

Oppgaver – Uke 2 – Klasser, I/O og Unntakshåndtering

1 (If-tester):

Skriv et program med to variabler a og b som er flyttall. Finn ut hvilke av tallene som er størst, og skriv det ut. Hvis begge to er like printer du det.

2 (For-løkker):

1. Lag et program som oppretter en array med plass til 10 int-er og fyller denne med alle sifrene mellom 0 og 9 i stigende rekkefølge. Skriv ut alle verdiene på hver sin linje.
2. Endre programmet slik at du får utskriften i motsatt rekkefølge
3. Endre nå programmet fra a slik at utskriften nå er tallene i arrayet og summen av alle tallene som har blitt skrevet ut til nå;
4. Endre programmet slik at tallet blir gjort om til en streng og at tallene blir lagt til i en string som samler alle tallene per iterasjon

Utskrift:

```
> 0
> 01
> 012
> 0123
> 01234
> 012345
> 0123456
> 01234567
> 012345678
> 0123456789
```

5. Lag et program som oppretter en array med plass til 10 string-objekter. Fyll dette arrayet med * hvor plass/index 0 har én * og hvor plass/index 9 har ti *-er, altså at hver plass inneholder plassnummer + 1 antall *-er. Skriv ut alle strengene på hver sin linje.

Utskrift:

```
> *
> **
> ***
> ****
> *****
> ****
> ***
> **
> *
> **
> ***
> ****
> *****
> ****
> ***
> **
> *
```

3 (I/O):

1. Gitt en fil kalt ord.txt som inneholder ett ord på hver linje. Lag et program som skriver ut alle linjene slått sammen til en setning/linje. Legg til et mellomrom etter hvert ord som blir lagt til.

Eksempel på ord.txt

Dette er en setning paa flere linjer.

2. Utvid programmet i oppgave a slik at brukeren kan skrive inn navnet på filen som skal leses

4 (Objekter og klasser):

Følgende klasseskjelett er gitt:

```
class Kanningaard {  
  
    private Kanin [] kaniner = new Kanin[100];  
  
    public boolean full() { . . . }  
    public boolean tom () { . . . }  
    public Kanin finnEn(String navn) { . . . }  
    public void settInn (Kanin kanin) { . . . }  
    public void fjern(String navn) { . . . }  
}
```

1. Lag en klasse kanin. Som har et navn som den tar inn ved opprettelse og en metode hentNavn() som returnerer navnet
2. Skriv ferdig klassen Kanningaard ved å fylle ut metodene. Dere kan gå ut fra at alle kaniner skal ha forskjellig navn, og derfor må dere ta høyde for at det ikke er lov å sette inn en kanin to ganger (altså med samme navn som en i lista). Dere skal også ta høyde for noen andre problemer som kan oppstå og gi gode feilmeldinger. To eksempler er dersom man forsøker å sette inn en eller når man forsøker å fjerne en kanin som ikke er i listen. Dere kan anta at klassen Kanin inneholder en metode *hentNavn*.
3. Skriv et hovedprogram der dere gjør minst tre tester av de forskjellige delene av klassen Kanningaard. Test for eksempel tilfeller som innsetting av kanin ved full kanningård eller fjerning av ikke-eksisterende kanin.
4. Tegn *datastrukturen* slik den ser ut etter at dere har kjørt hovedprogrammet deres. (*Tips*: Se notatet om datastrukturtegninger på emnets semesterside).