

Sista handen vid grafiska grejer

Att bara skriva `add` är samma sak som att skriva `this.add` vilket gör att du lägger sakerna på objektet du håller på med,

Nyhet

BorderLayout har fem default-ställen att lägga saker. Väderstrecken + Center.

Saker som ligger i norr och söder kommer få den höjden de kräver men bredden begränsas av fönstret.

Öst och väst kommer få den bredden de kräver men få höjd efter fönstret.

Center får den plats som blir över.

BoxLayout lägger komponenter på en given axel och tar hänsyn till knappars storlek.

Mellanrum mellan komponenter görs med hjälp av `createRigidArea` (lägger till ett tomt objekt mellan knapparna). Det finns såväl horisontal som vertikalt klistor.

Vertikalt klistor

Att knapparna ligger lite fult fixas med `alignment`. (`setAlignmentX("LEFT_ALIGNMENT")`);

```
private JButton b1 = new JButton("1");
private JButton b2 = new JButton("Knapp 2");
private JLabel l3 = new JLabel("Label 3");
private JTextField t4 = new JTextField("Text 4");
private JPanel p5 = new JPanel();
```

```
p5.setBackground(Color.white);
p5.setPreferredSize(new Dimension(75, 75));
setLayout(new FlowLayout());
add(b1); add(b2); add(l3); add(t4); add(p5);
pack();
```



```
setLayout(new GridLayout(3,2)); // 3 rader, 2 kolumner
add(b1); add(b2); add(l3); add(t4); add(p5);
```



```
setLayout(new BorderLayout());
add(b1, BorderLayout.WEST); add(b2, BorderLayout.NORTH);
add(l3, BorderLayout.EAST); add(t4, BorderLayout.SOUTH);
add(p5, BorderLayout.CENTER);
```



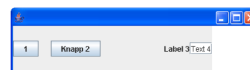
```
setLayout(new BoxLayout(getContentPane(), BoxLayout.X_AXIS));
add(b1); add(b2); add(l3); add(t4); add(p5);
```



```
t4.setMinimumSize(t4.getPreferredSize());
t4.setMaximumSize(t4.getPreferredSize());
p5.setMinimumSize(p5.getPreferredSize());
p5.setMaximumSize(p5.getPreferredSize());
```



```
add(b1);
add(Box.createRigidArea(new Dimension(20, 1)));
add(b2);
add(Box.createHorizontalGlue());
add(l3); add(t4); add(p5);
```

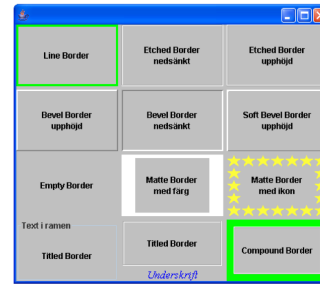


```
setLayout(new BoxLayout(getContentPane(),
    BoxLayout.Y_AXIS));
add(b1);
add(Box.createRigidArea(new Dimension(1, 20)));
add(b2);
add(Box.createVerticalGlue());
add(l3); add(t4); add(p5);
```



setBorder

Använd compoundborder för att göra knapparna likadana och sen göra den knapp man trycker på till en viss färg!!



Exempel Abonnemang.java

```
public class Abonnemang {
    String namn;
    double perMån;
    double perMinDag;
    double perMinÖvr;
    double friaDag;
    double friaÖvr;
}

import javax.swing.*;
import java.util.*;
import java.io.*;

public class SokAbonnemang {
    public static void main(String[] arg) throws IOException{

        // skapa fältet med abonnemang
        Abonnemang[] a = new Abonnemang[100]; // Steg 1 och 2

        // Koppla en scanner till filen och läs filen
        Scanner fil = new Scanner(new File("mobildata.txt"));
        int n = 0; // antal olika abonnemang

        while (fil.hasNext()) { // läs en rad i filen
            a[n] = new Abonnemang(); // Steg 3: skapa ett objekt
            a[n].namn = fil.next();
            a[n].perMån = fil.nextDouble();
            a[n].perMinDag = fil.nextDouble();
            a[n].perMinÖvr = fil.nextDouble();
            a[n].friaDag = fil.nextDouble();
            a[n].friaÖvr = fil.nextDouble();
            n++;
        }

        // Sök abonnemang
        while(true) {
            String sökt = JOptionPane.showInputDialog("Vilket abonnemang söker du?");
            if (sökt == null)
                break;

            // Leta efter abonnemanget i fältet a
            int i = 0;
            for (; i<n; i++)
                if (a[i].namn.equals(sökt))
                    break;
            if (i < n) // abonnemanget hittades
                JOptionPane.showMessageDialog(null,
                    String.format("Fast månadskostnad: %.2f \nPris per minut: %.2f",
                        a[i].perMån, a[i].perMinDag));
            else
                JOptionPane.showMessageDialog(null, "Kunde inte hitta " + sökt);
        }
    }
}
```

Hur gör man när man sorterar?

1. Hitta minsta och flytta första till minsta plats. (nu kan du strunta i första, det ligger ju rätt)
2. Repeat. Tillslut kommer alla ligga på plats.

Kod för att sortera och flytta till lägsta index.

Exempel SorteraAbonnemang2.java

```
import javax.swing.*;
import java.util.*;
import java.io.*;
import java.text.*;

public class SorteraAbonnemang2 {
    public static void main(String[] arg) throws IOException{
        // skapa fältet med abonnemang
        Abonnemang[] a = new Abonnemang[100]; // Steg 1 och 2
        // Koppla en scanner till filen och läs filen
        Scanner fil = new Scanner(new File("mobildata.txt"));
        int n = 0; // antal olika abonnemang
        while (fil.hasNext()) {
            // läs en rad i filen
            a[n] = new Abonnemang(); // Steg 3: skapa ett objekt
            a[n].namn = fil.next();
            a[n].perMån = fil.nextDouble();
            a[n].perMinDag = fil.nextDouble();
            a[n].perMinÖvr = fil.nextDouble();
            a[n].friaDag = fil.nextDouble();
            a[n].friaÖvr = fil.nextDouble();
            n++;
        }
        sortera(a, 0, n-1);
        PrintWriter utström = new PrintWriter(
            (new BufferedWriter(
                (new FileWriter("mobildata2.txt"))));
        for (int i=0; i<n; i++)
            // Använd metoden format för att få decimalkomman i utskriften
            utström.format("%s %1.0f %1.2f %1.2f %1.2f %1.2f\n",
                a[i].namn, a[i].perMån, a[i].perMinDag,
                a[i].perMinÖvr, a[i].friaDag, a[i].friaÖvr);
        utström.close();
    }

    public static void sortera(Abonnemang[] a,
                               int start, int slut) {
        Collator c = Collator.getInstance();
        c.setStrength(Collator.PRIMARY);

        for (int k=start; k < slut; k++) {
            // Sök den minsta bland komponenterna nr k till slut
            int m = k; // index för hittills minsta värde
            for (int i=k; i<=slut; i++)
                if (c.compare(a[i].namn, a[m].namn) < 0)
                    m = i; // nytt minsta värde funnet
            // Låt komponenterna nr k och m byta plats
            Abonnemang temp = a[k];
            a[k] = a[m];
            a[m] = temp;
        }
    }
}
```

Sortering, selection sort ("sök minsta och byt")

Före sortering	9	13	6	1	8	12
Efter varv 1	1	13	6	9	8	12
Efter varv 2	1	6	13	9	8	12
Efter varv 3	1	6	8	9	13	12
Efter varv 4	1	6	8	9	13	12
Efter varv 5	1	6	8	9	12	13

```
public static void sortera(double[] a, int start, int slut) {
    for (int k=start; k < slut; k++) {
        // Sök den minsta bland komponenterna nr k till slut
        int m = k; // index för hittills minsta värde
        for (int i=k; i<=slut; i++)
            if (a[i] < a[m])
                m = i; // nytt minsta värde funnet
        // Låt komponenterna nr k och m byta plats
        double temp = a[k];
        a[k] = a[m];
        a[m] = temp;
    }
}
```

2

En Collator har koll på länder, bokstäver osv.
PRIMARY sätter stora och små bokstäver till samma, mm.

Binärsökning

För att hitta något i en sorterad lista gör vi såhär:

1. Hoppa in vid hälften, kolla om det du söker ligger under eller över.
2. Repeat tills du hittar det du söker.

Binärkodning i praktiken, i Abonnemang-sökning.

// Filen SokAbonnemang2.java

```
import javax.swing.*;
import java.util.*;
import java.io.*;
import java.text.*;

public class SokAbonnemang2 {
    public static void main(String[] arg) throws IOException{
        // skapa fältet med abonnemang
        Abonnemang[] a = new Abonnemang[100]; // Steg 1 och 2
        // Koppla en scanner till filen och läs filen
        Scanner fil = new Scanner(new File("mobildata2.txt"));
        int n = 0; // antal olika abonnemang
        while (fil.hasNext()) {
            // läs en rad i filen
            a[n] = new Abonnemang(); // Steg 3: skapa ett objekt
            a[n].namn = fil.next();
            a[n].perMån = fil.nextDouble();
            a[n].perMinDag = fil.nextDouble();
            a[n].perMinÖvr = fil.nextDouble();
            a[n].friaDag = fil.nextDouble();
            a[n].friaÖvr = fil.nextDouble();
            n++;
        }
        // Sök abonnemang
        while(true) {
            String sökt = JOptionPane.showInputDialog
                ("Vilket abonnemang söker du?");

            if (sökt == null)
                break;
            // Leta efter abonnemanget i fältet a
            int k = binSök(a, 0, n-1, sökt);

            int i = 0;
            for (; i<n; i++)
                if (a[i].namn.equals(sökt))
                    break;
            if (k > 0) // abonnemanget hittades
                JOptionPane.showMessageDialog(null, String.format
                    ("Fast månadskostnad: %.2f " +
                     "\nPris per minut: %.2f",
                     a[k].perMån, a[k].perMinDag));
            else
                JOptionPane.showMessageDialog(null,
                    "Kunde inte hitta " + sökt);
        }
    }

    public static int binSök(Abonnemang[] a, int start,
                           int slut, String sökt) {
        Collator c = Collator.getInstance();
        c.setStrength(Collator.PRIMARY);

        int i1 = start, i2 = slut, m = 0;
        while (i1 <= i2) {
            m = (i1 + i2)/2;
            if (c.compare(sökt, a[m].namn)<0)
                i2 = m-1;
            else if (c.compare(sökt, a[m].namn)>0)
                i1 = m+1;
            else // lika
                break; // man hittade abonnemanget
        }
        if (c.compare(sökt, a[m].namn)==0)
            return m;
        else
            return start-1;
    }
}
```

Binärsökning

Exempel: Sök efter talet 60

Rad 1	1	6	8	9	12	13	20	27	31	32	43	57	60	61	75	80	92	98
Rad 2	6	8	9	12	13	20	27	31	32	43	57	60	61	75	80	92	98	
Rad 3	6	8	9	12	13	20	27	31	32	43	57	60	61	75	80	92	98	
Rad 4	6	8	9	12	13	20	27	31	32	43	57	60	61	75	80	92	98	
Rad 5	6	8	9	12	13	20	27	31	32	43	57	60	61	75	80	92	98	

5

Exempel MainDemo.java

```
public class MainDemo {
    public static void main (String[] arg) {
        for (int k=0; k<arg.length; k++)
            System.out.println("Argument " + k + ": " + arg[k]);
    }
}
```

Det händer ingenting när du kör programmet men om du ger argument vid start "java MainDemo" så kommer du få med dessa in i programmet.

Nu ska vi få veta vad (String[] arg) gör!

Arg är en referens till en array av typen String vilken vi skapar när vi kör programmet.

Flerdimensionella array's i OHArray2.pdf

Timer

```
Timer
I paketet javax.swing

Timer t = new Timer(intervall, lyssnare); // intervall i millisekunder
Ex.
Timer t = new Timer(1000, this); // en gång per sek, lyssnar själv

Lyssnarmetod:

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == t) {
        // nu har timern signalerat, gör något
    }
}

Andra metoder:
t.start();
t.stop();
t.restart();
t.setDelay(del); // ändrar intervall
t.setInitialDelay(del); // tid till första signal
t.setRepeats(bool); // true default
t.addActionListener(lyssnare);

Blanda inte ihop javax.swing.Timer med klassen java.util.TimerH
```