

Ukeoppgaver 8: 18. - 22. okt (INF1000 - Høst 2010)

Repetisjon, og *mer om klasser og objekter* (kap. 8)

Mål

Få mer øvelse i bruk av klasser og objekter, og repetere teori fra tidligere.

Oppgaver til teoritimen

1. Spør læreren din:

- Spør gruppelæreren din om et tema du ønsker litt mer forklaring på, f.eks. noe du finner vanskelig i [Oblig 3](#).
- Gå gjennom [hint a-h](#) i Oblig 3.
- Gjør ferdig [Ukeoppgaver 6](#) og [7](#), særlig oppgave [nr. 3](#) i Ukeoppgaver 7 om feilmeldinger.

2. Utskrift av tabell: kap. 3, oppg. 2 (side 70)

- Lag et program som benytter programpakken easyIO til å skrive ut følgende tabell på skjermen. *Tips: Se side 51 i læreboka.*

Bilmerke	Årsmode11	Reg.nr.
Mercedes	1999	UE65660
Ford	2003	ZE95523
Toyota	2006	DK53401

- Løs samme oppgave ved hjelp av `printf` i stedet for EasyIO. Du kan bruke `%s` for å skrive ut en String, og `%-16` for å angi venstrejustering av String-en på 16 plasser, for eksempel:

```
System.out.printf("%-16s", "Ford");
```

3. Filer, store bokstaver, og args: kap. 3 (side 54), og oppg. 7 (side 71)

- Filkopi:** Lag et program som leser inn en fil og kopierer innholdet over til en annen fil. Filen skal leses inn *ett tegn av gangen* ved hjelp av `inChar()`. Du kan ta utgangspunkt i programmet vist under, fra side 54 i læreboka, som leser en fil et tegn av gangen, men skriver innholdet i filen ut på skjermen. Endre programmet slik at det skriver ut til til en annen, nyopprettet fil, i stedet for å skrive ut på skjermen; endre klassenavnet til `Kopi`, og utvid programmet slik at det ber bruker taste inn filnavnene for de to filene (original og kopi).

```
import easyIO.*;

class Tegnlaser {
    public static void main(String[] args) {
        In fil = new In("minfil.txt");
        int antall = 0;

        while (!fil.eofOfFile()) {
            char tegn = fil.inChar();
            System.out.print(tegn);
            antall++;
        }
        System.out.println("Antall tegn: " + antall);
    }
}
```

(b) **Store bokstaver:** Ta utgangspunkt i programmet vist ovenfor, og endre det slik at det skriver ut tegnene fra den innleste filen til skjerm, men med alle små bokstaver konvertert til *store bokstaver*. Følgende setninger viser hvordan man kan konvertere innholdet i en `char`-variabel `c` til store eller små bokstaver:

```
char c = 'x';
char c2 = Character.toUpperCase(c);
char c3 = Character.toLowerCase(c);
```

(c) **Antall ord:** Når man skriver artikler for publisering er det ofte grenser for hvor mange ord de kan inneholde. Lag et program som teller *antall ord* i en fil. Filnavnet kan du bruke inn når programmet starter.

(d) **args[]:** Lag deretter en annen utgave av programmet som tar filnavnet fra første *kommandolinjeargument*, dvs. `args[0]`. "Kommandolinjeargumenter" er evt. tilleggs-ord som bruker angir i selve `java`-kommandolinjen når hun kjører programmet. For eksempel, hvis bruker starter programmet med følgende kommando:

```
$ java Antallord fil.txt ord2
```

...så putter Java de to siste tilleggs-ord i arrayen `String args[]` (som vi har sett øverst i alle programmene våre til nå uten å bruke det). I dette tilfellet vil Java sørge for at `args[0]` får verdien `"fil.txt"` når programmet starter, og `args[1]` får verdien `"ord2"`.

4. Filbehandling linje for linje: (eksempel side 55 i læreboka)

(a) Studér følgende program, fra side 55 i læreboka, som leser en fil en linje av gangen, og skriver den ut på skjermen med linjenummer foran i hver linje. Endre programmet slik at det i stedet for å skrive ut alle linjene bare skriver ut en melding til slutt om *hvor mange linjer* og *hvor mange tegn* filen inneholder. For å telle antall tegn kan du bruke en variabel som summerer verdiene av `linje.length()`.

```
import easyIO.*;

class Linjeleser {
    public static void main(String[] args) {
        In fil = new In("minfil.txt");
        int linjenummer = 0;

        while (!fil.eofOfFile()) {
            String linje = fil.readLine();
            linjenummer++;
            System.out.println(linjenummer + " " + linje);
        }
    }
}
```

(b) Ta utgangspunkt i programmet vist ovenfor, og endre det slik at det skriver ut linjene på skjerm, men med alle små bokstaver konvertert til **store bokstaver**. Følgende setninger viser hvordan man kan konvertere bokstavene i en `String`-variabel til store bokstaver:

```
String s = "Jeg ER 18 år";
String s2 = s.toUpperCase();
// Nå er s2 tekststrengen "JEG ER 18 ÅR"
```

5. Public/private, klasse-/objekt-variabler: i kapittel 8 (side 175)

(a) **Private:** Vi tar en ny titt på bankkonto-programmet fra [oppgave 4](#) i Ukeoppgaver 6. Endre deklarasjonen av `saldo` i klassen `Konto` til å være *private*, og vis hvordan vi da kan få tak i `saldo` fra den andre klassen. *Tips: Bruk get-metoden.*

```

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

```

(b) **Klassevariabler:** De fleste metoder og variabler i programmet ovenfor er *objekt*-variabler og *objekt*-metoder, men det er én *klasse*-variabel og én *klasse*-metode i programmet. Finn disse, og diskuter hvordan de er annerledes enn objekt-variantene. Hvordan fungerer klasse-variabelen som står i programmet, og hva ville skjedd hvis vi tok bort nøkkelordet *static* fra deklarasjonen av variabelen?

Oppgaver til terminaltiden

1. Tabell, filer, args, private, og static:

(Samme oppgave som [nr. 1](#) til [nr. 5](#) for teoritimen.)

2. Fortsett med [Oblig 3](#).

NB! I Oblig 3 er det veldig viktig at du skriver din **egen løsning** selv. Det er lov å *diskutere* med andre hvordan du tenker å gå frem for å løse deloppgavene, men det er ikke lov å *kopiere* programbiter fra obliger skrevet av andre, heller ikke hvis du deretter endrer på disse programbitene. **Hvis du samarbeidet tett med noen andre skriv navnene deres** i en kommentar øverst i programmet. Andre typer kilder bør også nevnes hvis det ikke er kurs-websidene, lærebøker, eller lærere og orakler i kurset.

3. Ukens nøtt er [Oblig 3-Pi](#).

Løsningsforslag

Kommer...

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