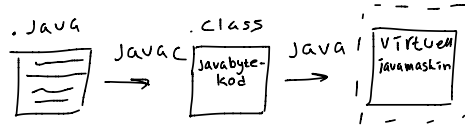


Kompilering av txt.java

Om du har en txt.java kompilerar man denna genom att (i kommandotolken) skriva "javac txt.java". Detta ger en .class-fil (javabytekod) vilken körs genom att i samma mapp skriva "java txt". Filen exekveras då i cmd.



Exempel Direkt1

```
public class Direkt1 {  
    public static void main (String [] arg) {  
        System.out.println("Välkommen till Java Direkt");  
    }  
}
```

Metod == funktion
Sats = statement (avslutas med ;)
Tilldelning = assignment

När vi kompilerar letar javac efter main i den class vi vill kompilera.

Exempel Hej2

```
import javax.swing.*;  
public class Hej2 {  
    public static void main (String[] arg) {  
        String namn;  
        String hälsning;  
        namn = JOptionPane.showInputDialog("Vad heter du?");  
        hälsning = "Välkommen " + namn;  
        JOptionPane.showMessageDialog(null, hälsning);  
    }  
}
```

namn och hälsning är två variabler av typen string.

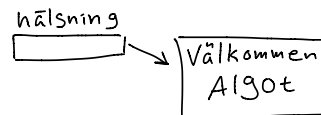
JOptionPane är en klass vilken innehåller showInputDialog (en metod)

showInputDialog lämnar ifrån sig ett resultatvärde, detta gäller dock inte för alla metoder (ex println i Direkt1)

showInputDialog finns i flera olika versioner, bland annat en med (null, "text") men showMessageDialog finns bara i versionen (null, "text") varför vi måste skriva null i parenteser



namn Variabel av typen string ger upphov till ett objekt av typen string.



Exempel Mobil

```
import javax.swing.*;  
  
public class Mobil {  
    public static void main (String[] arg) {  
        String s = JOptionPane.showInputDialog("Antal minuter per månad?");  
        int min = Integer.parseInt(s);  
        s = JOptionPane.showInputDialog("Pris per minut?");  
        double minutpris = Double.parseDouble(s);  
        double kostnad = min * minutpris;  
        JOptionPane.showMessageDialog(null, "Månadskostnad: " + kostnad);  
    }  
}
```