UiO : Universitetet i Oslo

Ukeoppgaver 4: 20. - 24. sep (INF1000 - Høst 2010)

Teori for Oblig 2: *Mer om* ***løkker og arrayer*** *(kap. 4 - 5); og* ***metoder*** *(kap. 7.1 - 7.9)*

Mål

Få øvelse i teorien som trengs for å løse **Oblig 2**. Hovedtema i obligen er løkker, arrayer, og metoder uten parametre, og følgende oppgaver gir nyttig trening i dette slik at du blir bedre i stand til å løse obligen.

Oppgaver til teoritimen

1. **Enkle løkker med tall:**
   (a) Nedtelling: Lag et program med en løkke som teller ned fra 10 til 0. Bruk en teller-variabel med startverdi 10 og redusér den med 1 i hver gjennomgang av løkka. Utskriften skal bli:

```
 ...10 ...9 ...8 ...7 ...6 ...5 ...4 ...3 ...2 ...1 ...0
```

   (b) Dobling: Lag en løkke som starter med å skrive ut tallet 2, og deretter dobler tallet, skriver det ut, og gjentar prosessen helt til 10 tall er skrevet ut. Bruk en teller-variabel "`i`" som teller de 10 gangene, og en annen variabel for tallet som ganges med 2 i hver omgang av løkka og skrives ut. Utskriften skal bli:

```
2  4  8  16  32  64  128  256  512  1024
```

   (c) Tabell: Lag to nestede for-løkker som gir følgende utskrift. Det skal bare skrives ut ett tall av gangen, som skal være verdien til telleren i den ytre løkka (som går fra 1 til 3 og skal hete *rad*) ganget med telleren i den indre løkka.

```
1  2  3  4
2  4  6  8
3  6  9  12
```

2. **2D-array (to-dimensjonal array) med tall:**
   (a) Utskrift: Lag to nestede for-løkker som setter inn følgende verdier i en 2D-array og skriver de ut. Arrayen skal deklareres slik: `int[][] tabell = new int[3][4];`. Se oppgave nr. 1 (c) over for tips om disse verdiene. Husk at indeksene i arrayen starter fra 0, ikke 1. Verdien i `tabell[2][3]` skal være 12.

```
1  2  3  4
2  4  6  8
3  6  9  12
```

   (b) Sum av kolonne: Skriv programkode som beregner summen av verdiene i hver kolonne i ovennevnte array og skriver summene ut. *Dette ligner på en deloppgave 5 oblig 2!*

```
6 12 18 24
```

   (c) Søk etter tall: Utvid programmet slik at det ber bruker taste inn et tall og deretter ser om tallet finnes i arrayen. Hvis tallet finnes skal alle array-indeksene der det ligger skrives ut, hvis ikke skal det gis beskjed om at tallet ikke finnes.

```
Hvilket tall søker du: 8
Tallet finnes i tabell[1][3]
```

3. **Blokker og skop:**
   Hvilke av disse programmene er lovlige?

```
class Prog1 {
    public static void main(String[] args) {
        int k = 0;
        if (k >= 0) {
            int n = k + 1;
        }
        System.out.println(n);
    }
}

class Prog2 {
    public static void main(String[] args) {
        int k = 0;
        if (k >= 0) {
            int n = k + 1;
        }
        if (k < 0) {
            System.out.println(n);
        }
    }
}

class Prog3 {
    public static void main(String[] args) {
        int k = 0;
        if (k >= 0) {
            k++;
            System.out.println(k);
        }
    }
}
```

4. **inLine(), inChar(), og ordreløkke:**
   Lag et program som spør bruker etter et navn. Programmet leser navnet inn vha. `inLine();`, og "fyller" så skjermen med navnet ved å skrive det ut 100 ganger. Videre skal programmet spørre bruker om hun vil gi et nytt navn `(j/n)`. Svaret leses nå med `.inChar("\n\r")`, og hvis det er `'j'` gjentas hele prosessen; hvis svaret er `'n'` avsluttes programmet. Kjøreeksempel:

```
Skriv et navn: Ola Nordmann
Ola Nordmann Ola Nordmann Ola Nordmann Ola Nordmann Ola Nordmann O
la Nordmann Ola Nordmann Ola Nordmann Ola Nordmann Ola Nordmann Ol
a Nordmann Ola Nordmann Ola Nordmann Ola Nordmann Ola  ...osv...
Gi nytt navn? (j/n): j
Skriv et navn:
```

   Tips til Oblig 2: Legg merke til det som står i parentesene til `.inChar("\n\r")` ovenfor. Dette angir at `inChar` skal hoppe over evt. linjeskift (`\n`) eller vognretur (`\r`) som bruker har tastet inn, og i stedet lese inn en vanlig bokstav fra tastatur, f.eks. `'j'`. Det samme bør du gjøre i deloppgave 1 *(c)* i oblig 2.

5. **Metoder:**

(a) Endre strukturen til programmet ditt fra oppgave nr. 4 over slik at det følger malen fra oblig 2 (vist i skissen nedenfor), dvs. med en liten kontrollklasse øverst, etterfulgt av en egen klasse for metodene. Hele programmet med begge klassene lagres i én fil, kalt `Navn100.java` (dvs. navnet til klassen med metoden `main()`). Lag bare én metode i hjelpeklassen, kalt `ordreløkke()`, som gjør alt som står i nr. 4 ovenfor.

```
import easyIO.*;

class Navn100 {
    public static void main(String[] args) {
        Hjelpeklasse hj = new Hjelpeklasse();
        hj.ordreløkke(); // Kjører metoden ordreløkke() i hjelpeklassen
    }
}

class Hjelpeklasse {
    // Klargjøring for innlesing/utskrift, gjelder for hele klassen:
    In tast = new In();
    Out skjerm = new Out();

    String navn;

    void ordreløkke() {
        char giNyttNavn = 'j'; // Startverdi

        while (giNyttNavn != 'n') {
            // - Be bruker taste et navn og les det inn med .inLine();
            // - Utskrift av navn 100 ganger.
            // - Spør om bruker vil "Gi nytt navn? (j/n):", og .inChar("\n\r"):


        }
    }
}
```

Mer info: Grunnen til at vi skriver programmet på denne måten med to klasser vil bli klarere når vi kommer til kapittel 8, men har sammenheng med at vi ønsker å lage gode "objektorienterte" program der vi jobber med "objekter". I denne skissen er det fem pekere til objekter: args[], hj, tast, skjerm, og navn.

(b) Flere metoder: Endre programmet slik at koden som skriver ut navnet 100 ganger flyttes til en egen metode kalt `utskrift()`. Husk å legge inn et kall på metoden på det stedet i programmet du flyttet koden fra.

(c) Inn-parameter: Endre programmet slik at det også spør bruker hvor mange ganger navnet skal skrives ut. Verdien som bruker taster skal overføres som parameter til metoden `utskrift()` som du lagde i del (b). Endre også metoden `utskrift(..)` slik at den tar imot argumentet vha. parameteren `int antall`, slik: `void utskrift(int antall) {`

Oppgaver til terminaltimen

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