 'getAllFont()' metodu, sistemin desteklediği tüm fontları döndürür.

Font Kontrolü

fontlar.java

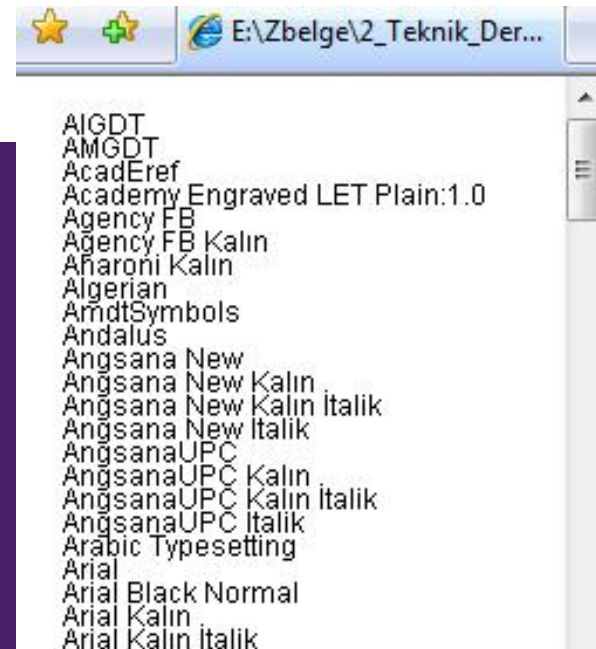
```
1 import java.awt.*;  
2 import java.applet.*;  
3 public class fontlar extends Applet{  
4     public void paint(Graphics g){  
5         g.setColor(Color.BLUE);  
6         Font f=new Font("Times New Roman",Font.ITALIC,24);  
7         g.setFont(f);  
8         g.drawString("Beykent MYO",10,20);  
9     }  
10 }
```



Font Kontrolü

applet6.java

```
1 import java.awt.*;
2 import java.applet.*;
3 import java.awt.GraphicsEnvironment;
4 public class applet6 extends Applet{
5     public void paint(Graphics g){
6         Font f[]=GraphicsEnvironment.getLocalGraphicsEnvironment().getAllFonts();
7         int j=10;
8         for(int i=0;i<f.length;i++){
9             g.drawString(f[i].getFontName(),10,j);
10            j+=10;
11        }
12    }
13 }
```



Font Kontrolü

◆ FontMetric sınıfı

- Height
- Baseline
- Ascent
- Descent
- Leading

Font Kontrolü

- ◆ Height
 - En uzun font karakterinin büyüklüğüdür.
- ◆ Baseline
 - Karakterlerinin tabanlarının (alt çıkıntıları hariç) üzerine yerleştirildiği çizgi.
- ◆ Ascent
 - Taban çizgisinden karakterin üst üst kısmına kadar olan uzaklık.
- ◆ Descent
 - Taban kısmından karakterin alt kısmına olan uzaklık
- ◆ Leading
 - Bir sonraki karakteri yerleştirmek için gerekli ek boşluklar.

Font Kontrolü

```
import java.awt.*;
import java.applet.*;
public class applet7 extends Applet{
    public void paint(Graphics g){
        Font f=new Font("Verdana",Font.BOLD,10);
        String s="Academy Of Information Technologies";
        FontMetrics fm=g.getFontMetrics(f);
        g.setFont(f);
        g.drawString(s,20,10);
        g.drawString(String.valueOf(fm.getHeight()),20,30);
        g.drawString(String.valueOf(fm.getAscent()),20,50);
        g.drawString(String.valueOf(fm.getDescent()),20,70);
        g.drawString(String.valueOf(fm.getLeading()),20,90);
    }
}
```

Font Kontrolü

◆ Boyama Modu

- Nesneler boyama modu kümesi kullanılarak çizilir, yeni bir nesne çizildiği zaman çizim nesneden önceki içeriğin üzerine yazılır. Benzer şekilde nesneler tekrar tekrar çizildiği zaman, çizim nesnesinin önceki içeriği silinir. Sadece yeni içerik gönderilir. Eski ve yeni içeriği görünür yapmak için 'graphics' sınıfı `setXORMode(Color c)` metodunu sunar.

Font Kontrolü

```
import java.awt.*;
import java.applet.*;
public class applet8 extends Applet{
    public void paint(Graphics g){
        g.setPaintMode();
        g.setColor(Color.black);
        g.fillRect(10,10,100,50);
        g.setColor(Color.blue);
        g.fillRect(60,35,100,50);

        g.setXORMode(Color.magenta);
        g.fillRect(10,150,100,50);
        g.setXORMode(Color.yellow);
        g.fillRect(60,175,100,50);

    }
}
```