

## Ukeoppgaver 5: 27. sep - 1. okt (INF1000 - Høst 2010)

Mer om **filer** (kap 3), **tekster** (kap. 6), og **metoder** (kap. 7)

### Mål

Samle sammen trådene med grunnleggende programmering (kap. 1 til 7) før vi starter med klasser og objekter neste uke.

### Oppgaver til teoritimen

#### 1. Metode med array som inn-parameter: kap. 7, oppg. 3 (side 134)

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

#### 2. Lese to arrayer fra fil: kap. 5, oppg. 3 (side 97)

Lag et program som skal behandle data om vekten til elevene i en skoleklasse. Det er 27 elever i klassen. Dataene ligger [på fil](#) slik:

```
Jens 52
Marit 43
...
```

(a) Les dataene inn i to arrayer: en navnearray (`String`) og en vektarray (`int`). **Hint:** Se `// Fil til array` på side 60 i læreboka.

(b) La programmet finne høyeste og laveste vekt og skrive ut navn og vekt på disse.

(c) La programmet beregne gjennomsnittsvekten i klassen. **Hint:** Bruk `gjennomsnitt`-metoden fra oppgave [nr. 1](#) ovenfor.

#### 3. Tekster: Hva blir skrevet ut?

```
class Tekster {
    public static void main(String[] args) {
        Out skjerm = new Out();

        String s1 = "hei";
        String s2 = "Java";
        String[] navn = { "Rune", "Martin", "", "Guro" };

        /* a */ System.out.println(navn.length + s1.length());

        /* b */ System.out.println(3.1415 + "" + 'x');

        /* c */ System.out.println("" + false);

        /* d */ System.out.println("" + ! "abc".equals("abc"));

        /* e */ System.out.println("heia" == (s1 + "a"));

        /* f */ System.out.println("heia".equals(s1 + s2.charAt(1)));

        /* g */ System.out.println(s1.equals("h" + navn[0].charAt(3) + 'i'));

        /* h */ System.out.println(navn[1].substring(1));

        /* i */ System.out.println(navn[1].substring(1, 4));

        /* j */ System.out.println(s2.replace('a', 'i'));

        /* k */ System.out.println(navn[1].indexOf("tin"));

        /* l */ System.out.println(navn[2].indexOf("tin"));

        /* m */ System.out.println("A".compareTo("A"));

        /* n */ if (s1.compareTo("zz") < 0) System.out.println("s1 alfabetisk foran");

        /* o */ System.out.println(navn[3].toUpperCase());

        /* p */ if ("hei på deg".startsWith(s1)) { System.out.println("ja"); }

        /* q */ int x = Integer.parseInt("123"); System.out.println(x + 1);
    }
}
```

4. **Ord baklengs:** kap. 6, oppg. 3(a-b), (side 115)(a) Lag et program som skriver ut teksten «Agnes i senga» baklengs. *Hint:* Bruk en for-løkke som teller nedover.

(b) Modifiser programmet over slik at det først gjør om teksten til kun å inneholde små bokstaver.

5. **Mer om metoder**

Fullfør følgende program, som viser bruk av metoder. Angi også hva programmet skriver ut.

```

import java.util.Scanner; // Tilsvarer: import easyIO.*;

class Metoder {
    public static void main(String[] args) {
        TestMetoder tm = new TestMetoder();
        tm.start();
    }
}

class TestMetoder {
    Scanner tast = new Scanner(System.in); // Tilsvarer: In tast = new In();

    void start() {
        // Kaller en enkel metode:
        metode1();

        // Kaller en metode med én inn-parameter:
        skrivTredoblet(123);

        // Leser to tall fra tastatur, og overfører de til en metode
        // som finner og skriver ut det høyeste av de to tall:
        skjerm.out("Skriv to tall (f.eks. 7 4): ");
        int tall1 = tast.nextInt(); // Tilsvarer: ... = tast.inInt();
        int tall2 = tast.nextInt(); // Tilsvarer: ... = tast.inInt();
        finnHøyesteAv2(tall1, tall2);

        // Kaller en metode som multipliserer de samme to tall lest
        // inn ovenfor, og returnerer resultatet hit.
        int resultat = multipliser( /* Fyll inn resten. . . */ );
        skjerm.outln("Resultat multiplisert: " + resultat);

        // Metode med array som inn-parameter:
        double[] verdier = { 0, -3, 5, 10, -20, -7.7, 1.2, -0.01 };
        //int antNeg = finnAntallNegativeTall(verdier);
        //skjerm.outln("Arrayen verdier[] har " + antNeg + " negative tall.");
        // <- Ta bort "/" i de to linjene over.
    }

    void metode1() {
        skjerm.outln("Dette er metode1");
    }

    void skrivTredoblet(int x) {
        int tredoblet = x * 3;
        skjerm.outln("Tredoblet resultat = " + tredoblet);
    }

    void finnHøyesteAv2(int a, int b) {
        // Hva mangler her?
        // . . .

        skjerm.outln("Høyest av de to tall er:" /* . . . */ );
    }

    int multipliser( /* Fyll inn resten. . . */ ) {
        // . . .

        return 0; // . . . Erstatt 0 med resultatet av x ganger y.
    }

    // Skriv metoden "finnAntallNegativeTall" her, som har en array med
    // double-verdier som inn-parameter, finner ut hvor mange av verdiene
    // i arrayen er negative tall, og returnerer det antallet (som en int).
    // . . .
}

```

**6. Finn feil:**

Når vi prøver å compilere og kjøre dette programmet får vi feilmeldingene vist nedenfor. Hva er feil?

```

1 class FinnFeil {
2     int[] fat = new int[5];
3     System.out.println("Feilplassert");
4
5     public static void main(String[] args) {
6         FinnFeil ff = new FinnFeil();
7         ff.metode();
8     }
9
10    void metode() {
11        for (int i = 0; i < 5; i++) {
12            fat[i] = 10 * i;
13        }
14
15        System.out.println(fat[5]);
16        System.out.println(fat2);
17        System.out.println(fat[1], 2);
18    }
19 }

```

Hva betyr disse feilmeldingene, og hvordan retter vi feilene?

```

(a)
FinnFeil.java:3: <identifiser> expected
    System.out.println("Feilplassert");
                   ^
FinnFeil.java:3: illegal start of type
    System.out.println("Feilplassert");^

(b)
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
    at FinnFeil.metode(FinnFeil.java:15)
    at FinnFeil.main(FinnFeil.java:7)

(c)
FinnFeil.java:16: cannot find symbol
symbol   : variable fat2
location: class FinnFeil
    System.out.println(fat2);
                   ^

(d)
FinnFeil.java:17: cannot find symbol
symbol   : method println(int,int)
location: class java.io.PrintStream
    System.out.println(fat[1], 2);
                   ^

```

**7. Metode med inn og ut-parametre: kap. 7, oppg. 2 (side 133)**

Lag en metode som regner ut hypotenusen  $c$  i en rettvinklet trekant når vi går ut fra Pytagoras formel:  $c^2 = a^2 + b^2$  der  $a$  og  $b$  er lengden på katetene – de to andre sidene i trekanten. Vi bruker  $a$  og  $b$  som parametre til metoden. Du trenger da å kalle kvadratrotmetoden i `Math.sqrt(double verdi)` i den metoden du lager. Returner verdien  $c$  som verdien på metoden. Test metoden ved å kalle den i en dobbel for-løkke for alle kombinasjoner av  $a$  og  $b$  med heltallsverdiene fra 1.0 til og med 6.0, og skriv ut svarene.

```

double finnHypotenus(double a, double b) {
    // ...
    return c;
}

```

**Hint:** Kallet på metoden kan se slik ut: `double c = finnHypotenus(a, b);` ...eller: `System.out.println(finnHypotenus(a, b));`

**Oppgaver til terminaltimen****1. Filer, tekster, og metoder:**

(Samme som oppgave [nr. 2](#), [nr. 4](#), og [nr. 5](#) for teoritimen.)

**2. Tallsiffer-oversetting: kap. 6, oppg. 7 og 8 (side 116)**

(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).

**3. Finn feil:***(Samme som oppgave [nr. 6](#) for teoritimen.)***4. Fortsett med [Oblig 2](#):****5. Pyramide med EasyIO:** *kap. 3, oppg. 1 (side 70)*Løs oppgave 5 fra kapittel 1 (side 24) ved å bruke formatert utskrift (**Hint:** Skriv ut alle linjene med samme bredde, og bruk `Out.CENTER` for å sentrere utskriften av hver linje).**Hint II:** *Se øverst på side 52.*

Vi ønsker å skrive ut følgende tekst på skjermen (med samme pyramideformatering):

```

  a
 a a
a b a
a c c a
a d f d a

```

Lag et Java-program som gjør dette, ved å bruke flere utskriftssetninger.

**6. Ekstraoppgave: Pyramide uten EasyIO**System.out.printf tilbyr mange flere måter å formatere utskrift på enn EasyIO, inkludert høyre- og venstre-justering (f.eks. kodene "%7s" og "%-7s" skriver ut en String-verdi henholdsvis høyre- og venstre-justert på 7 plasser), men det er en formatering i EasyIO som printf ikke har direkte %-kode for: sentrering som vist i [foregående oppgave](#).*(a)* Lag en metode som implementerer denne formateringen ved hjelp av printf (og uten EasyIO), metoden skal ta imot 2 parametre: teksten som skal skrives ut sentrert, og antall plasser man skal bruke. *Hint: Husk at tekst.length() gir deg lengden på teksten.**(b)* Bruk metoden du lagde i *(a)* til å løse [foregående oppgave](#) uten bruk av EasyIO.**7. Ukens nøtt: Sudoku hjelpeprogram***(a)* Lag et program som leser inn en Sudoku-oppgave fra fil og lagrer de forhåndsutfylte tallene i en 2-dimensjonal array. Deretter går programmet i en løkke som spør brukeren om et rad- og et kolonnennummer (i området 1-9, eller 0 for å avslutte). Programmet skal så svare brukeren med hvilke tall (1-9) som er mulige kandidater for plassering i den angitte rad/kolonne-plassen, ved å finne ut hvilke av sifrene 1-9 ikke er allerede brukt i samme rad, kolonne, eller 3×3-omsluttende boks.

Input-filen er på 9 linjer, med 9 tall per linje adskilt med mellomrom, og hvor 0 angir plassene som ikke har forhåndsutfylt siffer i Sudoku-oppgaven. Her er et eksempel på en slik fil (med middels vanskelig oppgave). Finn gjerne andre oppgaver fra aviser eller nettet.

```

6 0 7 0 0 0 8 0
0 0 0 1 0 4 0 7 0
0 0 5 0 0 8 0 3 0
8 0 0 3 0 0 7 0 0
4 0 0 5 0 6 0 0 8
0 0 1 0 0 2 0 0 6
0 8 0 4 0 0 5 0 0
0 9 0 2 0 3 0 0 0
0 7 0 0 0 0 1 0 3

```

*(b)* Utvid deretter programmet slik at det går gjennom *alle* ikke-utfylte ruter (i stedet for å be brukeren taste en), og for de rutene som bare har ett kandidatsiffer setter du sifferet inn i arrayen. Gjenta prosessen med de gjenværende ikke-utfylte ruter helt til to påfølgende gjennomkjøringer ikke finner nye tall å sette inn. Skriv ut resultatbrettet til slutt.**Løsningsforslag***Kommer...*Tilbakemelding om dette oppgavesettet kan du [skrive i bloggen](#) eller sende på mail til josek [a] ifi.uio.no