

Javaprogrammering - Del 4

- Inl2 - genomgång med UML-grafer
- Grafisk applikation + Javadoc
- Mera Applet
- Grafiska gränssnitt, händelsestyrning
- Grafikhantering (omritning)

1

Luddigt krav

- ”Det ska finnas möjlighet att monitorera serverns aktiviteter”
 - grafiskt fönster
 - CPU-utnyttjande
 - diskutnyttjande
 - nätverkslast per klient
 - NEJDÅ
- se upp för luddiga krav

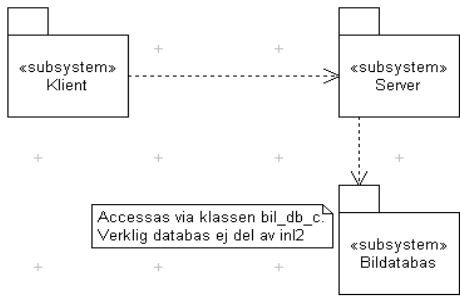
2

OO-analys Java Inl2

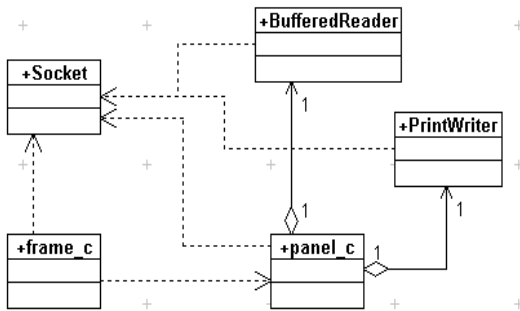
- paketdiagram
 - indelning i subsystem
- klassdiagram
- objektdiagram
- sekvensdiagram

3

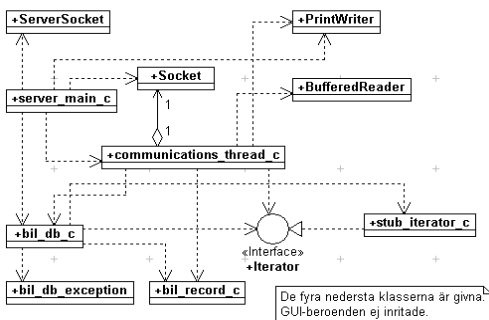
Bil-DB subsystem



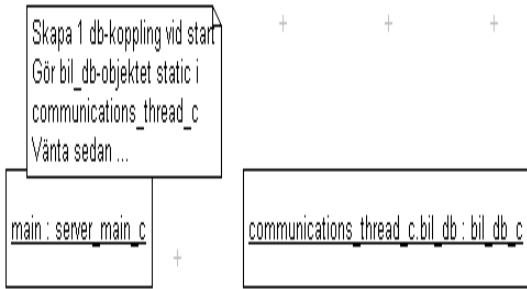
Klassdiagram - klient



Klassdiagram - Server

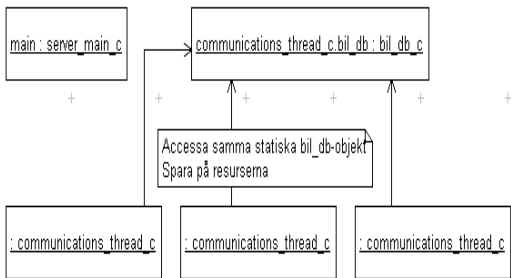


Objektdiagram - Server start



7

Objektdiagram - Server full



8

Winapp/Javadoc

- applikationer svårt
 - nej, som att skriva applets . . .
- JAVADOC
 - ger HTML-dok

9

Kodexempel

- Roten ur i en applikation
- å' så JAVADOC
 - ger HTML-dokumentation
- Special
 - Applet i en applikation

10

Windowsapplikation

- startas från main-metod
 - till skillnad från en applet
- skaffa en Frame
 - ramar in huvudfönster
- lägg en Panel i Frame
 - för utplacering av kontroller
- kör show på framen

11

Start/Avslut

- start från DOS-fönster
- avslut om inget görs
 - Ctrl-C krävs i DOS-fönstret
- glöm därför inte
 - addWindowListener
 - windowClosing-hantering
 - System.exit(0)

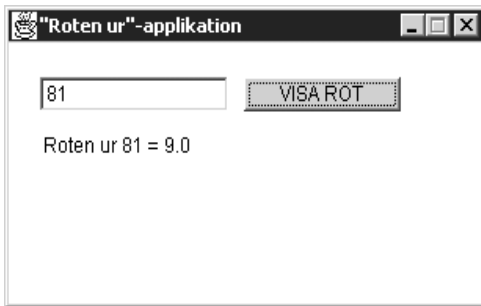
12

sqrt_c som Panel

- ta bort import java.applet.Applet
 - behövs inte i en applikation
- class sqrt_c extends Panel
 - istället för extends Applet
- public sqrt_c()
 - konstruktor
 - ERSÄTTER init i Appletklassen

13

frame_c - Körexempel



14

frame_c - 1

```
import java.awt.Frame;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

/**
 * Klassen innehåller en mainfunktion
 * som kan fungera som en mall för hur
 * man bygger Windowsapplikationer i
 * Java.
 */
public class frame_c
{
```

15

frame_c - 2

```
/**
 Skapar en frame och kör en Panel i ramen

@param args
    Utnyttjas inte

@return
    status till operativsystemet
*/
public static void main( String args[] )
{
```

16

frame_c - 3

```
Frame frame = new Frame(
    "\"Roten ur\"-applikation" );

sqrt_c panel = new sqrt_c();
panel.setBounds( 10, 10, 300, 150 );
frame.add( panel );
frame.pack();
frame.setResizable( false );
```

17

frame_c - 4

```
frame.addWindowListener(
    new WindowAdapter()
    {
        public void windowClosing(
            WindowEvent wev )
        {
            System.exit( 0 );
        } // windowClosing
    } // WindowAdapter
);
frame.show();
} // main

} // frame_c
```

18

JAVADOC på frame_c, sqrt_c

```
Kommandotolken
C:\net\java\ht00\Folex\ordgissapp>javadoc -private *.java
Loading source file frame_c.java...
Loading source file ordgiss_c.java...
Constructing javadoc information...
Building index for all the packages and classes...
Generating overview-tree.html...
Generating index-all.html...
Generating deprecated-list.html...
Building index for all classes...
Generating allclasses-frame.html...
Generating index.html...
Generating packages.html...
Generating frame_c.html...
Generating ordgiss_c.html...
Generating ordgiss_c_ordgiss_listener.html...
Generating ordgiss_c_sprak_cbx_listener.html...
Generating serialized-form.html...
Generating package-list...
Generating help-doc.html...
Generating stylesheet.css...
C:\net\java\ht00\Folex\ordgissapp>
```

Filer

- Uj vad mycket filer!
- tråligt och besvärligt
 - nej, inte alls
 - dubbelklicka på index.html
 - startpunkt för webbsajter

Körexempel - HTML-dok

The screenshot shows a Microsoft Internet Explorer window titled "Generated Documentation (Untitled) - Microsoft Internet Explorer". The address bar shows the path to the generated HTML documentation. The main content area displays the "Class frame_c" documentation, including its inheritance from java.lang.Object and a brief description in Swedish: "Klassen innehåller en mainfunktion som kan fungera som en mall för hur man bygger Windowsapplikationer i Java." Below the class description is a "Constructor Summary" section.

Special - Applet i applikation

- Applet ärver från Panel
 - en Applet är en Panel
- lägg en Applet i en Frame
 - anropa `init` i Appletklassen
 - ingen annan gör ju det . . .
- kör `show` på Framen
- går bara med en ”snäll” applet ...

22

Utnyttja Appletmetoder

- ***init***-metoden för engångsarbete
 - t.ex. hämta bilder
- ***start***-metoden
 - starta trådar
 - starta allt om det ändå ska nyköras
- ***stop***-metoden
 - stoppa startade trådar och stäng AV ljud !
- ***paint***-metoden
 - för omritning av applet

Applet-parametrar

- Anges i HTML-koden likt:

```
<APPLET CODE=TheApplet.class WIDTH=100 HEIGHT=100>  
<PARAM NAME=ParamTitel VALUE="En titel">  
<PARAM NAME=Antal VALUE=5>  
</APPLET>
```

- Läses i appleten med metoden
getParameter:

Runnable-Applet

- visar hur länge websidebesök pågått
- parameterstyrd
 - m.a.p. det som ska presenteras
- implementerar Runnable
- run-metoden
 - beräknar tid, ställer om Label-text
- start/stop
 - startar/stoppar tidtagning

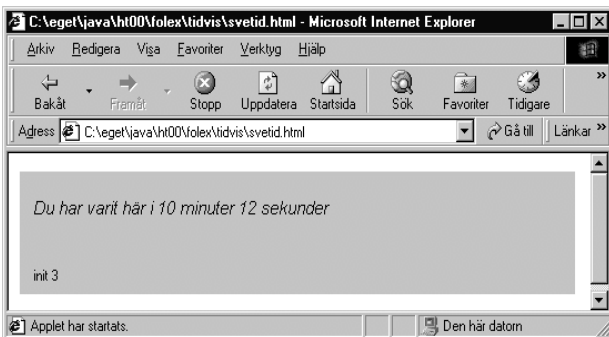
25

HTML-fil Svenska

```
<HTML>
<APPLET CODE = tid_c.class WIDTH = 500 HEIGHT = 100>
<PARAM NAME=base_msg VALUE="Du har varit här i">
<PARAM NAME=one_minute VALUE="minut">
<PARAM NAME=sev_minutes VALUE="minuter">
<PARAM NAME=one_hour VALUE="timme">
<PARAM NAME=sev_hours VALUE="timmar">
<PARAM NAME=one_second VALUE="sekund">
<PARAM NAME=sev_seconds VALUE="sekunder">
</APPLET>
</HTML>
```

26

Körexempel - svenska



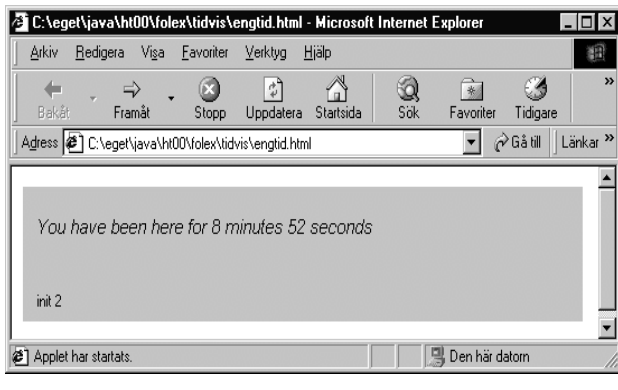
27

HTML-fil Engelska

```
<HTML>
<APPLET CODE = tid_c.class WIDTH = 500 HEIGHT = 100>
<PARAM NAME=base_msg VALUE="You have been here for">
<PARAM NAME=one_minute VALUE="minute">
<PARAM NAME=sev_minutes VALUE="minutes">
<PARAM NAME=one_hour VALUE="hour">
<PARAM NAME=sev_hours VALUE="hours">
<PARAM NAME=one_second VALUE="second">
<PARAM NAME=sev_seconds VALUE="seconds">
</APPLET>
</HTML>
```

28

Körexempel - engelska



Hämta bilder i Applets

- laddas ned från en webserver
 - *via getImage* med önskad URL

```
Image image = getImage(getCodeBase(), "imgDir/a.gif");
```

- *getCodeBase*
 - ger bas-URL från vilken Applet hämtades

Spela ljud i en applet

- metoden ***play(URL)***
 - för ljud i ".au" format
- hämta ljud separat
 - ***getAudioClip(URL)***
 - ger objekt som implementerar interfacet **AudioClip**
 - play - spela ljudet en gång
 - loop - spela ljudet kontinuerligt
 - stop - avbryt pågående spelning

Kommunicera med browsern

- ***showStatus***
 - ger information till användaren på raden längst ned i browsern

```
showStatus("MyApplet: Loading image file " + file);
```
- ***showDocument***
 - visa annan web-sida i browsern

Kommunicera med browsern

- **showDocument(URL)** - visa angiven web-sida
- **showDocument(FrameOrWindow)**
 - visa angiven web-sida i utpekad HTML-frame eller fönster
 - FrameOrWindow kan vara:
 - Ett specifikt namn eller
 - `_blank` - nytt tomt fönster
 - `_self` - aktuellt fönster och frame
 - `_parent` - övergripande frame
 - `_top` - toppnivå frame

Kommunicera med browsern

- kommunicera med andra applet:ar på samma web-sida
 - **AppletContext**-metoden *getApplet*
 - *getApplets* ger
 - ett **Enumeration**-objekt som innehåller alla applet:ar på sidan

Kommunicera med browsern

- *getApplet*
 - annan applet namnges i HTML-fil

```
<APPLET CODEBASE=example/ CODE=Sender.class WIDTH=450  
HEIGHT=200 NAME="buddy">
```

```
...  
</APPLET>
```

```
<APPLET CODEBASE=example/ CODE=Receiver.class  
WIDTH=450 HEIGHT=35>  
<PARAM NAME="name" value="old pal">
```

```
...  
</APPLET>
```

Grafiska användargränssnitt

- för enklare interaktion med användaren
- komponentbaserade
 - ett fåtal standardelement, t.ex.
 - knappar, listboxar, editfält m.m.
- jämför teckenbaserade gränssnitt
 - t.ex. DOS-program

Grafiska MMI:er i Java

- AWT
 - Abstract Window Toolkit
 - originalet
- JFC
 - Java Foundation Classes
 - Swingkomponenter
 - den ”nya” given

37

AWT

- Heavyweight components
- minsta gemensamma nämnare
 - för olika OS
 - innehåller (för) lite
- utnyttjar standardelement ur OS
- Peerklasser mot OS-element
 - Win32ButtonPeer
 - MacButtonPeer
 - MotifButtonPeer

38

Swing

- Lightweight Components
- beror ej av underliggande OS
- likadant oavsett OS ELLER
 - SetLookAndFeel
- rekommenderas av Javasoft
- fler komponenter än AWT

39

Komponenter i AWT

- Button, Checkbox, Choice, List
- MenuItem, TextField, Label
- TextArea, Scrollbar, ScrollPane
- Canvas
- Container
- Frame, Dialog, FileDialog, Panel

40

AWT - Checkbox

- Checkbox 1
- Checkbox 2
- Checkbox 3

```
cb1 = new Checkbox(); // Standardläget är av  
cb1.setLabel("Checkbox 1");  
cb2 = new Checkbox("Checkbox 2");  
cb3 = new Checkbox("Checkbox 3");  
cb3.setState(true); // Markera denna checkbox
```

41

AWT - CheckBoxGroup

- Checkbox 4
- Checkbox 5
- Checkbox 6

```
cbg = new CheckboxGroup();  
cb4 = new Checkbox("Checkbox 4", cbg, false);  
cb5 = new Checkbox("Checkbox 5", cbg, false);  
cb6 = new Checkbox("Checkbox 6", cbg, false);  
...
```

42

AWT - Scrollbar



```
slider = new Scrollbar(  
    Scrollbar.HORIZONTAL, 0, 100, min, max);  
. . .
```

- I JDK 1.1 och högre finns ScrollPane

43

AWT - ScrollPane



```
ScrollPane sp1 = new ScrollPane(  
    ScrollPane.SCROLLBARS_ALWAYS);  
sp1.add(aComponent);
```

44

AWT - gruppera komponenter

- basklass för gruppering
 - Container
- flera Components läggs i en Container
- Button i en Panel
- Panel i en Frame
- Panel i en Panel

45

AWT - komponentplacering

- styrs av LayoutManager
- inbördes ordning mellan komponenter
 - styr fysisk placering
 - undantag: setLayout(null)
- setLayout
 - för att välja LayoutManager

46

AWT - FlowLayout

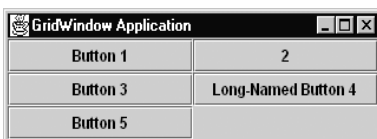
- default layout för Panel
- från vänster till höger radvis
- radbrytning vid behov



47

AWT - GridLayout

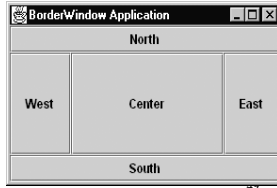
- ger samma storlek på komponenterna
- placeras i ett rutnät (tabell)
- kolumner och rader
 - argument till konstruktorn



48

AWT - BorderLayout

- högst fem komponenter
 - kan dock vara Panel
 - Norr, Syd, Öst, Väst, Mitten.
 - `add("North", comp)`



AWT - CardLayout

- hanterar ett antal kort
 - kan vara Panel
- `container.add("kortnamn", comp)`
- `cardlayout.show(container, "kortnamn")`



50

AWT - GridBagLayout

- baserar sig på ett rutnät
 - komplex
- komponent kan använda mer än en ruta
- komponent behöver inte fylla en ruta
- komponent kan skala olika



Händelsestyrd programmering

- JDK 1.0.2
 - en stor händelseprocedur
 - jämför Windows-programmering i C
- JDK 1.1
 - en lyssnare per objekt
 - kan hantera flera händelser
 - lyssnar bara på intressanta händelser

52

JDK 1.1 - händelser

- lyssnare implementerar rätt interface
 - alla metoder måste implementeras
- det finns Adapters
 - klasser som helt implementerar interface
 - tomma metoder
 - kan ärvas
 - bara det intressanta behöver implementeras

53

Swing - innehåll - 1

- JButton, JCheckBox, JRadioButton, JMenuBar, JMenu, JMenuItem,
- JCheckBoxMenuItem, JRadioButtonMenuItem, JSeparator, JComboBox, JList
- JSlider, JFrame, JPanel, JApplet, JTextComponent, JPasswordField, JLabel, JDialog

54

Swing - innehåll - 2

- JTabbedPane, JRootPane, JLayeredPane, JSplitPane
 - helt nya Containers
- JToolTip, JTree, JToolBar, JTable, JProgressBar
 - helt nya komponenter
- JColorChooser, JFileChooser
 - helt nya standarddialogboxar

55

Swing - paketnamn

- Swing 1.0.x
 - com.sun.java.swing
- Swing 1.1.x och senare
 - javax.swing
- varför javax ?
 - Swing en utökning
 - ingår inte i Javas standardlib

56

Swing - komponenthierarki

- basklass JComponent
 - ärver från java.awt.Container
 - kan innehålla Components
 - kan innehålla JComponents
- JFrame, JDialog, JApplet
 - ärver från motsvarande AWT-klasser

57

Swing - addera komponenter

- kan inte addera direkt
- gå via getContentPane()
- i en JApplet-subklass
 - super.getContentPane().add(...)
- i en JFrame frame
 - frame.getContentPane().add(...)

58

Swing - JLabel

- motsvarar Label i AWT
- kan dessutom visa bilder

```
label1 = new JLabel(
    "Image and Text", icon, JLabel.CENTER);
label1.setVerticalTextPosition(JLabel.BOTTOM);
label1.setHorizontalTextPosition(JLabel.CENTER);
label2 = new JLabel("Text-Only Label");
label3 = new JLabel(icon);
super.getContentPane().add(label1);
super.getContentPane().add(label2);
super.getContentPane().add(label3);
```



Swing - JButton

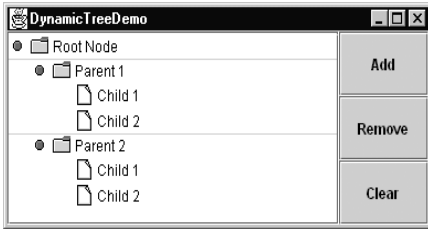
- JButton motsvarar Button i AWT
- kan dessutom ha ikon

```
b2 = new JButton(
    "Middle button", middleButtonIcon);
b2.setVerticalTextPosition(AbstractButton.BOTTOM);
b2.setHorizontalTextPosition(AbstractButton.CENTER);
b2.setMnemonic('m');
```



Swing - JTree

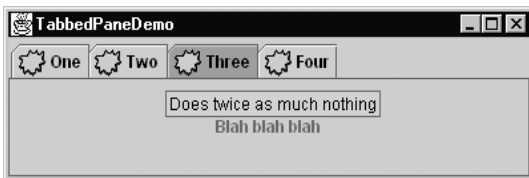
- helt ny komponent i Swing



61

Swing- JTabbedPane

- ny klass i Swing
- stöder fönster med flikar



62

Swing - hantering

- komponentplacering via LayoutManagers
 - tillägg BorderLayout
- händelsehantering via lyssnare
 - mer att lyssna efter än awt-klasserna
- med andra ord
 - samma hantering som för awt-klasser

63

Swingexempel - JLF



64

Swingexempel - WLF



65

frame_c - 1

```
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JPanel;  
import javax.swing.JButton;  
import javax.swing.BorderFactory;  
import javax.swing.UIManager;  
import java.awt.event.WindowAdapter;  
import java.awt.event.WindowEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.ActionEvent;  
import java.awt.GridLayout;
```

66

frame_c - 2

```
public class frame_c extends JFrame
{
    public frame_c( String titel )
    {
        super( titel );

        this.num_lbl = new JLabel(
            frame_c.num_prefix + "0  " );

        JButton count_btn = new JButton(
            "Jag är Swingknapp!" );
        count_btn.setMnemonic( 'J' );
        count_btn.addActionListener(
            new count_btn_listener() ); // ActionListener obj
    }
}
```

67

frame_c - 3

```
JPanel panel = new JPanel();
panel.setBorder(
    BorderFactory.createEmptyBorder(
        30,    // top
        30,    // left
        10,    // bottom
        30 ) ); // right

panel.setLayout( new GridLayout(
    0,    // rows, 0 = hur många som helst
    1)); // cols
panel.add( count_btn);
panel.add( this.num_lbl);
```

68

frame_c - 4

```
super.getContentPane().add( panel );
super.pack();

super.addWindowListener(
    new window_exit_listener() ); //WindowListener obj
} // frame_c
```

69

frame_c - 5

```
private class count_btn_listener
    implements ActionListener
{
    public void actionPerformed( ActionEvent aev )
    {
        handle_count_btn();
    }
} // count_btn_listener
```

70

frame_c - 6

```
private class window_exit_listener
    extends WindowAdapter
{
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
} // window_exit_listener
```

71

frame_c - 7

```
private void handle_count_btn()
{
    this.num_clicks++;
    this.num_lbl.setText(
        this.num_prefix + this.num_clicks );
} // handle_count_btn

private static String num_prefix =
    "Antal klickningar: ";
private JLabel num_lbl;
private int num_clicks = 0;
} // frame_c
```

72

frame_c - testmain - 1

```
public static void main(String[] args)
{
    try
    {
        if ( args[0].equalsIgnoreCase( "W" ) )
        {
            UIManager.setLookAndFeel(
                UIManager.getSystemLookAndFeelClassName());
        }
    }
}
```

73

frame_c - testmain - 2

```
else
{
    UIManager.setLookAndFeel(
        UIManager.getCrossPlatformLookAndFeelClassName());
} // if
} // try
catch (Exception e)
{
} // catch
```

74

frame_c - testmain - 3

```
frame_c frame = new frame_c(
    "SwingApplikation");
frame.setVisible(true);
} // main
```

- jämför tidigare exempel
 - Frame byggs upp i main
- ovanstående ex
 - JFrame byggs upp i konstruktor
- effekten blir densamma

75

Grafikhantering

- public void paint(Graphics gr)
- utnyttja gr för ritning
 - linjer, rektanglar, ovaler
 - bågar, polygoner
 - bilder (JPEG och GIF)
 - GIF 256 färger, enkla
 - JPEG för foton o.d.
- Graphics2D
 - mer avancerad hantering

76

Graphics - 1

- drawLine, drawPolyline, drawPolygon
 - linjeritning
- drawRect, drawRoundRect, draw3DRect
 - fyrkanter
- drawArc, drawOval
- dessutom för slutna figurer
 - motsvarande fill-rutiner
 - t.ex. fillOval

77

Graphics - 2

- setFont
- drawBytes, drawChars, drawString
 - för textutmatning
- drawImage
 - för bildvisning

78

Image - laddning

- JPEG eller GIF
- i Applet subclass
 - Image image = getImage(URL-objekt)
- i en applikation
 - Image image =
Toolkit.getDefaultToolkit().getImage(
fil eller URL-namn)

79

Vänta på bild

- OBS! getImage returnerar direkt
 - laddning sker i bakgrundstråd
- utnyttja MediaTracker
 - tracker.addImage(image, 27)
 - tracker.waitForID(27)
 - inväntar laddning av alla bilder med ID = 27

80

Canvas - rityta

- subclass till Component
 - inte till Container
- om man vill ha yta att rita på
- överrid Paint
- basklass för egna komponenter
 - JavaBeans

81

Omritning av Panel/Canvas

- i subklass till Panel eller Canvas
- anropa repaint()
 - begär omritning av area
- varianter
 - hela arean
 - del av area
 - inom viss tid

82

Omritning - update/flimner

- repaint ger normalt anrop av
 - public void update(Graphics gr)
- default update
 - ritas om hela bakgrunden
 - anropar paint
 - ritas om allt igen
 - ORSAK TILL FLIMMER

83

Undvika flimner

- överrid update
 - d.v.s skriv en egen
- rita bara om den bakgrund som berörs
- anropa sedan paint
- OBS! paint ska alltid implementeras
 - Java kan förbigå update
 - t.ex. då man drar i ett fönster

84

Dubbelbuffring

- rita bilder i ”osynligt” minne
- vid omritning
 - visa färdigritad bild
 - drawImage
- långsammare alternativ
 - rita om bilden varje gång
