

Swing 2.0

Kika på FemtonSpel.java

Exempel CirkelDemo.java

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

public class CircleDemo extends JFrame {
    public CircleDemo() {
        CircleDiagram d1 = new CircleDiagram();
        CircleDiagram d2 = new CircleDiagram(-100, 100);
        CircleDiagram d3 = new CircleDiagram(0, 100000);
        d2.setBorder(new EtchedBorder());
        d3.setBorder(new LineBorder(Color.lightGray, 5));
        d1.setValue(45);
        d2.setValue(50);
        d3.setValue(80000);
        add(d1); add(d2); add(d3);
        setLayout(new GridLayout(1, 3));
        setSize(450,155);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] arg) {
        CircleDemo demo = new CircleDemo();
    }
}
```

Konstruktion av egna grafiska komponenter

Tre tekniker:

1. Skapa en subklass till JFrame eller JPanel sätt på den samman flera standardkomponenter till en enhet, t.ex. ett formulär.
2. Utgå från en standardklass och skapa en subklassen med ytterligare egenskaper, t.ex. speciell sorts knapp.
3. Utgå från en standardklass (ofta JPanel) och rita komponenten själv, genom att omdefiniera metoden paintComponent.

```
// Teknik 1
class MinKlass extends JPanel implements ActionListener{
    private JLabel l = new JLabel("En text"; JLabel.CENTER);
    private JButton b = new JButton("En knapp");
    ...
    public MinKlass() { // Konstruktor
        add(l); add(b); // placera ut delkomponenter
        ...
        l.setOpaque(true); // sätt egenskaper för delkomponenter
        ...
        setBackground(Color.white); // sätt egenskaper för hela komponenten
        ...
    }
    // definiera instansmetoder för hela komponenten
    public void actionPerformed(ActionEvent e) {
        ...
    }
    ...
}
```

```
// Teknik 2
class MinKnapp extends JButton {
    private final int rad, kol; // extra egenskaper

    public MinKnapp (int r, int k, String txt) {
        setText(txt);
        rad = r;
        kol = k;
    }

    public int radNr() {
        return rad;
    }

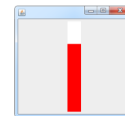
    public int kolNr() {
        return kol;
    }
}
```

```
// Teknik 3

class Stapel extends JPanel {
    private double procent;

    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(getForeground());
        int höjd = (int) (getHeight() * procent /100);
        g.fillRect(0, getHeight()-höjd, getWidth(), höjd);
    }

    public void sättProcent(double p){
        if (p >= 0 && p <= 100)
            procent = p;
        else
            throw new IllegalArgumentException("Felaktigt argument");
        repaint();
    }
}
```



```
Stapel st = new Stapel();
setLayout(new FlowLayout());
add(st);
st.sättProcent(75);
st.setPreferredSize(new Dimension(30, 200));
st.setBackground(Color.white);
st.setForeground(Color.red);
```