Variabler

I Java har vi tre typer variabler:

- Lokale variabler
- Instansvariabler
- Statiske variabler

Lokale variabler

En variabel som er deklarert inne i en blokk, metode eller konstruktør blir kalt en *lokal variabel*.

- Disse variablene blir opprettet når programmet går inn i blokken eller metoden den ligger i blir kalt, og blir destruert når programmet går ut av blokken, eller metodekallet blir returnert.
- Scopet til disse variablene eksisterer kun inne i blokken hvor variablen er deklarert, dvs. at vi kun kan bruke denne variablen inne i denne blokken.

Instansvariabler

Instantsvariabler er ikke-statiske variabler som blir deklarert i en klasse utenfor en metode, konstruktør eller blokk.

- En instansvariabel blir deklarert i en klasse. Disse variablene blir opprettet når en klasse blir opprettet, og destruert når en klasse blir destruert.
- I motsetning til lokale variabler, kan vi bruke *modifikatorer* på instansvariabler. Hvis vi ikke spesifiserer en *modifikator*, blir *default*-modifikatoren brukt.

Se på følgende program:

```
class Karakterer {
    int dat100;
    int mat100;
    int ing101;

    // Anta vi har en toString-metode
}

class Main {
    public static void main() {
        Karakterer obj1 = new Karakterer();
        obj1.dat100 = 80;
        obj1.mat100 = 45;
        obj1.ing101 = 20;
```

```
        Karakterer obj2 = new Karakterer();
        obj2.dat100 = 35;
        obj2.mat100 = 70;
        obj2.ing101 = 100;

        System.out.println(obj1.toString());
        System.out.prinln(obj2.toString());
    }
}
```

Hver enkelt klasse vi oppretter vil ha sine egne kopier av instansvariablene. Vi kan endre på disse variablene uten at det påvirker andre instanser av samme klasse. Derfor vil output være følgende:

```
Karakterer:
dat100 - 80
mat100 - 45
ing101 - 20

Karakterer:
dat100 - 35
mat100 - 70
ing101 - 100
```

Statiske variabler

Statiske variabler er også kjent som klassevariabler.

- Disse variablene blir deklarert ganske likt instansvariabler, men blir deklarert med nøkkelordet *static*, og *kun* i klassen, ikke i konstruktører eller metoder.
- Ulikt instansvariabler har vi kun en kopi av en statisk variabel per klasse, uahengig av hvor mange objekter vi oppretter.
- Statiske variabler blir opprettet i starten av et program, og destruert når programmet blir avsluttet.

For å bruke statiske variabler trenger vi ikke å lage et objekt fra en klasse. Vi kan enkelt få tak i variablen som:

`KlasseNavn.variabelNavn;`

Hvis vi gjør om programmet over:

```
class Karakterer {
    // Disse er nå statiske variabler
    static int dat100;
    static int mat100;
    static int ing101;

    // Anta vi har en toString-metode
```

```
}

class Main {
    public static void main() {

        Karakterer obj1 = new Karakterer();
        obj1.dat100 = 80;
        obj1.mat100 = 45;
        obj1.ing101 = 20;

        Karakterer obj2 = new Karakterer();
        obj2.dat100 = 35;
        obj2.mat100 = 70;
        obj2.ing101 = 100;

        System.out.println(obj1.toString());
        System.out.prinln(obj2.toString());
    }
}
```

Siden det kun finnes en kopi av statiske variabler per klasse, uavhengig av hvor mange objekter vi oppretter, vil output blir følgende:

```
Karakterer:
dat100 - 35
mat100 - 70
ing101 - 100

Karakterer:
dat100 - 35
mat100 - 70
ing101 - 100
```

Å bruke objekter til å endre på statiske variabler, slik som i programmet over, er generelt unødvendig og kan være forvirrende. Vi kan nemlig endre statiske variabler direkte, slik som i dette programmet:

```
class Karakterer {
    static int dat100;
    static int mat100;
    static int ing101;

    // Anta vi har en toString-metode
}

class Main {
    public static void main() {

        Karakterer.dat100 = 35;
        Karakterer.mat100 = 70;
```

```
        Karakterer.ing101 = 100;

        Karakterer obj1 = new Karakterer();
        System.out.println(obj1.toString())
    }
}
```

Hvor og når vi oppretter objektet `obj1` er irrelevant, da de statiske variablene i klassen har blitt satt og vil være de samme for alle objektene av denne klassen. Output vil da bli følgende:

```
Karakterer:
dat100 - 35
mat100 - 70
ing101 - 100
```