Sündmuste töötlus

Sündmuste töötlusega seotud sündmused , klassid ja liidesed paiknevad paketis *java.awt.event.* **Vastav pakett kirjekdab alates jdk1.1 kasutusele tulnud sündmuste töötlusmudelit ── nn. õiguste delegeerimise mudelit.**

- **Sündmuste tüübid**

─ tegevuse sündmused(action events).
─ hiire sündmused
─ klaviatuuri sündmused
─ akna süsteemi sündmused(window system events)
─ fokusseerimissündmused(focus events)
─ komponendi poolt genereeritavad sündmused
─ kasutaja defineeritavad sündmused
─ teised sündmused

- **Sündmuste klassid pärinevad klassist** *java.awt.AWTEvent*

- **Iga sündmuste tüübi jaoks on:**

─ sündmuste klass (*ActionEvent***)**
─ sündmustekuulamisliides (A*ctionListener***)**
─ meetodid kuulajate lisamiseks ja eemaldamiseks
addActionListener() **ja** *removeActionListener()*
─ meetod *addActionListener()* ***võimaldab komponentidel***
kuulata,genereerida ja välja saata sündmusi.

Näide

```
import java.awt.event.*;
public class Nupuke extends Applet
  {
     Button b1 = new Button ("Alusta");
    public void init ()  {
       b1.addActionListener ( new  B1Listener());
      add (b);
    }
  }
class B1Listener implements ActionListener {
     public void actionPerformed(ActionEvent e) {
     Toolkit.getDefaultToolkit().beep();

    }

  }
```

Kõrgtaseme sündmused

Sündmused , mis tekivad siis ,kui kasutaja klikib mingil graafilise kasutajaliidese komponendil.

Sündmused:

- *ActionEvent* – **näiteks nupuvajutus**
- *AdjustmentEvent* – **kasutaja valib midagi numbrilises vahemikus (kerimisriba)**
- *ItemEvent* – **kasutaja valib elemendi listist**

Kuulajad:

– *ActionListener*
meetod *ActionPerformed(ActionEvent e)*

```
class Nupukuulaja implements ActionListener {

     public void actionPerformed(ActionEvent e) {

       Object obj = e.getSource();

         If (obj == startButton)
         //...
         else if ( obj == stopButton)
        //...

    }

  }
```

– *AdjustmentListener*
meetod adjustmentValueChanged(AdjustmentEvent e)
– *ItemListener*
meetod itemStateChanged(Itemevent e)

Liides *AdjustmentListener*

Vastavate sündmuste töötlemiseks peab vastav klass realiseerima liidest *AdjustmentListener*. **Vastav sündmus genereeritakse juhul ,kui komponent või selle element valitakse kerimisriba abil.**

AdjustmentListener **on meetodi** *addAdjustmentListener()* **argumendiks.**

Liides defineerib meetodeid:

- *AdjustmentValueChanged(AdjustmentEvent e)*
- ***sündmuse tüüp määratakse meetodiga*** *GetAdjustmentType()*

– meetoodi poolt tagastatav väärtus
UNIT_INCREMENT,UNIT_DECREMENT,
BLOCK_DECREMENT , BLOCK_INCREMENT ***ja*** *TRACK*

Elemendisündmused – Item Events

Elemendisündmuste töötleja realiseerib *ItemListener* **liidest.**
Need sündmused leiavad aset kui toimub elementide valik või selle annul-leerimine List- , Choice- , Checkbox- tüüpi objektidest või menüüdest.
ItemListener on argumendiks meetodile *addItemListener().*
Liides *ItemListener* **defineerib meetodit*:***

ItemStateChangeged (ItemEvent e)

– sündmuse tüüp määratakse meetodiga
itemEvent.getStateChange().

meetodi poolt tagastatavad väärtused
SELECTED **ja** *DESELECTED*

Madalataseme sündmused

Neid registreerivad vastavad liidesed ja neid töödeldakse liidestes kirjeldatud meetotitega.

Madalataseme liidesed ja vastavad meetodid

- **Liides** *ComponentListener*

componentHidden (ComponentEven et)
componentMoved (ComponentEvent e)
componentRezised (ComponentEvent e)
componentShown(Componentevent e)

- ***L iides*** *KeyListener*

keyPressed (keyEvent e)
keyReleased(KeyEvent e)
keyTyped (KeyEvent e)

- ***Liides*** *MouseListener*

mouseClicked (MouseEvent e)
mouseEntered (MouseEvent e)
mouseExited(MouseEvent e)
mousePressed(MouseEvent e)
mouseReleased(MouseEvent e)

- ***Liides*** *MouseMotionListener*

mouseDragged(MouseEvent e)
mouseMoved(MouseEvent e)

- ***Liides*** *WindowListener*

windowsClosed(WindowEvent e)
windowClosing(WindowEvent e)
windowDeiconified(WindowEven e)
windowOpened(WindowEvent e)

Adapterklassid

Adapterklassid realiseerivad vastavat sünmustekuulamisliidest ning sisaldavad liidese meetodite tühje versioone.

— *ComponentAdapter* — *MouseAdapter*
— *FocusAdapter* — *MouseMotionAdapter*
— *KeyAdapter* — *WindowAdapter*

Sündmuste klassid

- **Klass** *InputEvent*

Klass on kõigi sisendsündmuste vanemklassiks.

Meetodid allavajutatud modifikaatorklahvide ja hiire nuppude kindlaks tegemiseks:

- *boolean isAltDown() // hiire parempoolne nupp*
- *boolean isControlDown() // klahv Ctrl*
- *boolean isMetaDown() // klahv Shift*
- *boolean isShiftDown() // klahv Alt*

Näide

```
  public void mousePressed (MouseEvent evt)
{
   if (evt.isMetaDown() && evt.isShiftDown() )

// ...
  }
```

Meetod *getModifiers()* **tagastab täisarvulise väärtuse. Meetod Teeb kindlaks , millisele modifikaatorklahvile või hiire nupule vajutati. Meetodi poolt tagastatavat väärtust saab võrrelda maskiga:**

- *ALT_MASK*
- *CTRL_MASK*
- *META_MASK*
- *SHIFT_MASK*
- *BUTTON1_MASK*
- *BUTTON2_MASK*
- *BUTTON3_MASK*

- **Klass** *KeyEvent*

- **Meetod** *getKeyCode()* **teeb kindlaks millisele klahvile vajutati.Meetod tagastab klahvi koodi. Kui allavajutatud klahv on eriklahv, siis meetodi poolt tagastav täisarv vastab klassis** *KeyEvent* **defineeritud konstandile. Mõningad konstandid:**

KeyEvent.VK_UNDEFINED, KeyEvent.VK_SPACE

```
int getKeyCode()
```

- ***Meetod*** *isActionKey()* **teeb kindlaks , kas vajutatud klahv on funktsiooniklahv.**

```
boolean  isActionKey()
```

- **Meetod** *getKeyChar()* **tagastab allavajutatud klahviga seotud märgi koodisüstemis** *UNICODE*.

```
char getKeyChar()
```

Näide

```
public void keyReleased(KeyEvent evt)

{

  int keyCode = evt.getKeyCode();
 if (keyCode == KeyEvent.VK_F1)
// ...

 }
```

- **Klass** *MouseEvent*

Hiire positsiooni saab määrata meetodiga *getPoint()* **või meetoditega** *getX()* **ja** *getY()*.

int getX() **määrab hiire X koordinaadi.**
int getY() **määrab hiire Y koordinaadi.**

Point getPoint() **määrab ära punkti(X,Y).**

Meetod getClickCount() tagastab sündmusega seotud hiireklikkide arvu.

int getClickCount()

Näide

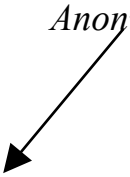
```
public void mouseReleased(MouseEvent e)
{
   if ( e.isCtrlDown() && e.getClickCount() == 2)

// ...
}
```

- ***Klass*** *WindowEvent*

Aknasündmused imformeerivad akent sulgemisest , ikooniks tegemisest või avamisest.
Anonüümse klassiga näide.

```
public class Aken   {
  public static void main(String[] args)
  {
    Frame k = new Frame("Aken");
      k.addWindowListener(
      new WindowAdapter()  {
public void windowClosing(  WindowEvent e)
  {
     System.exit(0);
      }
// ...
}
}
```

Anonüümne siseklass



Näide siseklassi ja adapterklassi kasutamise kohta

```
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;
public
class Mouse extends Applet {

   public void init()
   {
         // vaikimisi valge värv
      setBackground(Color.white);

         // Lisame appletile kuulaja
      addMouseListener(new Watcher());
   }
   // Siseklassi Watcher kirjeldus

   class Watcher extends MouseAdapter {

   /* Klass Watcher pärineb klassist MouseAdapter, mis
   *  realiseerib vaikimisi liidest MouseListener. Klassis
   * on meetodite mouseEntered() ja mouseExited()
   * kirjeldused. Meetodeid  mouseClicked() , mousePressed()
   *  ja mouseReleased ei ole vaja esitada
      */
```

```
        public void mouseEntered(MouseEvent evt)
        {
           setBackground(Color.yellow);
        }
              public void mouseExited(MouseEvent evt)
        {
           setBackground(Color.red);
        }
     }
  public void paint(Graphics g)
     {
        Color color = getBackground();
        if(color.equals(Color.red))
          g.setColor(Color.yellow);
        else
          g.setColor(Color.red);
         Dimension s = getSize();
        int w = s.width,
            h = s.height;
        g.drawOval(0, 0, w-1, h-1);
     }
}
```

Applet, mis töötleb hiire sündmusi ilma adapterklassi kasutamata

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
//Applet Mouse1 realiseerib liidest MouseListener
public
class Mouse1 extends Applet
   implements MouseListener {

   private int r = 30,
               d = 2*r;
// Kirjeldame graafika objekti g

   private Graphics g;
```

```
    public void init()
    {


        g = getGraphics();
        addMouseListener(this);
    }

    public void mouseReleased(MouseEvent evt)
    {
        int x = evt.getX(),
            y = evt.getY();

        gDC.setColor(Color.red);
        gDC.fillOval(x-r, y-r, d, d);
        gDC.setColor(Color.black);
        gDC.drawOval(x-r, y-r, d-1, d-1);
    }

        // liidese  MouseListener tühjad meetodid

    public void mousePressed(MouseEvent evt)
    {
    }
    public void mouseClicked(MouseEvent evt)
    {
    }
    public void mouseEntered(MouseEvent evt)
    {
    }
    public void mouseExited(MouseEvent evt)
    {
    }
}
```