

# Händelsehantering i Java, Events

## Uppgift 1, ActionListener

I filen ButtonApp skapas en JFrame med en panel som innehåller en knapp.

a) Skapa en lyssnarklass som implementerar ActionListener, lyssnaren kan till att börja med skriva ut ett meddelande i terminalfönstret när knappen trycks. Lägg till kod som skapar ett lyssnarobjekt och kopplar detta till händelsekällan, knappen.

b) Lägg till en av typen JLabel (en etikett) i panelen: JLabel label = new JLabel("Clicks: 0"); Ändra i lyssnarklassen så att varje gång knappen trycks uppdateras labeln (setText(String)) med det totala antalet tryckningar.

Notera att lyssnaren måste ha en referens till panelen eller labeln för att kunna göra detta.

## Uppgift 2, ActionListener

Denna uppgift bygger vidare på uppgift 1 från övningen om swing-grafik.

Hämta och studera GUITest\_a.java (finns under lösningsförslag).

Utöka programmet så att:

- När man trycker på en knapp så skrivs knappens text ut i den label som finns i gränssnittet (actionEvent.getActionCommand() ger texten.
- När ett text matats in i textfältet och ENTER trycks skrivs denna text på labeln.

Skriv lyssnaren som:

- a) En separat klass som implementerar ActionListener.
- b) En inre klass som implementerar ActionListener.

## Uppgift 3, ActionListener

Skriv en egen applikation som omvandlar ett inmatat belopp i SKR (använd JTextField) till motsvarande belopp i US\$. Resultatet skrivs ut på en JLabel. Uträkningen ska göras då användaren trycker enter i inmatningsfältet.

Utforma applikationens gränssnitt så att det blir intuitivt för användaren.

## Uppgift 4, (Söka i API-dok)

Leta upp interfacet MouseListener i API-dokumentationen

(<http://java.sun.com/j2se/1.5/docs/api/>).

Vilka metoder finns i interfacet? I vilka situationer anropas de?

Leta upp klassen MouseAdapter. Vad gör denna klass? Vad är poängen med klassen?

## Uppgift 5, MouseListener

Hämta MouseClickPanel.java. Testkör programmet och studera koden så du förstår vad som händer. Vilken grafisk komponent har lyssnaren registrerats på?

Ändra nu i lyssnarklassen så att när knappen trycks ned och sedan släpps upp i en annan position så ritas en linje mellan dessa.

Tips: Du behöver nu två Point-objekt i klassen, start och end.

Använd metoderna mousePressed (...) för att läsa av linjens startpunkt samt mouseReleased (...) för att läsa av linjens slutpunkt. I den senare metoden anropar du också repaint() (som anropar paintComponent (...) som ritar linjen).

**Uppgift 6, KeyListener**

Skriv ett program som placerar en bild mitt på en JPanel.

Skriv sedan en tangentyssnare som flyttar bilden en pixel uppåt/nedåt/höger/vänster när respektive piltangent alternativt w/z/a/s trycks. Glöm inte att panelen måste göras fokuserbar med `setFocusable()`.

Tips: Ladda bilden i ett ImageIcon-objekt med

```
ImageIcon icon = new ImageIcon("images/devil.gif");
```

Detta objekt kan sedan ritas ut i `paintComponent(Graphics g)` med

```
icon.paintIcon(panel, g, x, y);
```