

INF1000 - Obligatorisk innlevering 4

Temaer denne uka: *Innlesing fra fil og metoder.*

Oppgave 4.1)

Tema: *Innlesing fra fil*

Filnavn: `Innlesing.java`

- a) Lag et program som leser inn filen "winnie.txt" ved hjelp av Scanner og en løkke. Filen har et ord per linje.
- b) Tell antall forekomster av ordet "Winnie-the-Pooh" i teksten og skriv svaret til terminalen.
- c) Endre programmet slik at ordet som skal telles kommer fra brukeren ved hjelp av terminalen.

Synes du denne oppgaven var vanskelig? Se øvingsoppgaver 4.1.1. og 4.1.2.

Synes du denne oppgaven var lett? Se utfordringsoppgaver 4.1.3, 4.1.4 og 4.1.5.

Oppgave 4.2)

Tema: *Innlesing fra fil og metoder*

Filnavn: `Temperatur.java`

Vi har en fil (`temperatur.txt`) som inneholder 12 linjer hvor hver linje er gjennomsnittstemperaturen for en måned det siste året. Første linje tilsvarer januar, andre linje tilsvarer februar osv. For å gjøre det enklest har vi rundet av temperaturene til heltall.

- a) Les inn temperaturene en etter en og lagre dem i en array.
- b) Lag en metode som finner gjennomsnittstemperaturen i år og kall denne metoden.

Synes du denne oppgaven var vanskelig? Se øvingsoppgaver 4.2.1.

Synes du denne oppgaven var lett? Se utfordringsoppgaver 4.2.2.

Oppgave 4.3)

Tema: *Metoder med retur-verdi*

Tema: *Kalkulator.java*

Poenget med denne oppgaven er å øve på metoder med returverdi.

- a) Lag et program som heter Kalkulator og som inneholder en main-metode. Lag en metode "addisjon" som tar i mot to heltall og som plusser disse to tallene sammen og returnerer summen av disse to tallene.

Test programmet med å legge til følgende setning og print resultatet slik at du kan sjekke at resultatet er 7.

```
int addSvar = addisjon(3, 4);
```

- b) Lag en metode "subtraksjon" som tar i mot to heltall og som returnerer differansen mellom disse to tallene.

Test programmet med å legge til følgende setning og print resultatet slik at du kan sjekke at resultatet er 3.

```
int subSvar = subtraksjon(5, 2);
```

- c) Lag en metode "heltallsdivisjon" som tar i mot to heltall og som returnerer det første tallet delt på det andre tallet.

Test programmet med å legge til følgende setning og print resultatet slik at du kan sjekke at svaret er 3.

```
int helDivSvar = heltallsdivisjon(10, 3);
```

- d) Lag en metode "divisjon" som tar i mot to double verdier og som returnerer det første tallet delt på det andre tallet. (Hint: Her skal du ikke bruke heltallsdivisjon!)

Test programmet med å legge til følgende setning og print resultatet slik at du kan sjekke at svaret er 3.33333...

```
double divSvar = divisjon(10, 3);
```

Synes du denne oppgaven var vanskelig? Se øvingsoppgaver 4.3.1 og 4.3.2.

Synes du denne oppgaven var lett? Se utfordringsoppgaver 4.3.3.

Oppgave 4.4)

Tema: *Lek med arrayer*

Filnavn: *TicTacToe.java*

I denne oppgaven skal vi lage et enkelt "Tic Tac Toe"-spill, også kalt "3 på rad".

- a) Lag 3 arrayer som tar vare på heltall. Den første arrayen, `rad0`, skal representere den første raden, den andre arrayen, `rad1`, skal representere den andre raden og den tredje arrayen, `rad2` skal representere den tredje raden. Hver av arrayene skal ha plass til 3 tall.

Arrayen representerer spillet slik:

- 0 : Plassen er ledig
- 1 : Spiller1 har en brikke der
- 2 : Spiller2 har en brikke der

- b) Lag en brukermeny slik at to spillerne kan taste inn tall med hvor de vil plassere brikkene (ett tall for radnummer og ett tall for posisjon innad i raden). La først Spiller1 taste inn koordinat, deretter Spiller2. Dette skal skje vekselvis for Spiller1 og Spiller2 inntil brettet er fullt.
- c) Lag en metode som sjekker om en koordinat er ledig og lagre brikken til spilleren på brettet (i arrayen) dersom plassen er ledig.
- d) Når hele brettet er fullt, print ut hvordan brettet ser ut (se eksempelprintingen lenger ned).

Eksempelkjøring av programmet for de to første inntastingene:

```
Spiller1, Tast inn rad:
0
Spiller1, Tast inn plass:
0
Spiller2, Tast inn rad:
2
Spiller2, Tast inn plass:
1
Spiller1, Tast inn rad:
2
osv
```

Eksempelprinting:

```
1 1 2
2 1 2
2 1 1
```

Den neste oppgaven er frivillig:

- e) Lag en metode som finner ut om spillet er ferdig før alle brikkene er plassert og hvem som vant.

Synes du denne oppgaven var vanskelig? Se øvingsoppgaver 4.4.1, 4.4.2 og 4.4.3.
Synes du denne oppgaven var lett? Se utfordringsoppgaver 4.4.4.

Oppgave 4.5)

Tema: *Egen oppgave*

Filnavn: MinOppgave4.java

Lag en oppgave som handler om innlesing fra fil og metoder med returverdier.

Både oppgaveteksten du lager og besvarelsen på oppgaven skal leveres. Hvis du har laget en fil til innlesningen må du levere den også.

Fremgangsmåte for innleveringer i INF1000

1. Lag en fil som heter README.txt. Følgende spørsmål skal være besvart i filen:
 - Hvordan synes du innleveringen var? Hva var enkelt og hva var vanskelig?
 - Hvor lang tid (ca) brukte du på innleveringen?
 - Var det noen oppgaver du ikke fikk til? Hvis ja:
 - Hvilke(n) oppgave er det som ikke fungerer i innleveringen?
 - Hvorfor tror du at oppgaven ikke fungerer?
 - Hva ville du gjort for å få oppgaven til å fungere hvis du hadde mer tid?
2. Logg inn på [Devilry](#).
3. Lever de 5 .java-filene samt README.txt i *samme innlevering*.
4. Husk å trykke lever og sjekk deretter at innleveringen din er komplett.

Minstekrav for innleveringen er at .java-filene kompilerer og kjører på en av IFIs linuxmaskiner.

Den obligatoriske innleveringen er minimum av hva du bør ha programmert i løpet av en uke. Du finner flere oppgaver for denne uken [her](#) og flere utfordringsoppgaver [her](#).