

## Trinn 4 – Bruker-input, ArrayList og Debugger

I forrige trinn lærte vi hvordan vi kunne opprette objekter. Vi opprettet bok-objekter. Men dataene for bøkene lagde vi i koden får. Det er jo mye bedre hvis dataene kan komme fra en bruker! La oss trene litt på det...

Også denne gangen har jeg laget en lengre intro-video, så se gjerne den før du begynner.

Mål for dette trinnet:

- Jeg kan benytte Scanner til å ta imot bruker-input.
- Jeg kan benytte en ArrayList til å
  - o ta vare på objekter
  - o gå igjennom objekter
- Jeg kan lage en bruker-meny med bruk av løkker
- Jeg har blitt litt kjent med debuggeren i IntelliJ

Kapittel 6 i pensumboka omhandler både array og ArrayList (og inkluderer noe bruk av Scanner).

[Her](#) er spørreskjemaet for trinn 4 der du kan fortelle hvordan det gikk.

### Oppgave 1

Opprett et nytt prosjekt. Legg til en klasse som har en main-metode.

### Oppgave 2

Opprett en klasse «Program». Program-klassen skal ha en public metode «runProgram» som ikke tar noen parametre, men som bare skriver ut (System.out.println – heretter kalt SOUT) «Program starting».

### Oppgave 3

I main-metoden din; opprett et objekt av klassen Program og kall på runProgram-metoden.

Kjør main-metoden din. Forhåpentligvis printer programmet ditt ut «Program starting» før det avslutter. Så bra! Da er du ferdig med forberedelsene 😊

### Oppgave 4

I klassen Program, lag en metode «oppgave4». Metoden skal opprette en lokal variabel av type Scanner, og den skal referere til et nytt Scanner-objekt som kan lese fra System.in. Dette er forklart i introvideoen, men du kan også lese mer [her](#).

Benytt Scanner-objektet ditt til å lese inn tre strenger. Men hvor skal vi plassere dem? Ettersom de er av samme type (String) så kan vi jo plassere dem i en ArrayList! Hurra! Opprett derfor en ArrayList som kan ta vare på strenger, og legg strengene der. Du kan lese mer om ArrayList [her](#).

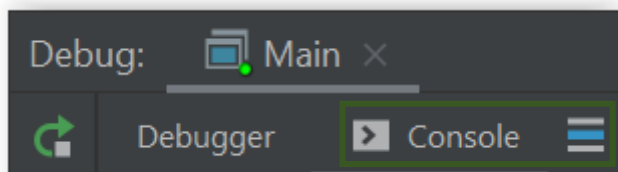
Men vi må jo få litt trening i å gå igjennom elementene i en ArrayList. Det er mange måter å gjøre det på (sjekk forrige lenke), men velg den måten du liker best. Gå igjennom alle elementene i arraylisten din, og skriv ut verdien (SOUT).

Men hvordan skal vi sjekke at dette fungerer? Dette skal du fundere på. Og for å sørge for at du tenker så det knaker (og kanskje finner det ut), så hjelper jeg deg med hvit tekst på hvit bakgrunn nedenfor. Gir du opp, så kopier teksten og paste inn et sted (f.eks Notepad) for å få hjelpen:

## Oppgave 5

Vi må ha litt trening i å bruke debugger. Den er nemlig helt rå. I main-metoden din, legg inn (helt i starten av metoden) en SOUT: «In the start of main()». Sett et break-point på denne kode-linjen (sjekk forelesning, eller les mer [her](#)). Start programmet ditt i debug-mode. Følg programmet ditt step by step. «Step into» metodene du har laget (runProgram og oppgave4). Underveis, se hvilke variabler som er opprettet og verdien på dem. Følg programmet helt til det avslutter.

Du vil oppdage at når du stepper inn i et steg som krever bruker-input, så må du gå til Console-fanen for å taste inn verdien din:



Ikke glem at debuggeren er gull hvis du sliter med å finne en feil i programmet ditt. Sett et break point i det området du antar at feilen din er, kjør programmet i debug mode, og du kan få masse verdifull hjelp til å finne den. (Dette gjelder kun programmer som kan kjøre, altså ikke har kompileringsfeil.)

Belo tutorial might help you 😊

<https://www.jetbrains.com/help/idea/debugging-your-first-java-application.html#setting-breakpoints>

## Oppgave 6

Lag en metode oppgave6 (i Program-klassen). Bruk en løkke (du velger selv hvilken type løkke) der du leser inn tall fra bruker. Ta vare på tallene som brukeren skriver inn. Løkken skal gå inntil brukeren taster inn et negativt tall. Når løkken er ferdig skal du skrive ut (SOUT) summen av tallene som ble tastet inn. Det negative tallet skal ikke inkluderes i summen. Hvis du synes dette var i overkant vanskelig, så har jeg litt mer herlig hvis tekst på hvit bakgrunn til deg nedenfor 😊 Men forsøk å løse oppgaven uten hjelp først.

## Oppgave 7

Dagens vanskeligste (før ekstra-oppgaven). Her er det bare å stålsette seg.

Lag en metode oppgave7. Metoden skal presentere en meny for brukeren:

1. Add single word
2. Display all words
3. End

Denne uken skal vi ikke ta høyde for at brukeren gjør noe galt (det kommer senere), så du kan forutsette at brukeren alltid vil klare å taste tallet 1, 2 eller 3.

Når brukeren taster 1, så skal du ta imot et ord fra brukeren og legge det i en arraylist.

Når brukeren taster 2, så skal du skrive ut (SOUT) alle verdiene i arraylisten.

Når brukeren taster 3, så skal du skrive ut (SOUT) en avskjedshilsen.

Test at metoden fungerer tilfredsstillende ved å legge inn noen tekster, og deretter velge alternativ 2 før du avslutter (3).

### Ekstraoppgave

Gå tilbake til ekstraoppgavene for trinn 3. Nå som vi har blitt introdusert for ArrayList kan vi erstatte arrayet vi benyttet i trinn 3 med en ArrayList. Da er det også mye enklere for oss å kunne fjerne bøker fra bok-registeret.

Utvid løsningen din (som du lagde i trinn 3) med mulighet **for bruker** å legge inn bøker. Lag en meny der bruker kan:

- Legge inn bøker
- Få oversikt over alle bøker
- Få oversikt over alle bøker av en spesifikk forfatter
- Få oversikt over alle bøker i en sjanger
- Få oversikt over alle bøker med en maksimal lesetid. Hvis du (som meg) synes at lesetid per side bør ligge plassert på Book og ikke Chapter, så flytt fieldet til Book.
- Fjerne bøker (kanskje legge inn et ekstra field ISBN i Book for å kunne fjerne basert på id?)

For å slippe å starte med et tomt register for hver gang du kjører programmet, så kan du i starten av programmet legge inn noen bøker via kode.

### Ekstraoppgave 2

Consider below ArrayList

```
ArrayList<String> nameList = new ArrayList<>();  
//Add elements to ArrayList object  
nameList.add("Kari");  
nameList.add("Lars");  
nameList.add("Markus");
```

How many different ways can you figure out to reverse this ArrayList?

Note: you should avoid using Collections.reverse(nameList)