

# Ukeoppgaver 6: 28. sep - 4. okt (INF1000 - Høst 2011)

*Klasser og objekter (kap. 8.1 - 8.14 i "Retten på Java" 3. utg.)*

## 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

### 0. KursRegister.java:

(a) Studér følgende program. Finn ut hva som er klassene, objektene, og pekerne i programmet, og hvordan disse er brukt. Undersøk gangen i programmet når det kjøres. Hvorfor skriver det ut forskjellig informasjon om to kurs når det bare er én System.out.println i programmet? Begge klassene lagres i én fil, kalt kursRegister.java.

```
class Kurs {
    // objektvariabler:
    String kode;
    int studiepoeng;

    // objekt-metode (dvs. uten "static"):
    void skrivUt() {
        System.out.println("Kurs med kode: " + kode
            + ", og studiepoeng: " + studiepoeng);
    }
}

class KursRegister {
    // klasse-metode (dvs. med "static"):
    public static void main(String[] args) {

        Kurs inf, mat; // pekere (variabler som kan peke på kurs-objekter)

        inf = new Kurs(); // Lager et objekt av klassen Kurs
        inf.kode = "INF1000"; // Setter verdier i objektet...
        inf.studiepoeng = 10;
        inf.skrivUt();

        mat = new Kurs(); // Lager et objekt til av klassen Kurs
        mat.kode = "MAT1001";
        mat.skrivUt();
    }
}
```

#### KJØREEKSEMPEL:

```
> java KursRegister
Kurs med kode: INF1000, og studiepoeng: 10
Kurs med kode: MAT1001, og studiepoeng: 0
```

(b) **Mange kurs:** Endre programmet slik at kursene lagres i en **array** av **Kurs**-pekere, i stedet for enkelt-pekere `inf` og `mat`. F.eks. skal første element i arrayen være `kurs[0]`, og koden til denne (`kurs[0].kode`) skal være "INF1000". Lag en løkke som oppretter tre kurs-objekter, setter verdien 10 som studiepoeng i alle, og setter kode til kursene ved å ta ett og ett navn fra følgende tilleggsarray:

```
String[] koder = { "INF1000", "MAT1001", "INF1080" };
```

Løkken skal til slutt kalle metoden `skrivUt()` for å skrive ut info om kursene som opprettes.

(c) **Legg til foreleser:** Anta at hvert kurs har én foreleser og at vi bare ønsker å lagre *navnet* på foreleseren i hvert kurs. Utvid programmet med objektvariabelen `foreleser` i klassen **Kurs**, og slik at verdien i denne initialiseres med følgende foreleser-navn for de tre kurs, henholdsvis "Ragnhild K.", "Erik L.", og "Roger A." (bruk gjerne en tilleggsarray slik som i del (b) ovenfor). Utvid også metoden `skrivUt()` slik at det får tak i og skriver ut navnet på foreleseren i hvert kurs.

(d) **Legg til studenter:** Anta at hvert kurs har maks. 200 studenter. Hvordan kan vi legge til informasjon om studentene i kursene? Undersøk følgende to muligheter: Legge til en array `String[] studenter`; vs. å lage en ny klasse **Student** med en objektvariabel `navn`. Hvilken av de to variantene vil være mest hensiktsmessig hvis vi vil lagre enda mer informasjon om hver student i systemet (i tillegg til `navn`)?

1. **Oppgave 1** i kapittel 8 (side 176)

Lag et program hvor `main` ligger i en klasse `Prog1`, og skriv en annen klasse `ABC` med et heltall `i`. Begge klassene skal være i samme fil (`Prog1.java`).

- (a) Deklarer en peker `pek` til `ABC`-klassen i `main`.
- (b) Lag et objekt av klassen `ABC`, og la `pek` peke på det objektet.
- (c) Sett verdien av `i` til 14 i dette objektet.
- (d) Skriv ut på skjermen verdien av `i` vha. en setning i `main`.
- (e) Deklarer en metode `doublet()` i `ABC` som dobler verdien av `i`.
- (f) Kall denne metoden fra `main` to ganger, og skriv så ut fra `main` verdien av `i` i objektet.

2. **Oppgave 2** i kapittel 8 (side 176)

Fjern klassen `ABC` fra fila `Prog1.java` og legg den på en egen fil, kalt `ABC.java`, i samme mappe.

- (a) Kompilér så `ABC.java` og `Prog1.java` hver for seg.
- (b) Kjør nå `Prog1.java` og se at du får det samme resultat som i Oppgave 1.

3. **Oppgave 3** i kapittel 8 (side 176)

- (a) Deklarer en objektvariabel `double x` i klassen `Prog1`.
- (b) Lag en objektmetode `double settx(double y)` i klassen `Prog1` som setter verdien av `x` til parameterens verdi, og som returnerer den gamle verdien av `x` før den fikk den nye verdien.
- (c) Lag en ny metode som tester `settx()` med 10 000 kall i en løkke.

4. **Oppgave 4** i kapittel 8 (side 177)

Utvid klassen `Konto` med `get`- og `set`-metoder for tekstene `eier` og `adresse` (dvs. metoder som kan returnere verdien, og sette nye verdier i disse objektvariabler).

```
class KontoEksempel {
    public static void main(String[] args) {
        Konto k1 = new Konto();
        k1.bestemKontonr();

        k1.settInn(500);
        System.out.println("Saldo er: " + k1.saldo);

        k1.taUt(300);
        System.out.println("Saldo er: " + k1.saldo);
    }
}

class Konto {
    int kontonr;
    int saldo;
    String eier, adresse;
    double rente = 2.5; // 2.5% per år
    static int nummer = 0; // klassevariabel

    void bestemKontonr() {
        nummer++;
        kontonr = nummer;
    }

    void settInn(int innskudd) {
        saldo = saldo + innskudd;
    }

    boolean taUt(int uttak) {
        if (uttak > saldo) {
            return false;
        }
        saldo = saldo - uttak;
        return true;
    }

    int getSaldo() {
        return saldo;
    }
}
```

```
KJØREEKSEMPEL:
> java KontoEksempel
Saldo er: 500
Saldo er: 200
```

5.  **Oppgave 5** i kapittel 8 (side 177)

Lag en metode `årsoppgjør()` som legger renten til saldo for et år (i programmet fra [oppgave 4](#)). Ikke gjør endringer på saldo direkte, men kall `settInn`-metoden fra `årsoppgjør`-metoden for å løse problemet.

6. **Oppgave 6** i kapittel 8 (side 177)

Lag to klasser A og B som har pekere til hverandre, og bruk disse pekerne til å kalle en metode i A fra B, og tilsvarende bruk pekeren i A til å kalle en metode i B. Begge disse metodene skal skrive ut en tekst på skjermen. Kjør programmet og se at du får riktig utskrift.

## Repetisjonsoppgaver

7. **Løkker: Gangetabell**

(a) Lag et program som ber bruker taste inn et tall, og skriver ut gangetabellen for det tallet, ganget med 1, 2, osv. til 12. Hvis bruker taster inn 5 skal resultatet se slik ut:

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
...osv...
```

(b) Endre programmet slik at utskriften av gangetabellen lages i en egen metode. Sett opp programmet med to klasser som vist i [oppgave 4](#) ovenfor.

8. **Tekster: Tallsiffer-oversetting:** *kap. 6, oppg. 7 og 8 (side 118)*

(a) Lag et program som oversetter fra tallsiffer til tekst slik at f.eks. 3 blir oversatt til "tre". Programmet skal kunne oversette alle 10 sifre (fra 0 til 9). *Hint:* Bruk en array med tekstene "null", "en", "to", osv.

(b) Lag et program som oversetter fra tekst til tall. Programmet skal be brukeren skrive inn et tall mellom null og ni (med bokstaver), og skrive ut tilsvarende siffer. *Hint:* Bruk arrayen fra del (a).

9. **Array med String-er:**

Hva blir skrevet ut i følgende program?, og hvorfor?

```
class NavneArray {
    public static void main(String[] args) {
        String[] navn = { "Anne", "Kari", "Ole", null };

        // (a)
        System.out.println(navn[1] + navn[navn.length/2]);

        // (b)
        for (int i = 0; i < navn.length; i++) {
            // Testen "!= null" sikrer at neste ledd ikke blir null.equals(..)
            if (navn[i] != null
                && (navn[i].equals("Ole") || navn[i].equals("Anne"))) {
                System.out.println(i);
            }
        }

        // (c)
        int indeks = 0;
        boolean funnet = false;
        while (indeks < 4 && !funnet) {
            if (navn[indeks].equals("Kari")) {
                funnet = true;
            }
            indeks++;
        }
        System.out.println(indeks);

        // (d)
        String[] andreNavn = { "Per", "Anne", "Ole" };
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 3; j++) {
                // Testen "!= null" sikrer at neste ledd ikke blir null.equals(..)
                if (navn[i] != null && navn[i].equals(andreNavn[j])) {
                    System.out.println(i + " " + j);
                }
            }
        }
    }
}
```

10. **Metode med array som inn-parameter:** *kap. 7, oppg. 3 (side 136)*

Lag en metode `double gjennomsnitt(int[] a)` som summerer alle elementene i heltallsarrayen `a`, og som returnerer (det aritmetiske) gjennomsnittet av verdiene i `a`.

---

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