

OOP Objekt-orienterad programmering

Föreläsning 16

Grafiska användargränssnitt, GUI

Ytor

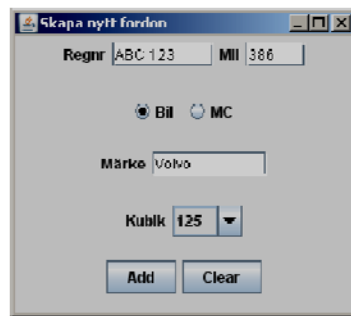
Komponenter

Layout-managers

Lyssnare

Marie Olsson

GUI - Graphical User Interface



Man skapar en yta (ett fönster) och på den ytan adderar man ett antal komponenter. Olika ytor:

- JFrame - "vanligt" fönster
- JApplet - fönster i en webbrowser

Olika komponenter: JLabel, JTextField, JRadioButton, JComboBox, JButton

Till komponenter kopplas Lyssnare som anger vad som skall hända när man t ex trycker på en knapp.

Marie Olsson

AWT kontra Swing

OOP F16:3

AWT (Abstract Windowing Toolkit) Javas bibliotek för GUI

Swing (från version 1.2) fler komponenter, utbyggt funktionalitet

Denna föreläsning behandlar Swing-klassernas enklare delar (liknar AWT) samt de delar av AWT som fortfarande "behövs"

i AWT
Button, TextField etc

i Swing
JButton, JTextField etc

Marie Olsson

Olika delar som behövs

OOP F16:4

Ytor

(olika sorters fönster)

AWT: Frame, Applet

Swing: JFrame, JApplet

Komponenter

(placeras i ytorna)

AWT: Label, TextField, Button, Checkbox, TextArea, Panel m.fl.

Swing: JLabel, JTextField, JButton, JRadioButton, JTextArea, JPanel m.fl. (motsvarande AWT's + fler)

Layout-managers

(hur komponenterna ligger i ytorna)

AWT: BorderLayout, FlowLayout, GridLayout m.fl.

Swing: använder AWT's + några till

Lyssnare

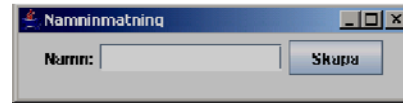
(kod för vad som skall hända när man t ex trycker på en knapp etc.)

AWT: ActionListener, KeyListener, MouseListener m.fl.

Swing: använder AWT's + några till

Marie Olsson

Hur gör man för att skapa ett fönster?



```
import javax.swing.*;
import java.awt.*;
class Namninmatning extends JFrame{
    Namninmatning(){
        super("Namninmatning");
        setLayout(new FlowLayout());
        JLabel l=new JLabel("Namn:");
        add(l);
        JTextField tf=new JTextField(12);
        add(tf);
        JButton knapp=new JButton("Skapa");
        add(knapp);
        setSize(300,75);
        setVisible(true);
    }
    public static void main(String[]args){
        new Namninmatning();
    }
}
```

Marie Olsson

JLabel

En textsträng, kan ej editeras av användaren.



```
import javax.swing.*;
class LabelTest extends JFrame{
    LabelTest(){
        super("Exempel Label");
        JLabel l=new JLabel("Hej Hopp!");
        add(l);
        setSize(300,75);
        setVisible(true);
    }
    public static void main(String[]args){
        new LabelTest();
    }
}
```

Några metoder hos JLabel:

```
l.setText("Ny text"); //Sätta ny text på JLabel'n
String s = l.getText(); //Returnerar texten just nu
```

Marie Olsson

JButton

OOP F16:7

En knapp man kan trycka på. För att nåt skall hända måste lyssnare kopplas på.

```
import javax.swing.*;

class KnappTest extends JFrame{
    KnappTest(){
        super("Exempel Knapp");
        JButton knapp=new JButton("Knapp");
        add(knapp);
        setSize(200,75);
        setVisible(true);
    }
    public static void main(String[]args){
        new KnappTest();
    }
}
```



Några metoder hos JButton:

```
knapp.setText("Annat");    //Ändra knapptexten
String s = knapp.getText(); //Hämta aktuell knapptext
knapp.setEnabled(false);   //Göra knappen "otryckbar"
knapp.setEnabled(true);
```

Marie Olsson

OOP F16:8

Vilka olika komponenter finns och vilka metoder & attribut har de?
Se java-dokumentationen:

<http://java.sun.com/javase/6/docs/api/>

javax.swing
Class JButton

java.lang.Object

```
|
+--java.awt.Component
    |
    +--java.awt.Container
        |
        +--javax.swing.JComponent
            |
            +--javax.swing.AbstractButton
                |
                +--javax.swing.JButton
```

Marie Olsson

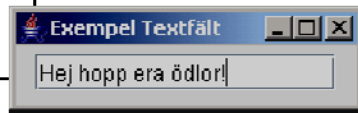
JTextField

OOP F16:9

Ett textinmatningsfält där en textsträng kan skrivas in. Scrollar automatiskt.

```
import javax.swing.*;
import java.awt.*;

class TextTest extends JFrame{
    TextTest(){
        super("Exempel Textfält");
        JTextField tf=new JTextField(15);
        add(tf);
        setSize(200,60);
        setVisible(true);
    }
    public static void main(String[]args){
        new TextTest();
    }
}
```



Några metoder hos JTextField:

```
String s = tf.getText(); //Avläser texten
tf.setEditable(false);   //Tillåter inte inmatning
```

Kan även skapas med text från början:

```
JTextField tf = new JTextField("Starttext", 15);
```

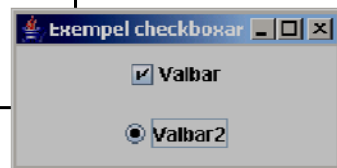
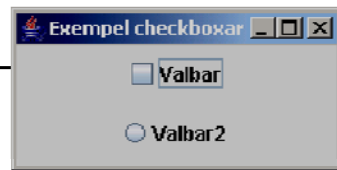
Marie Olsson

JCheckBox och JRadioButton

OOP F16:10

En checkruta som kan vara på eller av.

```
RadioTest(){
    super("Exempel checkboxar");
    setLayout(new GridLayout(2,1));
    JCheckBox cb=new JCheckBox("Valbar");
    JPanel p1=new JPanel();
    p1.add(cb);
    add(p1);
    JRadioButton rb=new JRadioButton("Valbar2");
    JPanel p2=new JPanel();
    p2.add(rb);
    add(p2);
    setSize(200,100);
    setVisible(true);
}
```



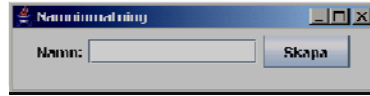
Avläses med metoden isSelected() som returnerar en boolean

```
if (cb.isSelected()) ... ;
```

Marie Olsson

Vad händer om man trycker på krysset uppe till höger?

Default-beteendet är att fönstret "göms" men applikationen fortfarande "snurrar".



Kan ändras med metoden `setDefaultCloseOperation`:

```
setDefaultCloseOperation(EXIT_ON_CLOSE);
```

Marie Olsson

LayoutManager

Om man adderar flera komponenter till en JFrame - hur hamnar dessa då?

Det bestäms av vilken LayoutManager som JFrame'n har. Alla ytor där komponenter kan adderas har en LayoutManager. Man kan byta LayoutManager om man vill ha en annan än den som är default.

Några av de vanligaste:

BorderLayout

Fem delytor Nord, Syd, Väst, Öst, Center. Default hos JFrame.

FlowLayout

Komponenterna efter varandra. Radbyte.

GridLayout

Ett rutmönster med ett antal rader och kolumner.

BoxLayout

Komponenterna i rad horisontellt eller vertikalt. Ej "radbyte"

LayoutManager ändras med metoden `setLayout`

```
JFrame fr = new JFrame();  
fr.setLayout(new FlowLayout());
```

Marie Olsson

10 knappar med FlowLayout

```
TenKnapps(){
    super("10 Knappar");
    setLayout(new FlowLayout());
    for (int x=1; x<=10; x++)
        add(new JButton("Knapp "+x));
    setSize(250,150);
    setVisible(true);
}
```

Knapparna hamnar olika
beroende på fönsterstorlek



Marie Olsson

10 knappar med GridLayout

```
TenKnapps(){
    super("10 Knappar");
    setLayout(new GridLayout(4,3));
    for (int x=1; x<=10; x++)
        add(new JButton("Knapp "+x));
    setSize(250,150);
    setVisible(true);
}
```



GridLayout(4,3) ger 4 rader med 3 kolumner.

Marie Olsson

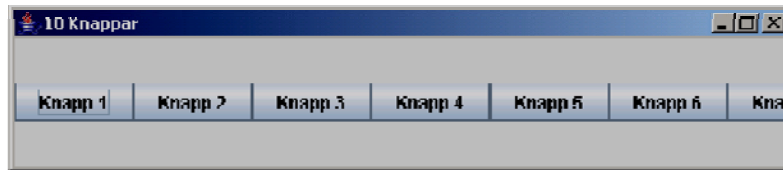
10 knappar med BorderLayout

OOP F16:15

```
TenKnapps(){
    super("10 Knappar");
    JPanel pan=new JPanel();
    pan.setLayout(new BorderLayout(pan, BorderLayout.Y_AXIS));
    for (int x=1; x<=10; x++)
        pan.add(new JButton("Knapp "+x));
    add(pan);
    setSize(200, 200);
    setVisible(true);
}
```



med: `pan.setLayout(new BorderLayout(pan, BorderLayout.X_AXIS));`

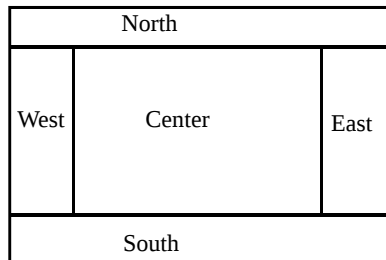


Marie Olsson

BorderLayout

OOP F16:16

Delar upp ett fönster i fem logiska delytor



De olika delarna är String-konstanter i BorderLayout:

NORTH
WEST
CENTER
EAST
SOUTH

Man adderar en komponent per delyta. Vill man ha flera på en delyta så läggs dessa i en JPanel.

När man adderar väljer man vart man vill addera:

`add(komponent, BorderLayout.SOUTH);`

De delytor som ej har något adderat i sig "krymps" så att andra delytor expanderar. Center expanderas i första hand.

Marie Olsson

Exempel BorderLayout

OOP F16:17

```
BorderTest(){
    super("Test av BorderLayout");
    JPanel upp=new JPanel();
    upp.add(new JLabel("Namn:"));
    upp.add(new JTextField(15));
    add(upp, BorderLayout.NORTH);

    JPanel ner=new JPanel();
    ner.add(new JButton("Skapa"));
    ner.add(new JButton("Ta bort"));
    ner.add(new JButton("Sluta"));
    add(ner, BorderLayout.SOUTH);

    setSize(300, 250);
    setVisible(true);
}
```



Man "jobbar" hela tiden genom att skapa små delpaneler (JPanel's) och i dessa lägga komponenter. Panelerna adderas sedan på lämpligt ställe i JFrame'n. Ibland (ofta...) skapar man delpaneler som läggs i paneler (som läggs i paneler ...) och sedan adderas till fönstret (JFrame'n).

FlowLayout är default hos JFrame.

Marie Olsson

Händelser

OOP F16:18

När användaren "gör något" i ett GUI-fönster skapas en händelse:

- trycker på en knapp
- skriver i ett textfält
- fyller i en checkbox
- rör musen över ett fält
- etc.

Denna händelse kan "fångas" så att man ser till att en viss kod exekveras. Man kopplar en lyssnare till en komponent. När något händer med den komponenten aktiveras lyssnaren och en metod i lyssnaren anropas.

En lyssnare är en klass som implementerar ett gränssnitt. Olika gränssnitt för olika händelser. Enklaste gränssnittet är ActionListener.

```
interface ActionListener{
    void actionPerformed(ActionEvent e);
}
```

Lyssnar-interfacen ligger på java.awt.event
(de flesta, några nya finns i javax.swing.event)

Marie Olsson

Varje fönster har egen "livsloop"

OOP F16:19

När ett fönster ritas upp - setVisible(true) anropas - startar en egen exekveringstråd. Denna känner av användarens aktiviteter i fönstret. När olika saker görs anropas lyssnare. Exekveringen fortsätter tills fönstret stängs.

En klass som avslutar en applikation, kopplas t.ex. till en knapp "Quit".

```
class QuitLyssnare implements ActionListener{
    public void actionPerformed(ActionEvent e){
        System.exit(0);
    }
}
```

Kopplas ihop med en komponent genom att adderas till komponenten

```
JButton quitknapp = new JButton("Quit");
quitknapp.addActionListener(new QuitLyssnare());
```

Knappen adderas på lämpligt ställe i fönstret. När man klickar på knappen kommer metoden actionPerformed i QuitLyssnare att anropas och i detta fall programmet att avslutas.

Marie Olsson

Placering av lyssnarklasser

OOP F16:20

Kan placeras i egen fil. Dock behöver lyssnarna ofta komma åt olika komponenter för att kunna avläsa dessa. Smidigast att placera som **inre klasser** i JFrame-subklassen.

```
class ExempelGUI extends JFrame{
    JButton skapaknapp, quitknapp;
    JTextField inmatning;
    ExempelGUI(){
        //Konstruktor som skapar komponenterna, placerar dem i fönstret
        //kopplar ihop med lyssnare och startar fönsterexekveringstråden
    }
    class SkapaLyssnare implements ActionListener{
        public void actionPerformed(ActionEvent e){
            //Det som skall göras. Har åtkomst till de globala komponenterna
        }
    }
    public static void main(String[] args){
        new ExempelGUI();
    }
}
```

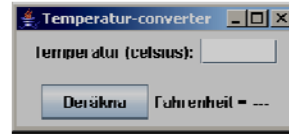
Marie Olsson

```

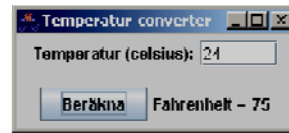
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Celsius extends JFrame{
    JTextField inmat; JLabel res;
    public Celsius(){
        super("Temperatur-converter");
        setLayout(new GridLayout(2,1));
        JPanel upp=new JPanel();
        upp.add(new JLabel("Temperatur (celsius):"));
        inmat=new JTextField(5);
        upp.add(inmat);
        add(upp);
        JPanel ner=new JPanel();
        JButton knapp=new JButton("Beräkna");
        knapp.addActionListener(new Lyss());
        ner.add(knapp);
        ner.add(new JLabel("Fahrenheit ="));
        res=new JLabel("---");
        ner.add(res);
        add(ner);
        setSize(220, 100);
        setLocation(200,300);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setVisible(true);
    }
    class Lyss implements ActionListener{
        public void actionPerformed(ActionEvent e){
            String str = inmat.getText();
            int cels = Integer.parseInt(str);
            int fahr = (int)(9.0/5.0*cels+32);
            res.setText(String.valueOf(fahr));
        }
    }
    public static void main(String[] args){
        new Celsius();
    }
}

```

Vi vill skapa ett litet GUI
där man kan kolla göra om
celsius till fahrenheit



Till vänster finns koden
som skapar ett sådant
fönster. Med lyssnare.



Marie Olsson