

# Løsningsforslag [ukeoppg. 7](#): 5 - 11. okt (INF1000 - Høst 2011)

Her finner du løsningsforslag til nøkkeloppgaven

Noen av oppgavene fra læreboka har løsningsforslag på [lærebokens hjemmeside...](#)

**Klasser og objekter del 2** (kap. 8 (hele) og 12 (hele) i "Rett på Java" 3. utg.)

**NB!** Legg merke til at disse er **løsningsforslag**. Løsningene dine trenger ikke å være like med disse forslag for å være riktige. Det er vanlig i programmering at samme oppgave kan løses på mange vidt forskjellige måter, og alle fremgangsmåter er ok i INF1000 så lenge de leder fram til riktig resultat og oppfyller kravene som står i oppgaveteksten.

## Mål

Få et første innblikk i hvordan man programmerer med klasser og objekter.

🔑 Oppgave merket med nøkkelsymbol er plukket ut som spesielt representativ for de viktigste temaene fra ukens forelesning, og alle bør ha som minimumsmål å løse denne selvstendig.

## Oppgaver

For de som ønsker å lese en annerledes og fin forklaring av klasser og objekter anbefaler vi "objekter.pdf" i [Marit Nybakkens notater](#), skrevet av en legendarisk tidligere gruppelærerinne i kurset.

- Oppgavene fra forrige uke:** (kap. 8, oppg. 1 - 6, side 175)  
Det er en god idé å starte med enkle oppgaver når du skal lære å programmere med objekter, så begynn med å løse ferdig de av [oppgave 0 - 6 fra forrige uke](#) som du ikke ble ferdig med.
- 🔑 **Modell av barneskole:** (kap. 8, oppg. 7, side 176)  
[ **Flere tips** til denne oppgaven finner du i [kommentar #11 - #18](#) i bloggposten "Ukeoppgaver". ]  
En forenklet modell av en barneskole er: «En barneskole har N klasserom (for eksempel N=17) og underviser barn i klasser på alle klassetrinn fra 1 til 7. På et klassetrinn er det et antall grupper. Det er maksimalt 28 elever i en gruppe. Det kan godt hende at det er *opptil* 4 grupper på et bestemt klassetrinn (eks. 2A, 3B, ...). En gruppe er alene om å bruke ett bestemt klasserom.»

(a) Lag et *klassediagram* i UML om opplysningene (klassene Skole, KlasseTrinn, og Gruppe).

(b) Programmér så klassene i Java slik at du ved hjelp av en heltallsarray-parameter til konstruktøren til klassen Skole får opprettet en array med pekere til syv KlasseTrinn-objekter (*Arrayen som brukes som parameter skal inneholde 7 elementer, som angir hvor mange grupper det er i hvert av klassetrinnene 1 - 7*). Hvert av disse (*KlasseTrinn-objekter*) har en array med plass til: fire pekere til hvert sitt *Gruppe*-objekt. La parameteren til *Skole-konstruktøren* bestemme hvor mange *Gruppe*-objekter du lager i hvert KlasseTrinn-objekt.

(c) Legg inn et navn i hvert *Gruppe*-objekt (1A, 4B, osv.). Lag en metode skrivUt() i hver av de tre klasseene, slik at du ved kall på skrivUt()-metoden i det ene Skole-objektet du har, får skrevet ut alle navnene på gruppene slik at du starter med førsteklassene.

```
// Løsningsforslag til deloppgave (b), og (c):
class Skole {
    KlasseTrinn[] trinn = new KlasseTrinn[7];
    int antTrinn = 7;

    public static void main(String[] args) {
        int[] antGrupperPerTrinn = { 4, 3, 3, 2, 2, 2, 1 };
        Skole s = new Skole(antGrupperPerTrinn);

        System.out.println("Disse klasserom ble opprettet:");
        s.skrivUt();
    }

    Skole(int[] antGrupperPerTrinn) {
```

```

    for (int i = 0; i < antGrupperPerTrinn.length; i++) {
        trinn[i] = new KlasseTrinn(); //1. gang: trinn 1 (1. kl)
        int ant = antGrupperPerTrinn[i]; //1. gang: 4
        for (int j = 0; j < ant; j++) {
            trinn[i].grupper[j] = new Gruppe();
            trinn[i].antGrupper++;
            trinn[i].grupper[j].klasserom
                = "" + (i + 1) + (char) (j + 'A');
        }
    }
}

void skrivUt() {
    for (int i = 0; i < antTrinn; i++) {
        trinn[i].skrivUt();
    }
}
}

class KlasseTrinn {
    Gruppe[] grupper = new Gruppe[4];
    int antGrupper;

    void skrivUt() {
        for (int i = 0; i < antGrupper; i++) {
            grupper[i].skrivUt();
        }
    }
}

class Gruppe {
    String klasserom; // 1A, 1B, ..., 2A, 2B, ...

    void skrivUt() {
        System.out.println("klasserom: " + klasserom);
    }
}
}

```

### 3. UML klasse- og objekt-diagrammer: (kap. 12, oppg. 1 - 5, side 239)

- **Oppgave 1:**  
En student har null eller flere eksamener. Lag et *klassediagram* om dette forholdet mellom klassene Student og Eksamen.
- **Oppgave 2:**  
Ane har frimerker fra forskjellige land i frimerkesamlingen sin. Lag et *objektdiagram* for et datasystem som skal kunne skrive ut hvilke frimerker Ane har fra et bestemt land, alle frimerkene hun har, og data om et bestemt frimerke.
- **Oppgave 3:**  
En B-bank har flere bankkonti, og en kunde kan ha flere bankkonti i B-bank.  
(a) Lag et *klassediagram* i UML over B-bank, kunder, og konti, og få med antall på forholdene mellom klassene.  
(b) Finn ut hvordan du vil lage en programstruktur som kan svare på spørsmål om hvilke bankkonti en kunde har i B-bank, og som kan skrive ut samlet saldo på alle konti for en kunde og samlet for B-bank. Tegn et *objektdiagram* for systemet.
- **Oppgave 4:**  
En forenklet modell av en barneskole er at den har  $N$  klasserom (for eksempel  $N = 17$ ) og underviser barn i klasser på alle klassetrinn fra 1 til 7. Det er ikke lov å ha flere enn 28 elever i én klasse. Det kan godt hende at det er mer enn én klasse på et bestemt klassetrinn (eks. 2A og 2B), og hver klasse har sitt eget klasserom.  
Tenk at du skal lage et system som håndterer at det begynner og slutter elever (skoleklasser kan bli opprettet og nedlagt). Tegn et *klassediagram* for problemet.
- **Oppgave 5:**  
(Se side 240 i læreboka)

**Tibakemelding** om dette oppgavesettet kan du [skrive i bloggen](#) eller sende på mail til josek [a] ifi.uio.no