



Bölüm 8: Grafikler

JAVA ile Nesne Yönelimli Programlama



Grafik Programlama

- Java, grafik programlama için güçlü ve kapsamlı bir araç seti sunar.
- Etkileşimli ve görsel uygulamalar geliştirmek mümkündür.
- Bileşenler, olay işleme gibi konuların anlaşılmasını gerektirir.
- **Bileşenler** (components), pencere, düğme, etiket gibi kullanıcı arayüzünün temel yapı taşlarıdır.
- **Olay işleme** (event handling), fare tıklamaları, klavye tuşları gibi kullanıcının etkileşimde bulunduğu olayları yakalar.
- **Resimleme** (painting), grafik nesneleri çizme, ekranın güncellenmesi ve bileşenlerin görüntülenmesini sağlar.



Uygulama Adımları

- Java'nın grafik sınıfları kullanılarak arayüze bileşenler eklenir.
- Kullanıcının eylemlerine yanıt veren olay işleme fonksiyonları tanımlanır.
- `paint()` veya `paintComponent()` gibi metotlar kullanılarak resimleme işlemleri gerçekleştirilir.



Temel Grafik Kavramları

- Canvas (Tuval):
 - Resim çizimine olanak tanıyan bir tuval.
 - JPanel veya Canvas gibi bileşenler üzerinde grafik çizilir.
- Graphics Sınıfı:
 - Grafik işlemlerini gerçekleştirmek için kullanılır.
 - Çizim metotları içerir: `drawLine()`, `drawRect()`, `drawImage()` gibi.
- Renk ve Font Kullanımı:
 - Grafik nesnelerini renklendirmek için `Color` sınıfı kullanılır.
 - Metinleri özelleştirmek için `Font` sınıfı kullanılır.



Grafik Nesnesi Oluşturma

- `Graphics g = panel.getGraphics();` şeklinde bir grafik nesnesi elde edilir.
- `g.drawLine()`, `g.drawRect()`, `g.drawImage()` gibi metotlar ile çizim yapılır.
- `g.setColor()`, `g.setFont()` gibi metotlar ile renk ve font özellikleri ayarlanır.

```
public void paintComponent(Graphics g) {  
    g.setColor(Color.BLUE);  
    g.drawRect(20, 30, 50, 70);  
    g.drawString("Merhaba, Java Grafik!", 100, 100);  
}
```



JFrame Sınıfı

- JFrame sınıfından bir nesne oluşturarak temel bir pencere oluşturulur.
- JFrame pencere = new JFrame("Başlık"); gibi.
- Pencereye diğer bileşenleri eklemek için kullanılır.
- pencere.add(bilesen); şeklinde.
- pencere.setVisible(true); ile oluşturulan pencereyi görünür hale getirir.
- setSize(width, height) ile pencerenin boyutu ayarlanır.
- pack() ile içerik boyutuna uygun olarak pencere boyutu ayarlanır.
- setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) ile pencere kapatıldığında programın sonlandırılmasını sağlar.
- setLocation(x, y) ile pencerenin konumu belirlenir.



JFrame Sınıfı

```
JFrame pencere = new JFrame("Java JFrame Örneği");  
pencere.setSize(400, 300);  
pencere.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
pencere.setVisible(true);
```



JPanel Sınıfı

- Swing kütüphanesinde bulunur.
- Bir pencerenin içindeki bileşenleri gruplamak ve düzenlemek için kullanılır.
- Diğer GUI öğelerini içinde barındıran bir konteynerdir.
- setBackground(), setBorder() gibi metodlar ile renk, arka plan, boyut gibi özellikleri özelleştirilebilir.
- Hareketli veya dinamik içerikleri içinde barındırabilir.
- Animasyonlar, grafikler gibi öğeleri içine yerleştirebilir.
- Kodunuzu parçalara ayırmak ve yönetmek için kullanılır.
- Birden fazla JPanel kullanarak bileşenleri düzenlemek kolaylaşır.



JPanel Sınıfı

```
JPanel panel = new JPanel();  
panel.setBackground(Color.LIGHT_GRAY);  
panel.add(new JButton("Click Me"));
```



Point Sınıfı

- Bir noktanın (x, y) koordinatlarını temsil eder.
- Point(int x, int y) constructor'ı ile nokta oluşturulur.
- Ekran koordinatları, nesneleri yerleştirme ve hareket ettirme için kullanılır.
- setLocation(x, y): Noktanın koordinatlarını değiştirir.
- distance(x1, y1, x2, y2): İki nokta arasındaki uzaklığı hesaplar.

```
Point nokta = new Point(30, 40);  
double xKoordinati = nokta.getX();  
double yKoordinati = nokta.getY();
```



Color Sınıfı

- Arayüzün dinamik olarak renklendirilmesini sağlar.
- Temel renkleri ve RGB değerlerini içerir.
 - Temel: `Color.RED`, `Color.GREEN`, `Color.BLUE` gibi önceden tanımlanmış renk sabitleri.
 - RGB: Kırmızı, yeşil ve mavi bileşenleri belirten değerler.
- `new Color(int red, int green, int blue)` constructor'ı ile özel renk oluşturulur.
- `brighter()` ve `darker()`: Renk parlaklığını değiştirmek için kullanılır.



BorderLayout Sınıfı

- Bileşenlerin yerleşim yöneticisidir.
- Farklı öncelikteki bileşenleri belirli bir düzende göstermeye yarar.
- Belirli bir bölgeye (NORTH, SOUTH, WEST, EAST, CENTER) yerleştirilir.
- Bileşen bir bölgeye eklenmezse, varsayılan olarak merkeze yerleştirilir.
- `setHgap()` ve `setVgap()` metotları ile yatay ve dikey boşluklar ayarlanır.



BorderLayout Sınıfı

```
JPanel panel = new JPanel();
```

```
BorderLayout layout = new BorderLayout();  
panel.setLayout(layout);
```

```
panel.add(new JButton("Kuzey"), BorderLayout.NORTH);  
panel.add(new JButton("Güney"), BorderLayout.SOUTH);  
panel.add(new JButton("Batı"), BorderLayout.WEST);  
panel.add(new JButton("Doğu"), BorderLayout.EAST);  
panel.add(new JButton("Merkez"), BorderLayout.CENTER);
```



Giris Yasaktir Isareti





Giris Yasaktir Isareti

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    var c = new Point(getWidth()/2, getHeight()/2); // merkez  
    var r = Math.min(getWidth()/2, getHeight()/2) - 5; //yarıçap  
    var r2 = (int)(r * 0.9); // iç yarıçap  
    var bw = (int)(r2 * 1.4); // bar genişlik  
    var bh = (int)(r2 * 0.35); // bar yükseklik  
    g.setColor(Color.WHITE);  
    g.fillOval(c.x - r, c.y - r, 2*r, 2*r);  
    g.setColor(Color.RED);  
    g.fillOval(c.x - r2, c.y - r2, r2 * 2, r2 * 2);  
    g.setColor(Color.WHITE);  
    g.fillRect(c.x - bw/2, c.y - bh/2, bw, bh);  
}
```



GirisYasaktirIsareti

```
public static void main(String[] args) {  
    SwingUtilities.invokeLater(() -> {  
        var panel = new GirisYasaktirIsareti();  
        panel.setBackground(Color.GREEN.darker());  
        var frame = new JFrame("Basit bir grafik programı");  
        frame.setSize(400, 300);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.getContentPane().add(panel, BorderLayout.CENTER);  
        frame.setVisible(true);  
    });  
}
```




Etkileşimli Bilgisayar Grafikleri

- Bilgisayar grafikleri, görsel içerik oluşturma ve manipüle etme sürecidir.
- İnteraktif bilgisayar grafikleri, kullanıcının etkileşimde bulunabildiği grafik uygulamalarını ifade eder.
- Fare, klavye gibi giriş aygıtlarıyla etkileşimli grafik uygulamaları geliştirilir.
- Java grafik kütüphaneleri aracılığıyla şekil çizme, renklendirme ve grafik nesneleri manipüle etmeyi sağlar.



MouseAdapter Sınıfı

- Fare olaylarını dinlemek ve işlemek için kullanılır.
- MouseListener arayüzünü uygular.
- `mouseClicked()`, `mousePressed()`, `mouseReleased()`, `mouseEntered()`, `mouseExited()` gibi fare olaylarına tepki gösteren metotları içerir.



MouseEvent Sınıfı

- Fare etkileşimi sonrasında ortaya çıkan olayları temsil eder.
- Fare olayları: `mouseClicked`, `mousePressed`, `mouseReleased`, `mouseEntered`, `mouseExited`.
- Fare Pozisyonu:
 - `getX()` ve `getY()` metotları ile fare pozisyonu alınır.
- Düğme Durumu:
 - `getButton()` metodu kullanılan fare düğmesini söyler.
- Tuş Kontrolleri:
 - `isAltDown()`, `isCtrlDown()`, `isShiftDown()` gibi metotlar ile tuş durumları kontrol edilir.

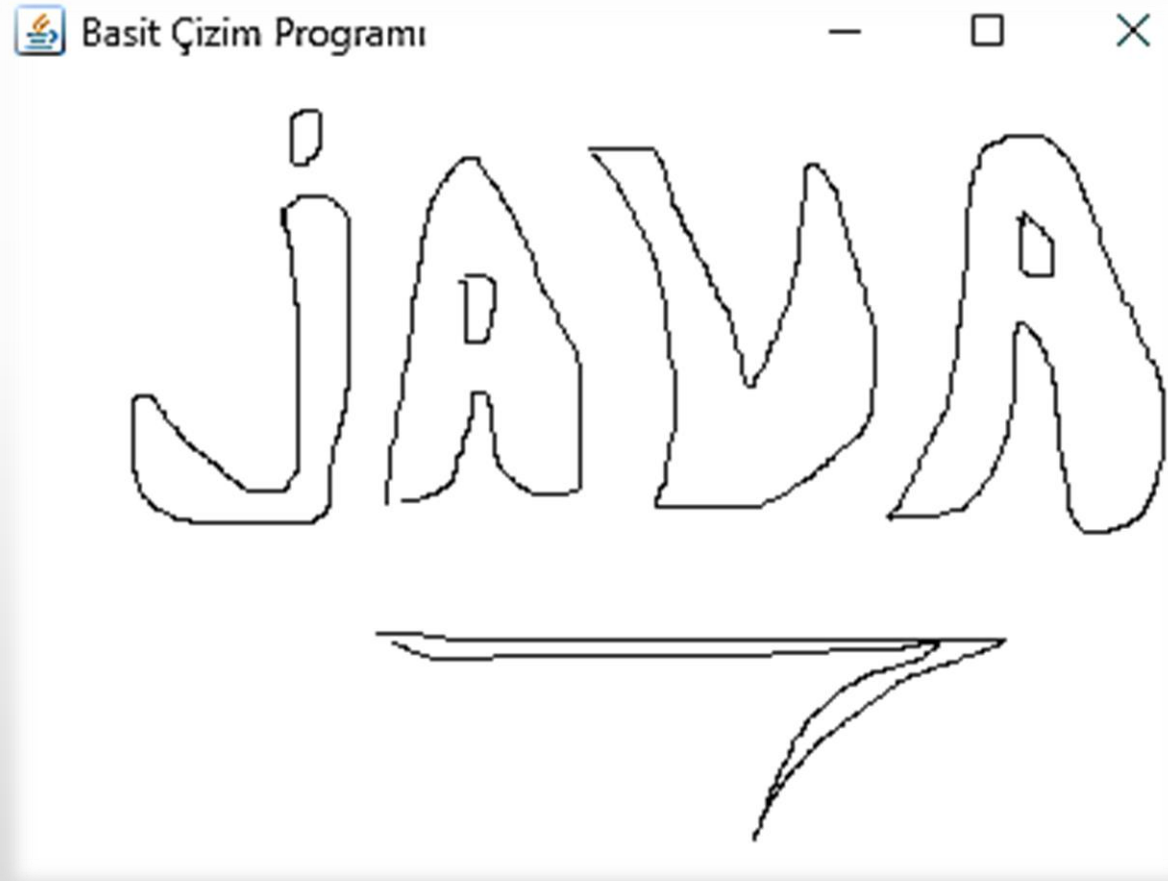


SwingUtilities Sınıfı

- javax.swing paketinde bulunur.
- Çoklu thread kullanımında ortaya çıkabilecek sorunları önler.
- invokeLater():
 - Bir işlemi Swing olay işleme thread'inde asenkron olarak çalıştırır.
- invokeAndWait():
 - Bir işlemi Swing olay işleme thread'inde senkron olarak çalıştırır.



BasitCizimPaneli





BasitCizimPaneli

```
addMouseMotionListener(new MouseMotionAdapter() {  
    public void mouseDragged(MouseEvent e) {  
        egriler.get(egriler.size() - 1).add(  
            new Point(e.getX(), e.getY()));  
        repaint(0, 0, getWidth(), getHeight());  
    }  
});
```

```
addMouseListener(new MouseAdapter() {  
    public void mousePressed(MouseEvent e) {  
        var yeniEgri = new ArrayList<Point>();  
        yeniEgri.add(new Point(e.getX(), e.getY()));  
        egriler.add(yeniEgri);  
    }  
});
```



BasitCizimPaneli

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
  
    for (var egri : egriler) {  
        var onceki = egri.get(0);  
        for (var nokta : egri) {  
            g.drawLine(onceki.x, onceki.y, nokta.x, nokta.y);  
            onceki = nokta;  
        }  
    }  
}
```



BasitCizimPaneli

```
public static void main(String[] args) {  
    SwingUtilities.invokeLater(() -> {  
        var frame = new JFrame("Basit Çizim Programı");  
        frame.getContentPane().add(new BasitCizimPaneli(),  
                                    BorderLayout.CENTER);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(400, 300);  
        frame.setVisible(true);  
    });  
}
```




JButton Sınıfı

- Kullanıcı tarafından tıklanabilir, etkileşimli düğmeler oluşturur.
- Metin, ikon ve olay işleyicileri kullanarak özelleştirilebilir.
- setText() metodu ile düğme üzerinde görünen metin belirtilir.
- setIcon() metodu ile düğmeye bir ikon eklenebilir.
- addActionListener() metodu ile düğmeye tıklanma olaylarını dinleyen bir olay işleyici eklenebilir.



JScrollPane Sınıfı

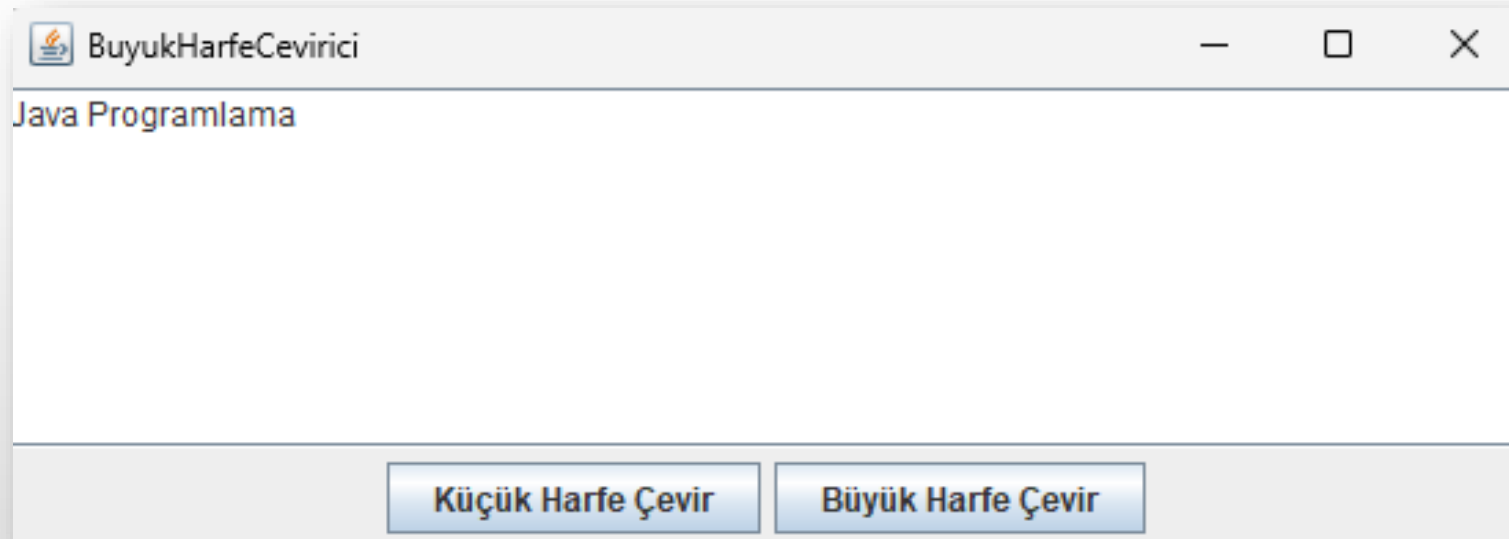
- Metin alanları, tablolar gibi içerikleri kullanıcıya göstermek için kullanılır.
- İçerik ekrana sığmadığında otomatik olarak kaydırma çubukları ekler.
- Kaydırma çubukları, kullanıcıya içeriğin farklı bölümlerini gösterir.
- İçeriğin boyutu değişikliklerine otomatik uyum sağlar.
- `setPreferredSize()` veya `setViewportView()` gibi metotlarla özelleştirilebilir.



JTextArea Sınıfı

- Çok satırlı metin alanları oluşturur.
- Metin içeriği dinamik olarak okunabilir, atanabilir ve güncellenebilir.
- Metin eklemek, silmek, okumak ve düzenlemek için bir dizi metot içerir.
- Belirli bir satır ve sütun sayısı ile önceden tanımlanabilir.
- `getText()`, `setText()`, `append()` gibi metotlarla metin içeriğini yönetir.

BuyukHarfeCevirici





BuyukHarfeCevirici

```
SwingUtilities.invokeLater(() -> {  
    var alan = new JTextArea("Java Programlama", 8, 50);  
    var küçükHarfButton = new JButton("Küçük Harfe Çevir");  
    var büyükHarfButton = new JButton("Büyük Harfe Çevir");  
    küçükHarfButton.addActionListener(e -> alan.setText(  
        alan.getText().toLowerCase()));  
    büyükHarfButton.addActionListener(e -> alan.setText(  
        alan.getText().toUpperCase()));  
  
    var panel = new JPanel();  
    panel.add(küçükHarfButton);  
    panel.add(büyükHarfButton);  
});
```



BuyukHarfeCevirici

```
var frame = new JFrame("BuyukHarfeCevirici");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.getContentPane().add(new JScrollPane(alan),
                                BorderLayout.CENTER);
frame.getContentPane().add(panel, BorderLayout.SOUTH);
frame.pack();
frame.setVisible(true);
});
```



JLabel Sınıfı

- Metin, resim gibi bir içeriği görüntülemek için kullanılır.
- `setText()` metodu ile görüntülenecek metin belirlenir.
- `setIcon()` metodu ile görüntülenecek resim belirlenir.
- Metin stili (`setFont()`) ve rengi (`setForeground()`) özelleştirilebilir.



JTextField Sınıfı

- Kullanıcılara metin girmeleri için bir alan sağlar.
- Formlardan veya kullanıcı girişlerinden veri toplamak için kullanılır.
- `getText()` ve `setText()` metotları ile metni okuma ve atama yapılır.
- `setColumns()` metodu ile görünen karakter sütun sayısı belirlenir.
- `addActionListener()` metodu ile kullanıcının hareketleri dinlenir.



ParaUstuHesaplama

Para Üstü Hesaplama

Miktar:

93 kuruş yapmak için kullanılacaklar:

- 1 50 kuruş
- 1 25 kuruş
- 1 10 kuruş
- 1 5 kuruş
- 3 1 kuruş



ParaUstuHesaplama

```
public ParaUstuHesaplama() {  
    var ustPanel = new JPanel();  
    ustPanel.add(new JLabel("Miktar:"));  
    ustPanel.add(miktarAlani);  
    getContentPane().add(ustPanel, BorderLayout.NORTH);  
    getContentPane().add(new JScrollPane(rapor),  
                           BorderLayout.CENTER);  
    setBackground(Color.LIGHT_GRAY);  
    rapor.setEditable(false);  
    metin.addDocumentListener(new DocumentListener() {}  
});
```



ParaUstuHesaplama

```
void raporGuncelle() {  
    try {  
        var miktar = Integer.parseInt(metin.getText(0, getLength()));  
        int dizi[] = {50, 25, 10, 5, 1};  
        rapor.setText(miktar + " kuruş için kullanılacaklar:\n");  
        for(int x: dizi) {  
            rapor.append(miktar / x + " " + x + " kuruş\n");  
            miktar %= x;  
        }  
    } catch (NumberFormatException e) {  
        rapor.setText("Tam sayı değil veya aralık dışında");  
    } catch (Exception e) {  
        rapor.setText(e.toString());  
    }  
}
```



ParaUstuHesaplama

```
public static void main(String[] args) {  
    SwingUtilities.invokeLater(() -> {  
        var frame = new ParaUstuHesaplama();  
        frame.setTitle("Para Üstü Hesaplama");  
        frame.pack();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setVisible(true);  
    });  
}
```



SON