



PRG1000B Grunnleggende programmering

Leksjon 4

Metoder

Del 2

Roy M. Istad, 2017

Program, klasse og metode

Kode.java

```
import static java. ... // ulike klasser
public class Kode {
    main-metode

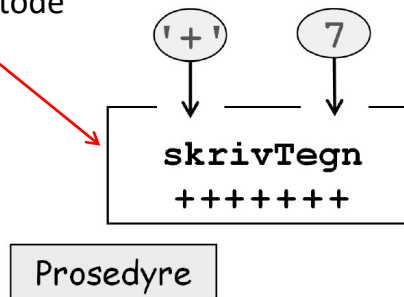
    hjelpemetode 1
    hjelpemetode 2
    :
    hjelpemetode N
}
```

Oppdeling av program

- Én fil
- Én klasse (import?)
- Én main-metode
- Én eller flere hjelpemetoder

«Prosedyre» kontra «funksjon»

void-metode



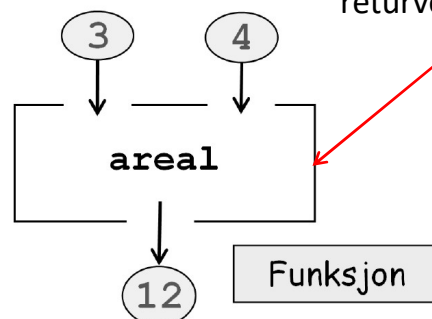
Metoden står som egen setning

```
private static void skrivTegn() {
    out.println("++++++");
}
```

6108 Programmering i Java

Leksjon 4

returverdi-metode



Metoden står som del av et uttrykk

```
private static int areal(int l, int b) {
    int flate = l*b;
    return flate; // evt. return l*b;
}
```

side 3

Parameterliste

Metodekall: skrivTegn('+', 7);

```
private static void skrivTegn(char t, int antall) {
    for (int i=1; i<=antall; i++)
        out.print(t);
    out.println();
}
```

Metodekode?

Konsoll

++++++

Samme metode, men andre aktuelle parametre i kallet:

skrivTegn('P', 12);

Konsoll

PPPPPPPPPPPP

6108 Programmering i Java

Leksjon 4

side 4

Oppgaver

1) Avgjør om et heltall er odde (et oddetall), eller ikke

Metodekall: `if (odde(7))`

2) Avgjør om et heltall er like (er partall), eller ikke

Bruke `odde()` inne i metoden?

3) Skifter bokstav fra liten til stor (andre tegn uendret), og motsatt

Metodekall:

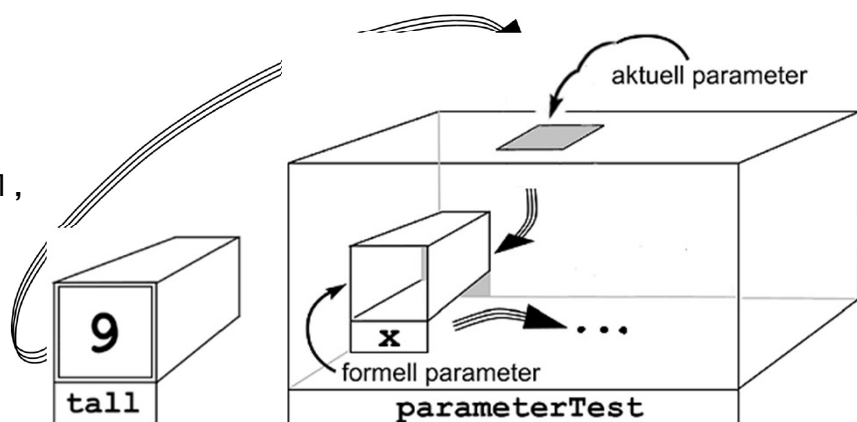
```
out.println( liten(bokstav) + " -> " + stor(bokstav) );
// Hvis bokstav: a -> A
```

4) Tverrsummen til et heltall ...

Verdioverføring av parameter

```
private static void parameterTest(int x) {
    x = x + 1;
    int w = x;
}
```

```
: // Før metodekall
int tall = 9;
parameterTest(tall);
: // Etter metodekall,
: // hva er tall nå?
:
```



Oppdeling av problem

- «**Topp-ned**»: Oppdeling av større og mer sammensatte problem i mindre deler.
- Metoder bidrar naturlig til mer oppdeling.
- «Dummy-metoder» (uferdig kode): Metoder som ikke gjør noe, men kompilerer, og kan altså brukes i løsningsforslaget.

- Eksempel: Sum øyne for tre terningkast (metoden returnerer bare en verdi)

```
private static int sumøyne(int antallTerninger) {
    return 11; // Dummy-metode, kompilerbar...
} // NB! Husk å fjerne static i EasyGraphic
```

- «**Bunn-opp**»: Bibliotek, pakke, klasse med metoder utviklet for standard bruk.
- Byggeklosser som kan settes sammen til store og gjerne kompliserte løsninger
 - Java API
 - Verktøyklasser (Samling statiske metoder – som vi muligens har laget selv)

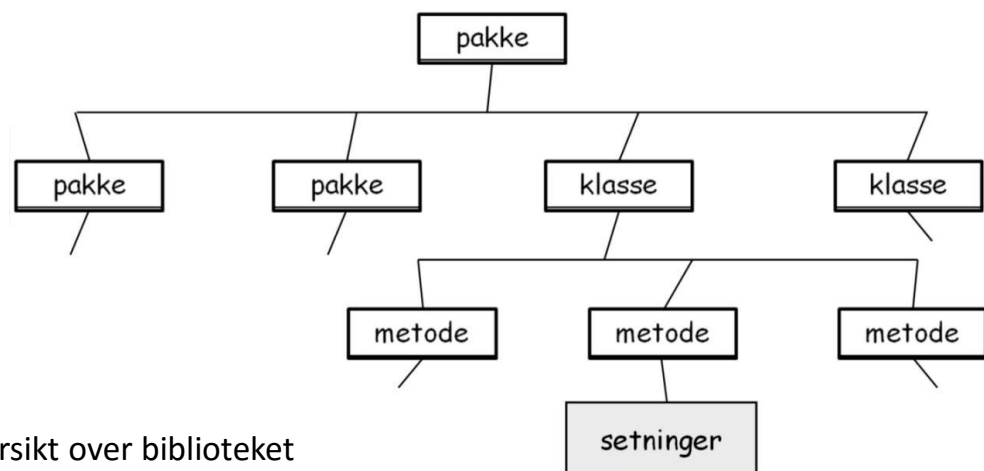
PRG1000B Grunnleggende programmering

Leksjon 4 – del 2

7

Programbibliotek

«Bunn-opp»-programmering: Standard byggeklosser ligger i *programbibliotek* slik at ulike program kan bruke dem etter behov.



Java-API: Oversikt over biblioteket

PRG1000B Grunnleggende programmering

Leksjon 4 – del 2

8

Klassemetoder og verktøyklasser

Egen fil!

Temperatur.java

```
public class Temperatur {
    public static double celsius2Fahrenheit(double c) {
        return (c * 9/5) + 32;
    }
    public static double fahrenheit2Celsius(double f) {
        return (f - 32) * 5/9;
    }
}
```

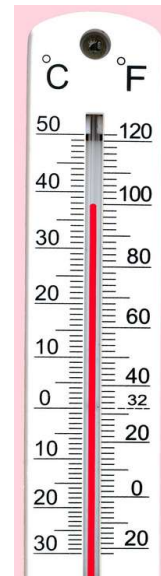
OBS! Trykkfeil i læreboka s.155

NB! Må sette *modifikatorene* til `public` for at andre klasser skal få tilgang til metodene (ikke `private` altså).

Og, klassen har ikke main-metode.

Kall på klassemetode fra et brukerprogram :

```
:
double grC = 20.0;
double grF = Temperatur.celsius2Fahrenheit(grC);
grF = Math.round(grF);
:
```



Fysisk tilgang til metodene? Import, plassert i samme mappe eller sette opp en filsti ?

PRG1000B Grunnleggende programmering

Leksjon 4 – del 2

9



PRG1000B Grunnleggende programmering

Slutt på leksjon 4 – Del 2