

INF1000 - Løsning på seminaroppgaver til uke 10

Oppgave 1 - Bit, byte og binær

- a) Bit er den grunnleggende enheten for digital informasjon. Den kan ha én av to mulige verdier, vanligvis representert ved «0» og «1».
- b) Det er vanligvis 8 bit i en byte.
- c) Både boolean og bit kan kun ha to verdier.
- d) Binærtall tolkes som en rekke av enkelt bit. Man leser den fra høyre, hvor den første verdien (0 eller 1) tilsvarer 1, andre plass 2, tredje plass 4 - for hver gang man beveger seg mot venstre opphører man to i ett høyere tall. Om man plusser sammen alle som er skrudd på (har verdien 1), får man tallet i det desimale tallsystemet.

Oppgave 2 - Tallsystemer og ASCII

- a) Hovedforskjellen mellom desimalverdier, heksadesimale verdier og binærverdier er hvilket tallsystem de bruker - henholdsvis titallsystemet (desimal), sekstentallsystemet (heksadesimal) og totaltallsystemet (binær).
- b) De 16 første verdiene i det heksadesimale tallsystemet er 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.
- c) De 5 første verdiene i binærtallsystemet er: 1, 10, 11, 100, 101.
- d) ASCII-tabellen er en tabell med oversikt over alle de 255 første tegnene som en datamaskin forstår. I ASCII-tabellen kan vi eksempelvis finne den desimale og heksadesimale verdien til bokstaven A.
- e) ASCII-tabellen av akkurat 255 tegn fordi det er så mange ulike tall som kan lagres i en byte (8 bit).
- f) Datatypen char har (oftest) 8 bit.

Oppgave 3 - Konvertering mellom tallsystemer

- a) Binærtallet 1001 som desimaltall er 9 (1+0+0+8).
- b) Desimaltallet 37 som binærtall er 100101.
- c) Summen av binærtallene 101 og 110 er 1011 (desimal: 11, heksadesimal: B).
- d) Det heksadesimale tallet 3F som et desimaltall er 62 (binær: 111111).
- e) 11000111 skrevet som heksadesimal verdi er C7 (desimal: 199).

Oppgave 4 - Gjennomløping / Traversering

- a) Skriv ferdig metoden under - som skal returnere gjennomsnittet til **int-arrayet** *tallrekke*:

```
public double finnGjennomsnitt(int[] tallrekke){
    double sum = 0;
    for (int i = 0; i < tallrekke.length; i++){
        sum += tallrekke[i];
    }
    double gjennomsnitt = sum / tallrekke.length;
    return gjennomsnitt;
}
```

- b) Skriv ferdig metoden *finnGjennomsnittligHoyde* under - som skal returnere gjennomsnittshøyden til en **ArrayList** av Person-objekter. Vi tenker oss at alle objekter av Person-klassen har følgende metode som en del av sitt grensesnitt:

```
public class Person{
    ...
    public double hentHoyde(){
        ...
    }
}

public double finnGjennomsnittligHoyde(ArrayList<Person> personliste){
    double sum = 0;
    for (Person p : personliste){
        sum += p.hentHoyde();
    }
    double gjennomsnitt = sum / personliste.size();
    return gjennomsnitt;
}
```

- c) Anta at du er gitt den samme Person-klassen som i oppgave b. Finn nå den gjennomsnittlige høyden til personene, nå som parameteren til metoden i stedet er en **HashMap** av Person-objekter:

```
public double finnGjennomsnittligHoyde(HashMap<String, Person> personliste){
    double sum = 0;
    for (Person p : personliste.values()){
        sum += p.hentHoyde();
    }
    double gjennomsnitt = sum / personliste.size();
    return gjennomsnitt;
}
```

- d) I ArrayList og HashMap bruker vi metoder (et grensesnitt) for å håndtere ting som størrelse, i array bruker vi indeksplasseringen. I ArrayList kan vi også bruke indeksen, selv om det her er det brukt en spesialisert for-løkke.