



PRG1000B Grunnleggende programmering

# Leksjon 3

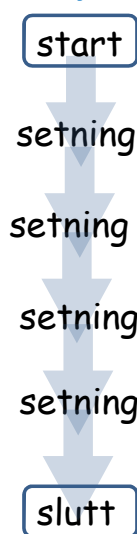
## Kontrollstrukturer

### Del 3

Roy M. Istad, 2017

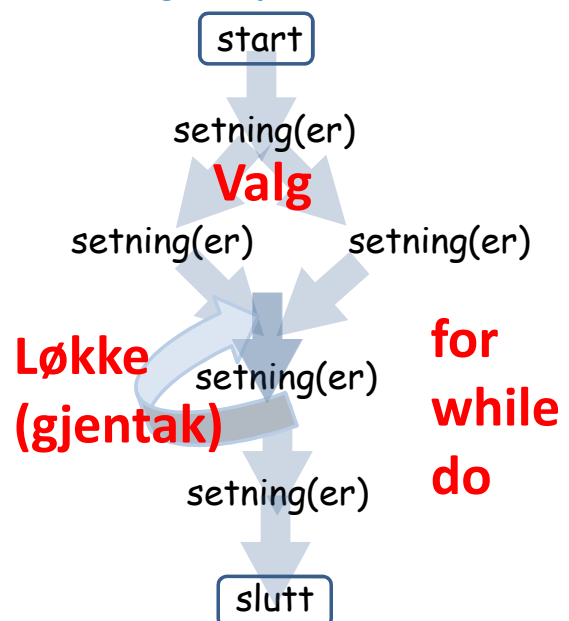
### Utførelse av et program

Programflyt så langt:



**Sekvensielt:**  
Alle setninger utføres, den ene etter den andre, og ingen blir hoppet over.

Programflyt fra nå av:



**Oppgave:** Skriv ut positive tall opp til og med innlest øvre grense

```
: // leser inn verdi til øvreGrense
for (int tall=1; tall<=øvreGrense; tall++)
    out.print(tall + " ");
```

7  
øvreGrense

Konsoll

1 2 3 4 5 6 7

**Oppgave:** Skriv ut positive oddetall opp til og med innlest øvre grense

```
: // leser inn verdi til øvreGrense
for (int tall=1; tall<=øvreGrense; tall+=2)
    if ( tall er odde )
        out.print(tall + " ");
```

Konsoll

1 3 5 7

**Oppgave:** Skrive ut de store engelske bokstavene

```
for (char bokstav='A'; bokstav<='Z'; bokstav++)
    out.print(bokstav + " ");
out.println();
```

Neste tegn i ASCII-tabellen  
(ordningsnummer + 1)

Store og små?

Konsoll

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvXxYyZz

## Oppsett av for-løkke

1. **initiering**; 2. **betingelse**; 3. **oppdatering**

for ( **initiering**; **betingelse**; **oppdatering** ) {  
 // Løkke kropp - setninger som skal gjentas  
}

**datatype** tellevariabel = start(verdi);

NB! En tellevariabel kan være deklartert tidligere i programmet

Stopp når logisk uttrykk blir false  
 Oppover: tellevariabel > stopp  
 Nedover: tellevariabel < stopp

start stopp  
 i++ i+=steg  
 start stopp  
 i-- i-=steg

Teller opp, eller ned – mye eller lite.

### Løkke kropp:

- 1) En enkelt setning: setning1;
- 2) Blokk: { setningsliste }

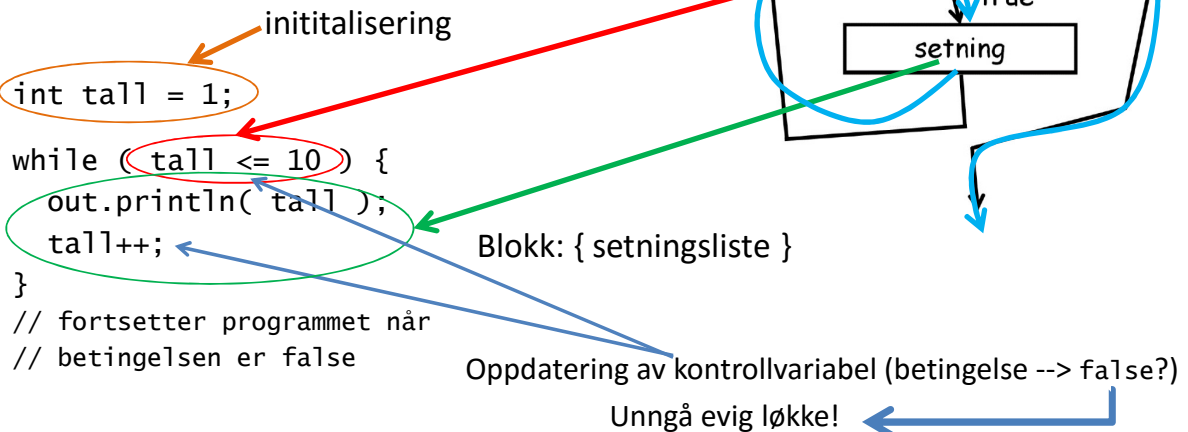
### Merk følgende oppsett:

```
for (int i=1; i<=4; i++)
    out.print(i + "*");
    out.println("-");
```

## 2. while-løkke

- Ukjent antall gjentak/utførelser

// Eks. skriver ut tallene 1 - 10



PRG1000B Grunnleggende programmering

Leksjon 3 - Del 3

side 5

**Eksempel:** Hvor lang tid tar det å doble verdien på et innskudd?

Trenger: innskudd, rentesats → beregner saldo for hvert år → doblingstid?

Ved 10% rente: 1000 → 1000 + 100 = 1100 → 1100 + 110 = 1210 → ...

: // leser innskudd og setter årsrente ... → 2000 (eller litt mer)

```
final double RENTESATS = 10.0;
```

```
double saldo = innskudd;
```

```
int år = 0;
```

```
while ( saldo < 2*innskudd ) {
```

```
    double årsrenter = saldo * RENTESATS / 100;
```

```
    saldo = saldo + årsrenter; // saldo += årsrenter;
```

```
    år++;
```

```
}
```

```
String utTxt = "Kr " + innskudd + " dobles i verdi ved "
```

```
    + "forrentning til " + RENTESATS + "% p.a. "
```

```
    + "i løpet av " + år + " år";
```

```
:
```

PRG1000B Grunnleggende programmering

Leksjon 3 - Del 3

side 6

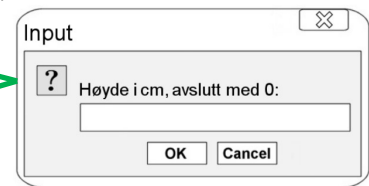
## Innlesing med while-løkke

```

: // Beregner gjennomsnittshøyde for en gruppe personer
String melding = "Høyde i cm, avslutt med 0:";
double sum = 0; // double for korrekt avrunding
int antall = 0; // Ingen høydemåling ennå
int høyde = parseInt( showInputDialog(melding) );
while ( høyde != 0 ) {
    sum += høyde;
    antall++;

    høyde = parseInt( showInputDialog(melding) );
}
showMessageDialog(null, "Gj. snitt: " + sum/antall);
:

```



PRG1000B Grunnleggende programmering

Leksjon 3 - Del 3

side 7

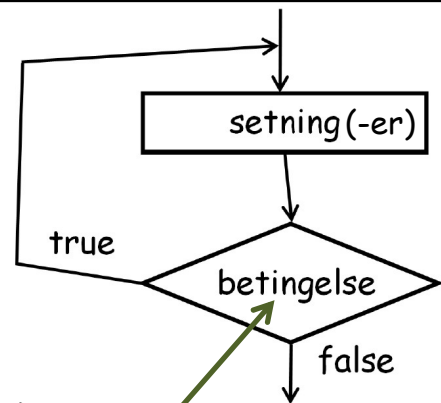
## 3. do-løkke

Velegnet når det er minst en utførelse.

```

:
String melding = "Resultat av terningkast:";
int sum = 0;
int antall = 0;
do {
    int verdi = parseInt( showInputDialog(melding) );
    sum += verdi;
    antall++; // antall kast utført
} while (sum < 50); // NB! Betingelsen står til slutt
showMessageDialog(null, "Antall kast til sum øyne >= 50: " + antall);
:

```



PRG1000B Grunnleggende programmering

Leksjon 3 - Del 3

side 8

## Lokale variabler og skopregler

- *Skopet*, eller virkeområdet, til en variabel er det området av koden der variabelen er "synlig" og kan brukes.
- Variabler deklart i en metode kalles for *lokale variabler*. Skopet til slike variabler er fra deklarasjonsstedet og ut til slutten av den aktuelle blokken.

Variabler kan ikke brukes utenfor blokken der de er deklart, enten det er i en løkke, i en if-grein eller i en else-grein, eller i en metodeblokk.

// x kan ikke brukes før blokken

Løkke- eller metodeblokk {

**Variabel x deklarerer og brukes herfra i blokken**

**skop** for lokale variabler

}

// x kan ikke brukes etter blokken

## Lovlighetskontroll av inndata

```
: // Låser utførelsen til en
: // lovlig verdi er innlest
int mnd = 0;
do {
    mnd = parseInt( showInputDialog("Månedsnr.(1-12):") );
} while (mnd < 1 || mnd > 12);
: // Nå er vi sikret at mnd inneholder et lovlig månedsnummer,
: // og programutførelsen kan fortsette
:
```

**OBS!** Variabelen `mnd` må deklarerer før do-løkke for å kunne brukes i betingelsen

Dersom inndata har feil format, f.eks. månedsnummer som en tekst (juni), så vil java-kjøresystemet avbryte programutførelsen og kaste en feilmelding à la:

**Exception in thread "main": java.lang.NumberFormatException**

```

:                                     Gjentak av hele program
boolean fortsett;
do {
:   // En setningsliste . . .
:   // Hele poenget med programmet skjer
:   // på linjene som står her
int svar = PI( SID("Gjenta program (Ja=1/Nei=0):") );
fortsett = (svar == 1);
} while (fortsett);
:

```

pI: parseInt  
sID: showInputDialog

Vi kommer senere tilbake til enklere brukerkommunikasjon for å avgjøre programgjentak eller ikke, noe a la:

```
fortsett: svar er lik "Ja"
```




---

## Slutt på leksjon 3 – Del 3

---