



# JAVAUTVECKLING – LEKTION 9

© 2016

Mahmud Al Hakim

[mahmud.al.hakim@nackademin.se](mailto:mahmud.al.hakim@nackademin.se)

[www.alhakim.se](http://www.alhakim.se)

## AGENDA

Klasser och objekt

Statiska/instans-variabler

## ENKEL KLASSDEKLARATION OCH INSTANSVARIABLER

```
public class Klassnamn {  
    typ1 instansvariabelnamn1;  
    typ2 instansvariabelnamn1;  
    ...  
}
```

Ordet `public` skall utelämnas om klassen inte ligger i en gen fil.

Variabler som deklarerats direkt i klassen, utanför metoder och som det inte står `static` framför kallas **instansvariabler**.

## ÖVNING 13.1

```
// Filen Bil.java  
public class Bil {  
    String regNr;  
    String fabrikat;  
    int årsmodell;  
    double tjänstevikt;  
    double effekt;  
}
```

## ATT SKAPA ETT OBJEKT

```
klassnamn namn; // deklarerar en referensvariabel
```

namn är namnet på variabeln vilken blir en referens (pekare).

```
namn = new klassnamn(); // skapa ett objekt
```

Här skapas ett nytt objekt av klassen Klassnamn.

Man kan skapa godtyckligt många objekt av en viss klass.

## ÖVNING 13.2

```
// Filen BilTest.java
```

```
public class BilTest {  
    public static void main(String[] arg) {  
        Bil b1 = new Bil();  
        Bil b2 = new Bil();  
    }  
}
```

## ÖVNING 13.3

```
Bil b1 = new Bil();
b1.regNr = "ABC123";
b1.fabrikat = "Citroen XSara Picasso";
b1.årsmodell = 2003;
b1.tjänstevikt = 1360;
b1.effekt = 115;
OptionPane.showMessageDialog(null, b1.regNr + "\n" +
b1.fabrikat + " " + b1.årsmodell + "\ntjänstevikt: " +
b1.tjänstevikt + "\nmotoreffekt: " + b1.effekt);
```

## TILLDELNINGAR OCH JÄMFÖRELSE AV OBJEKT

Om man gör en vanlig tilldelning kopieras bara **referensen** till objektet:

```
a = b ; // a kommer att peka på samma objekt som b
```

Vill man göra en kopiering av själva objektet kan man kopiera en instansvariabel i taget.

Om man gör en vanlig jämförelse jämförs referenserna till objekten:

```
a == b // lika endast om a och b pekar på samma objekt
```

Vill man jämföra objekten kan man jämföra en instansvariabel i taget.

## ÖVNING 13.4 – KLASSEN PERSON

```
// Filen Person.java
public class Person {
    String förnamn;
    String efternamn;
    int föddÅr;
    boolean singel = true;    // initiering
}
```

## ÖVNING 13.4 – KLASSEN BIL

```
// Filen Bil.java
public class Bil {
    String regNr;
    String fabrikat;
    int årsmodell;
    double tjänstevikt;
    double effekt;
    Person ägare;
}
```

## ÖVNING 13.4 – TESTPROGRAM

```
Bil b1 = new Bil();
b1.regNr = "ABC123";
b1.fabrikat = "Citroen XSara Picasso";
b1.årsmodell = 2010;
b1.tjänstevikt = 1360;
b1.effekt = 115;
b1.ägare = new Person();
b1.ägare.förnamn = "Lisa";
b1.ägare.efternamn = "Andersson";
b2.ägare.efternamn = "Olsson";
```

## KLASSVARIABLER

Klassvariabler deklarereras med ordet **static**

Finns bara i **en upplaga**, gemensam för alla objekt som tillhör klassen.

Skapas och initieras när programmet startar.

Existerar under resten av programmet.

Används ofta för att definiera konstanter.

För att komma åt dem skriver man **Klassnamn.variabelnamn**

## ÖVNING 13.5

```
public class Bil {  
    static double skattPerKg = 1.3;  
    String regNr;  
    String fabrikat;  
    int årsmodell;  
    double tjänstevikt;  
    double effekt;  
    Person ägare;  
}
```

## FÄLT MED REFERENSVARIABLER

```
Bil[] a = new Bil[3];  
for (int i=0; i<a.length; i++)  
    a[i] = new Bil();  
a[0].regNr = "zzz999";  
a[0].fabrikat = "Ford";  
a[0].årsmodell = 2011;  
a[0].tjänstevikt = 1400;  
a[0].effekt = 109;
```

## UPPGIFT 13.1

Deklarera en klass **Bok** som beskriver en bok.

Varje bok har en titel, en författare, ett sidantal och ett pris.

Skriv sedan ett program i vilket du skapar två objekt av klassen Bok.

Det ena objektet skall initieras så att det beskriver boken "Java Steg för steg".

Det andra objektet ska beskriva en annan valfri bok.

Låt slutligen programmet skriva ut informationen i Bok-objekten.