



JAVA2D

Java2D API

- Geniş çaplı 2D geometrik şekillerin çizdirilmesini
- Birleşik (Composite) nesnelerin oluşturmalarını
- Renk yönetimini kolaylaştıracak genişletilmiş renk desteği özelliklerini
- Kompleks dokümanlar için çıktı özelliklerini sağlar.

Java2D API

- Java2D API
 - Çizgilerin kalınlığını
 - Dolguları
 - Temel dönüşüm işlemlerini
 - Bileşik nesne oluşturma işlemlerini kontrol etmemizi sağlar.
- Java2D API resimleri yükleme ve üzerlerinde işlem yapılmasına olanak tanır.
- AWT ile entegredir.

Java'da Painting

- AWT'de boyama işlemleri `paint(Graphics g)` metodu içinde gerçekleştirilir. Bu metod `java.awt.Canvas` sınıfının bir metodudur.
- Swing'de ise bu işlemler `java.swing.JComponent` sınıfının bir metodu olan `paintComponent(Graphics g)` metodu içinde gerçekleştirilir.
- `Graphics2D` nesneleri için gerekli tüm işlemler `paintComponent(Graphics g)` metodu içinde gerçekleştirilmelidir.

Şekillerin Çizimi

- AWT'de şekiller drawXxx veya fillXxx gibi komutlar çalıştırılarak gerçekleştirilirken Java2D'de öncelikle nesnenin kendisi oluşturulur daha sonra Graphics2D nesnesinin draw veya fill komutlarına bu nesne parametre olarak geçirilerek kullanılır.

AWT

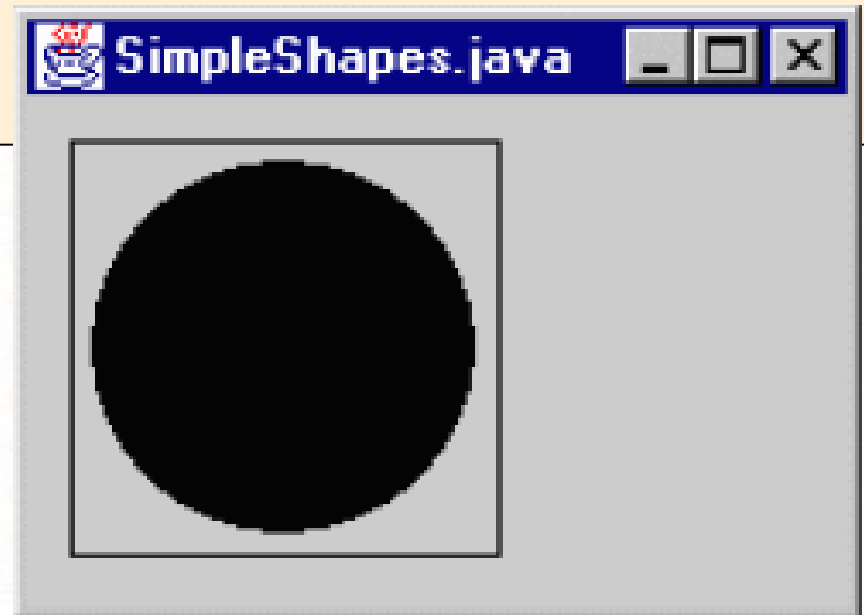
```
public void paint(Graphics g) {  
    g.setColor(...);  
    // Draw a shape  
    g.drawOval(...);  
}
```

Java2D

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    Graphics2D g2d = (Graphics2D)g;  
    Ellipse2D.Double circle = new Ellipse2D.Double(...);  
    g2d.fill(circle);  
    ...  
}
```

Örnek-1

```
public void paintComponent(Graphics g) {  
  
    Graphics2D g2d = (Graphics2D)g;  
    Ellipse2D.Double circle = new Ellipse2D.Double(15, 15, 90, 90);  
    Rectangle2D.Double square =  
        new Rectangle2D.Double(10, 10, 100, 100);  
  
    g2d.fill(circle);  
    g2d.draw(square);  
}
```

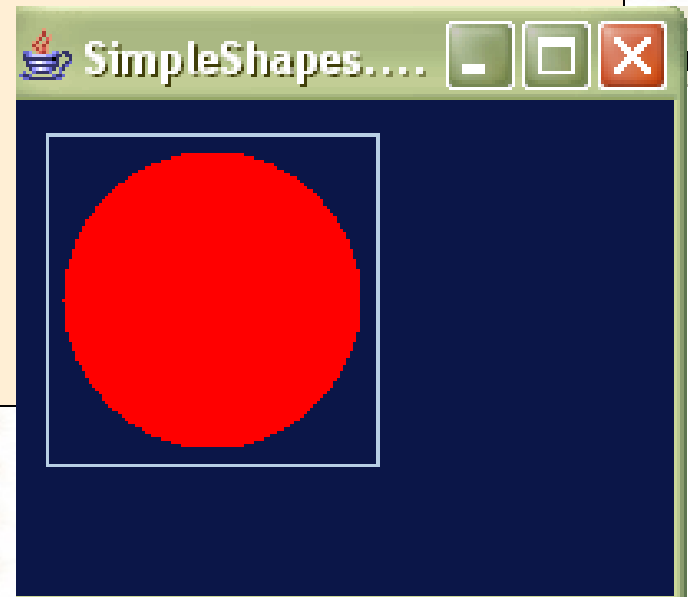


Java2D'de Renkler

- Renkler java.awt paketi içinde bulunur.
- Bir çok constructor'a sahiptir;
 - Color(float/int r, float/int g, float/int b)
 - Color(float/int r, float/int g, float/int b, float/int a)
 - Color(int rgb) (Mavi ilk byte, Yeşil 2. ve Kırmızı 3.byte)
 - JFC masaüstünüzle uygun olarak renkleri seçmenizi sağlar. Örneğin arkaplan rengi olarak genel pencere rengini seçmek istiyorsanız aşağıdaki yeterli olacaktır;
 - setBackground(UIManager.getColor("window"));
 - Diğer bazı sistem rengi anahtar kelimeleri ; desktop, windowtext, menu, text, info, scrollbar, ...

Örnek-2

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    Graphics2D g2d = (Graphics2D)g;  
    Ellipse2D.Double circle = new Ellipse2D.Double(15, 15, 90, 90);  
    Rectangle2D.Double square = new Rectangle2D.Double(10, 10, 100,  
100);  
  
    Color c = new Color(11,22,72);  
    setBackground(c);  
    g2d.setColor(Color.RED);  
    g2d.fill(circle);  
    g2d.setColor(UIManager.getColor("info"));  
    g2d.draw(square);  
}
```



java.awt.geom package içindeki sınıflar

- Point2D
- Arc2D
- Area
- CubicCurve2D
- Dimension2D
- Ellipse2D
- GeneralPath

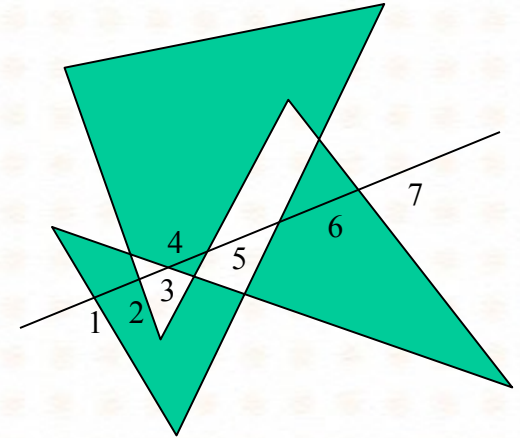
Noktalar

- Bir nokta ile pixel farklı şeylerdir.
- Point2D abstract bir sınıf olup içsel sınıflar içerir.
 - Point2D.Float, Point2D.Double gibi
- Bu sınıf ayrıca iki nokta arasındaki uzaklığı bulmak için fonksiyonlar sağlar;
 - public double distance (double PX, double PY)
 - public double distance (Point2D pt)

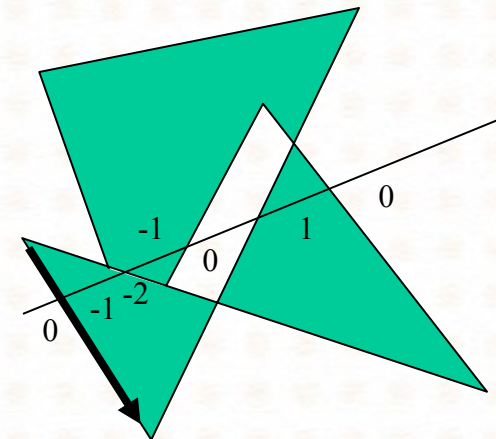
Şekiller – Winding Kuralı

- Karmaşık bir şekil çizdiniz. Fakat içini nasıl doldurabiliriz?
- Winding kuralı fill() komutu çalıştırıldığında çizilen şeklin hangi parçasının içsel hangisinin dışsal olduğunu belirler.
- Çift-Tek kuralı
 - Tüm şekil boyunca hat çizilir.
 - Her yüzey çakışmasında counter bir arttırılır
 - Eğer counter çift ise dışarıda tek ise içeridedir.
- Non-Zero Kuralı
 - Tüm şekil boyunca hat çizilir.
 - Eğer hat soldan sağa doğru uzanan bir hat ile çalışırsa counter arttırılır aksi halde azaltılır.
 - 0 olmayan bölgeler içeride kabul edilir

Çift-Tek



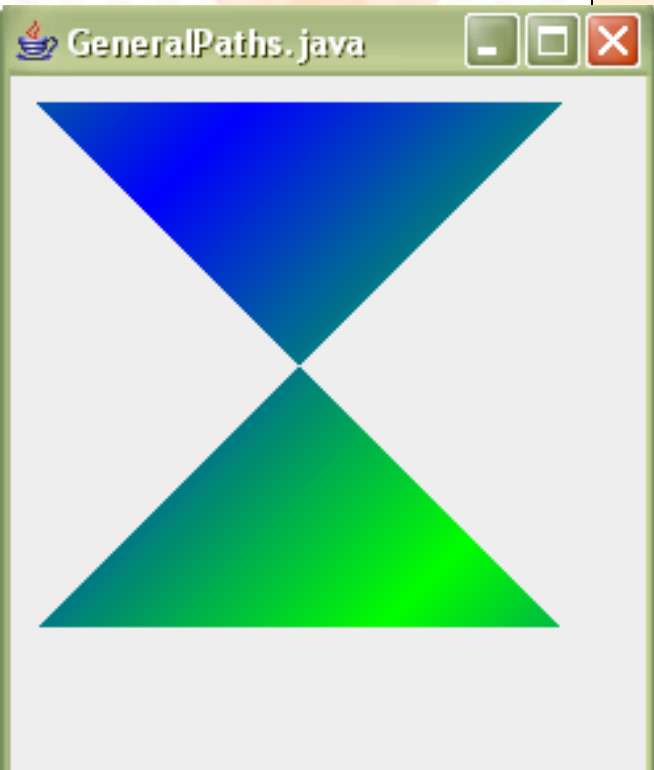
Non-zero



Şekiller – GeneralPath

- Parça parça bir path oluşturmayı sağlar
- Default olarak non-zero winding kuralını kullanır.
- Bu sınıfın constructor'ları
 - `public GeneralPath(int windingRule)`
 - `public GeneralPath(int windingRule, int InitialCapacity)`
- Metodları
 - `moveTo(float x, float y)`
 - `lineTo(float x, float y)`
 - `quadTo(float controlX1, float controlY1, float endX, float endY)`
 - `cubicTo(float controlX1, float controlY1, float controlX2, float controlY2, float endX, float endY)`
 - `append(Shape s, boolean connect)`
 - `closePath()`

Örnek 3



```
public class GeneralPaths2 extends JPanel {  
    private GradientPaint gPaint = new GradientPaint(50, 50,  
        Color.blue, 175, 175, Color.green, true);  
    int x = 10;  
    int y = 10;  
    int rectWidth = 200;  
    int rectHeight = 200;  
    int x1Points[] = {x, x+rectWidth, x, x+rectWidth};  
    int y1Points[] = {y, y+rectHeight, y+rectHeight, y};  
    GeneralPath polygon = new  
        GeneralPath(GeneralPath.WIND_EVEN_ODD, x1Points.length);  
    public GeneralPaths2() {  
        super();  
        setLayout(null);  
        polygon.moveTo(x1Points[0], y1Points[0]);  
  
        for (int index = 1; index < x1Points.length;  
            polygon.lineTo(x1Points[index],  
                y1Points[index]);  
        polygon.closePath();  
    }  
    public void paintComponent(Graphics g) {  
        super.paintComponent(g);  
        Graphics2D g2d = (Graphics2D)g;  
        g2d.setPaint(gPaint);  
        g2d.fill(polygon);  
    }  
    ....}
```

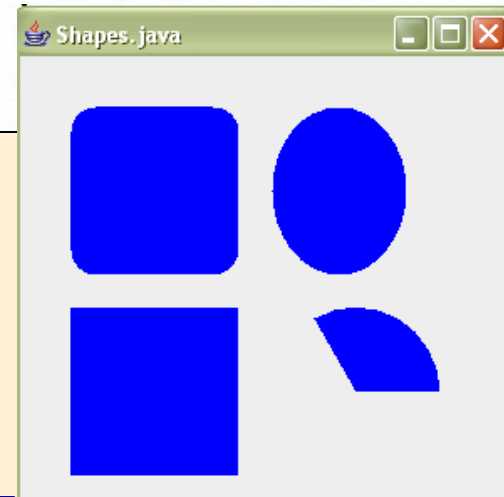
Çizgiler ve Eğriler

- Line2D
 - Bir çizgi double,float veya Point2D parametre alabilir.
- QuadCurve2D
 - Bir çizgi double,float veya Point2D parametre alabilir.
 - Tek kontrol noktası vardır.
- CubicCurve2D
 - Bir çizgi double,float veya Point2D parametre alabilir.
 - İki kontrol noktası vardır.

Şekiller

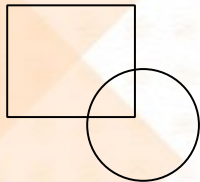
- Rectangle2D, RoundedRectangle2D, Ellipse2D, Arc2D
 - Arc2D için 3 closure seçeneği vardır; CHORT, PIE ve OPEN

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    Graphics2D g2d = (Graphics2D)g;  
    g2d.setColor(Color.blue);  
    RoundRectangle2D.Double rrect = new  
RoundRectangle2D.Double(30,30,100,100,30,30);  
    Rectangle2D.Double rect = new Rectangle2D.Double(30,150,100,100);  
    Ellipse2D.Double ellipse = new Ellipse2D.Double(150,30,80,100);  
    Arc2D.Double arc = new Arc2D.Double(150,150,100,100,0,120,Arc2D.PIE);  
    g2d.fill(rrect);  
    g2d.fill(rect);  
    g2d.fill(ellipse);  
    g2d.fill(arc);  
}
```

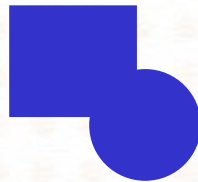


Constractive Area Geometry

- Basit şekilleri özel tarzlarda birleştirerek kompleks şekiller oluşturmayı sağlar.
 - Toplama, çıkarma, kesiştirme, XOR
- Bir Area oluşturulur ve bunun üstünde tüm işlemler yapılır



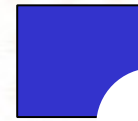
Orjinal
Şekiller



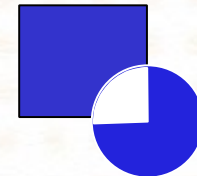
Toplama



Kesiştirme



Çıkarma



Xor