

Exempel VisaKlocka.java

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

public class VisaKlocka {
    public static void main (String[] arg) {
        KlockVisare v = new KlockVisare(); // skapa en KlockVisare
    }
}

class KlockVisare extends JFrame implements ActionListener {
    private Tidpunkt tp = new Tidpunkt(); // referens till Tidpunktobjektet
    private JLabel l; // Används för att skriva ut text.
    public KlockVisare() { // Konstruktor, anropas automatiskt
        javax.swing.Timer tim = new javax.swing.Timer(1000, this);
        tim.start(); // starta timern

        Calendar c = Calendar.getInstance();
        tp.sätt(c.get(Calendar.HOUR_OF_DAY),
            c.get(Calendar.MINUTE),
            c.get(Calendar.SECOND)); // Ger datum, klockslag

        l = new JLabel(tp.toString(), JLabel.CENTER);
        add(l); // Ärver via JFrame. // se kommentar på sidan 38 om äldre Javaversioner
        // Samma som this.add
        // lägger till i JLabel.

        l.setOpaque(true); // ogenomskinlig bakgrund
        l.setBackground(Color.black);
        l.setForeground(Color.white);
        l.setFont(new Font("SansSerif", Font.BOLD, 24));

        setSize(200,75); // fönstrets storlek
        setVisible(true); // gör fönstret synligt
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    public void actionPerformed(ActionEvent e) {
        // hit kommer man automatiskt en gång per sekund
        tp.tick();
        l.setText(tp.toString());
    }
}
```

Paket

```
import java.util.*;
import java.util.Scanner;
import java.lang.*; // läggs till automatiskt

import static javax.swing.JOptionPane.*;
import static java.lang.Math.*;

// static kräver ...Klass.(något i klassen);
// Stor bokstav = Klass
// Hämta alla klasser.
```

```
package minaklasser;
public class MinMath {
    // metoder mm. i klassen MinMath
}
```

```
package minaklasser;
public class EnklaDialoger {
    // metoder mm. i klassen EnklaDialoger
}
```

```
package minaklasser.filhantering;
public class TextfilInput {
    // metoder mm. i klassen TextfilInput
}
```

```
> jar cf minaklasser.jar minaklasser\*
```

```
> set CLASSPATH=.;c:\grupp\paket;c:\javapaket\minaklasser.jar
```

Synlighet

Synlighet i Java avhandlar om att alla inte ska se allt, dels för att man inte ska kunna förstöra och dels för att man inte ska behöva bry sig om spänst man egentligen inte behöver för tillfället.

Synlighet för metoder, instansvariabler och klassvariabler				
synlig i	private	<i>inget</i>	protected	public
en annan klass i samma paket	nej	ja	ja	ja
en subclass i ett annat paket	nej	nej	ja	ja
en klass i ett annat paket	nej	nej	nej	ja

Saker vilka är deklarerade i en metod kan bara användas av / i den metoden.

Detta gäller **inne i klasser INTE i metoder!**

Ett package kan vara antingen public eller " "

Private

Bara de metoderna i klassen själv kan använda. Typiskt är att instansvariabler är private.

Public

Alla kan använda metoderna, etc. Typiskt är att metoder är public.

Protected

Även om subclassen ligger i ett annat paket kommer man åt den. Detta gäller för just **subklasser**.

Klassvariabler

Instansvariabler har att göra med ett visst objekt. Ett exempel är konottnnehavare och saldo. Å andra sidan kanske vi undrar hur räntan är just nu och detta behöver vi inte upprepa för varje objekt. Vi vill ha en variabel som håller koll på räntan, är kopplad till ett konto, men inte är bunden till ett objekt!

Klassvariabler deklarereras inne i en klass men har prefix **static**!

Exempel Konto.java

```
public class Konto {  
  
    // klassvariabler  
    private static double räntesats;  
  
    // klassmetoder  
    public static double avläsRänta() {  
        return räntesats;  
    }  
  
    public static void ändraRänta(double nyRänta) {  
        räntesats = nyRänta;  
    }  
  
    // instansvariabler  
    private int kundNr;  
    private double saldo, intjänadRänta;  
    // konstruktor  
    public Konto(int kund) {  
        kundNr = kund;  
    }  
  
    // instansmetoder  
    public double avläsSaldo() {  
        return saldo;  
    }  
  
    public void transaktion (double belopp) {  
        if (belopp<0 && saldo+belopp<0) // negativt belopp => uttag  
            System.out.println("Uttag ej möjligt!");  
        else  
            saldo = saldo+belopp;  
    }  
  
    public void beräknaDagsränta() {  
        intjänadRänta = intjänadRänta + saldo*räntesats/100/365;  
    }  
  
    public void läggTillRänta() {  
        saldo = saldo+intjänadRänta;  
        intjänadRänta = 0;  
    }  
}
```