



# Bölüm 8: Grafikler

## JAVA ile Nesne Yönelimli Programlama



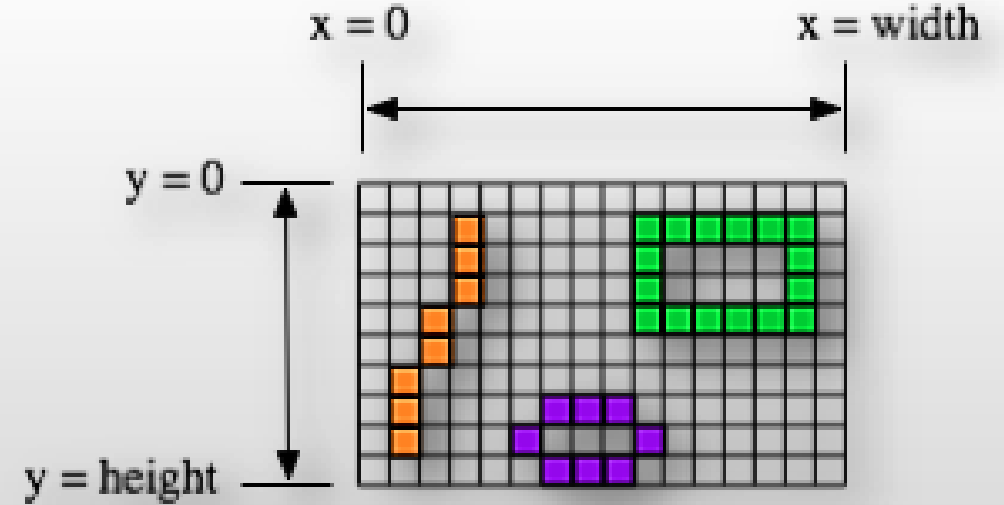
# Grafik Programlama

- 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 tuşlarına basma gibi kullanıcının etkileşimde bulunduğu olayları yakalar.
- **Görsel oluşturma** (painting), grafik nesnelerini çizme, ekranı güncelleme 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 görsel oluşturulur.





# Temel Grafik Kavramları

- Canvas (Tuval):
  - Resim çizimine olanak tanır.
  - *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

- Pencere oluşturmak için kullanılır.
- `JFrame pencere = new JFrame("Başlık");`
- Pencereye bileşen eklemek için kullanılır.
- `pencere.add(bilesen);`
- `pencere.setVisible(true);` ile oluşturulan pencere görünür hale getirilir.
- `setSize(width, height)` ile pencerenin boyutu ayarlanır.
- `pack()` ile içerik boyutuna uygun olarak pencere boyutu ayarlanır.
- `setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)` pencere kapatılınca 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.
- GUI öğelerini içinde barındıran bir konteynerdir.
- *setBackground()*, *setBorder()* gibi metodlar ile renk, arka plan, boyut gibi özellikleri özelleştirilebilir.
- Hareketli veya dinamik içerikleri içinde barındırabilir.
- Animasyon, grafik gibi öğeler içine yerleştirilebilir.
- Kodun parçalara ayrılarak yönetimi için kullanılır.
- Birden fazla *JPanel* kullanılarak bileşenler kolayca düzenlenebilir.





# 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)* yapıcı metodu ile nokta nesnesi oluşturulur.
- Ekran koordinatları, nesnelere yerleştirme ve hareket ettirme için kullanılır.
- *setLocation(x, y)* Noktanın koordinatları değiştirilir.
- *distance(x1, y1, x2, y2)* İki nokta arasındaki uzaklık hesaplanır.

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



# Color Sınıfı

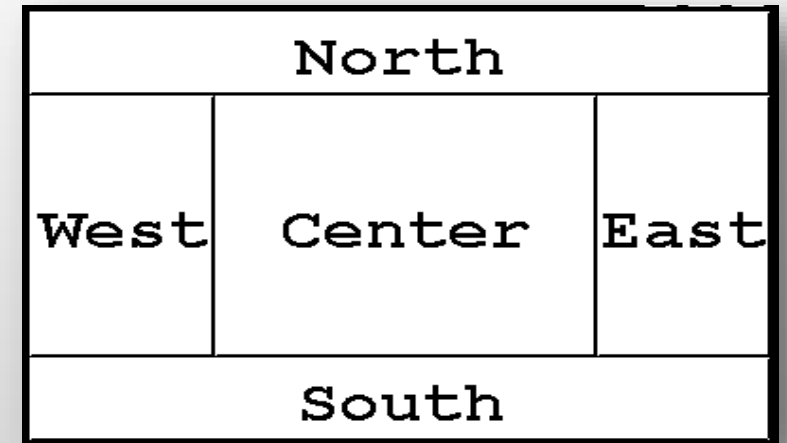
- Kullanıcı arayüzünün dinamik olarak renklendirilmesini sağlar.
- Temel renkleri ve RGB değerlerini içerir.
  - **Temel:** *Color.RED*, *Color.GREEN*, önceden tanımlanmış renk sabitleri.
  - **RGB:** Kırmızı, yeşil ve mavi oranlarını belirten değerler.
- *new Color(int red, int green, int blue)* özel renk oluşturulur.
- *brighter()* ve *darker()* Renk parlaklığını değiştirir.





# BorderLayout Sınıfı

- Bileşenleri yerleştirmek için kullanılır.
- 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 yerlesim = new BorderLayout();  
panel.setLayout(yerlesim);
```

```
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);  
}
```



# Giris Yasaktir Isareti

```
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 işleme sürecidir.
- İnteraktif bilgisayar grafikleri, kullanıcının etkileşimde bulunabildiği uygulamaları ifade eder.
- Fare, klavye gibi giriş aygıtlarıyla etkileşimli grafik uygulamaları geliştirilir.
- Grafik kütüphaneleri aracılığıyla şekil çizme, renklendirme ve grafik nesnelerini 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.
- *getX()* ve *getY()* metotları ile fare işaretçisi konumu alınır.
- *getButton()* metodu kullanılan fare düğmesini söyler.
- *isAltDown()*, *isCtrlDown()*, *isShiftDown()* metotları ile tuş durumları alınır.

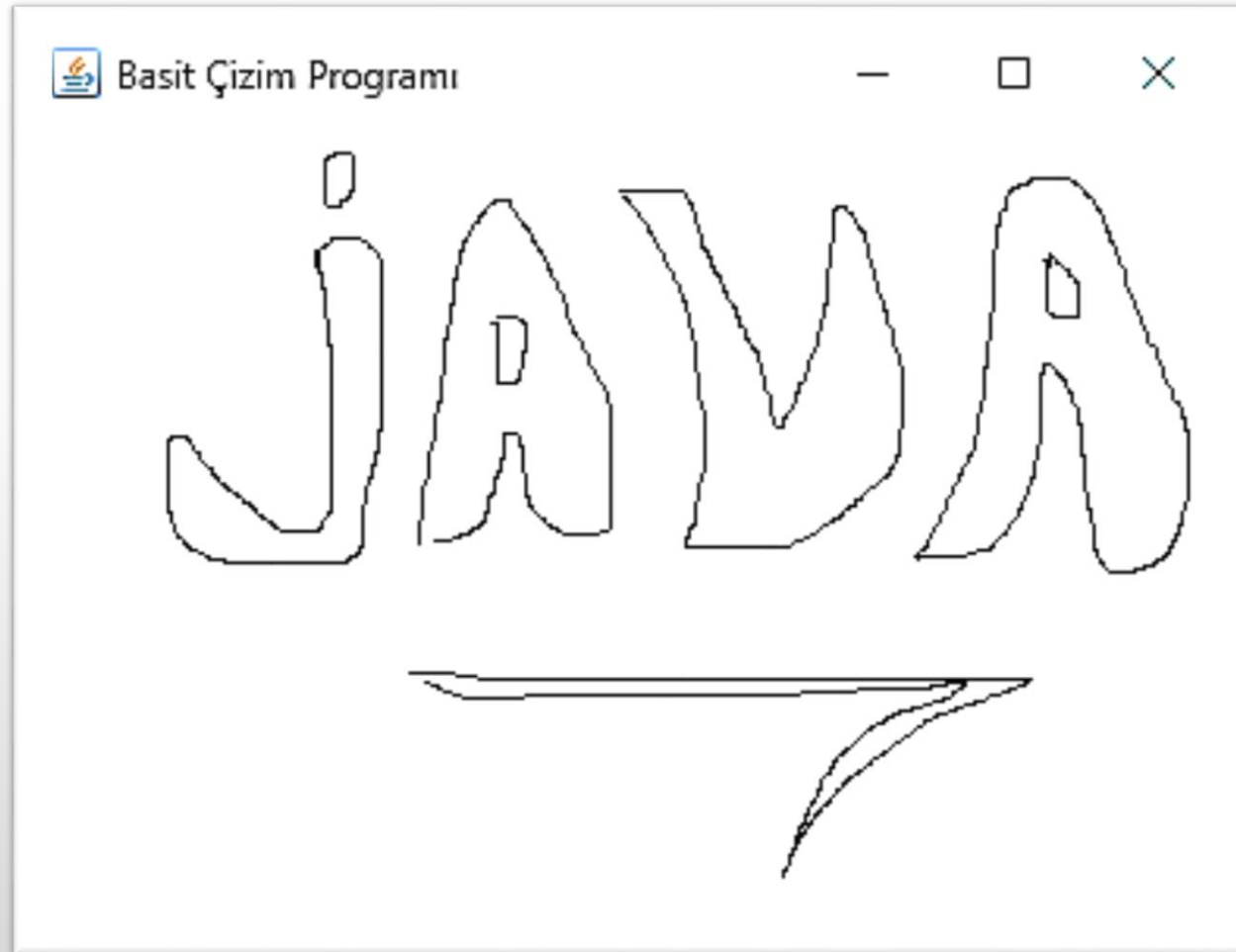


# SwingUtilities Sınıfı

- javax.swing paketinde bulunur.
- Çoklu iş parçacığı ortamında ortaya çıkabilecek sorunları önler.
- **invokeLater()**
  - Bir işlemi *Swing* olay işleme iş parçacığında *asenكرون* olarak çalıştırır.
- **invokeAndWait()**
  - Bir işlemi *Swing* olay işleme iş parçacığında *senكرون* olarak çalıştırır.



# BasitCizimPaneli





# BasitCizimPaneli

```
addMouseMotionListener(new MouseMotionAdapter() {  
    public void mouseDragged(MouseEvent e) {  
        egriler.get(egriler.size()).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()* ile düğmeye tıklanınca çalışacak olay işleyici eklenir.



# JScrollPane Sınıfı

- Metin alanı, tablo gibi içerikleri kullanıcıya göstermek için kullanılır.
- İçerik ekrana sığmadığında otomatik olarak kaydırma çubukları ekler.
- Kaydırma çubukları, kullanıcıya içeriğin farklı bölümlerini gösterir.
- İçerik boyutu değişikliklerine otomatik uyum sağlar.
- *setPreferredSize()* veya *setViewportView()* metotları ile özelleştirilebilir.

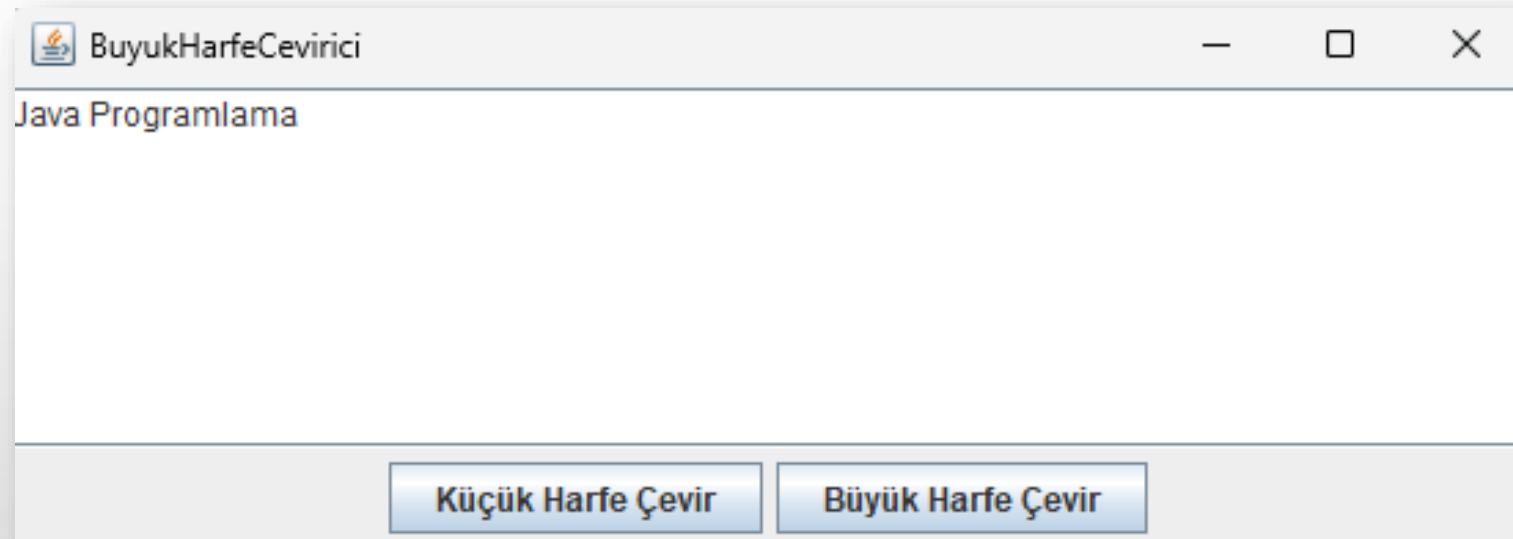


# JTextArea Sınıfı

- 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), CENTER);
frame.getContentPane().add(panel, BorderLayout.SOUTH);
frame.pack();
frame.setVisible(true);
});
```



# JLabel Sınıfı

- Metin, resim gibi içerikleri görüntülemek için kullanılır.
- *setText()* metodu ile görüntülenecek metin belirlenir.
- *setIcon()* metodu ile görüntülenecek resim belirlenir.
- *setFont()* ile metin fontu özelleştirilebilir.
- *setForeground()* ile metin rengi 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), CENTER);  
    setBackground(Color.LIGHT_GRAY);  
    rapor.setEditable(false);  
    metin.addDocumentListener(new DocumentListener() {}  
});
```



# ParaUstuHesaplama

```
void raporuguncelle() {
    try {
        var miktar = Integer.parseInt(metin.getText(0, getLength()));
        int dizi[] = {50, 25, 10, 5, 1};
        rapor.setText(miktar + " kuruş için:\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