


# Ukeoppgaver 5: 21. - 27. sep (INF1000 - Høst 2011)

Filer; tekst (kapittel 3 og 6 i "Rett på Java" 3. utg.)

## Mål

Få øvelse i teorien du trenger for å løse Oblig 2 (løkker, arrayer, tekster, og metoder); og i tillegg litt om innlesing og utskrift til fil.

 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

### 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 doubler 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 deklarerer 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 i 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. Tekster: Antall tegn, store bokstaver, og like tekster kap. 6, oppg. 1 (side 117)

Lag et program som:

(a) ber brukeren skrive inn en tekst og beregner antall tegn i teksten.

(b) ber brukeren om å skrive inn en tekst og lager en ny tekst av den gamle med bare store bokstaver, og skriver ut den nye teksten. (*Tips: Se spørsmål "o" i oppgave 5 nedenfor.*)

(c) ber brukeren skrive inn to ord og tester om disse er like.

### 4. String og charAt: Ord baklengs kap. 6, oppg. 3 (a)-(b), (side 117)

(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. 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);
    }
}
```

## 6. Innlesing og utskrift til fil: (eksempel side 52 i læreboka)

(a) Ta utgangspunkt i følgende program, fra side 52 i læreboka, som skriver fire typer verdier til skjerm (tegn, linje, heltall, og desimaltall), og endre programmet slik at utskriften går til en fil i stedet. Kall utfilen "minfil.txt".

```
import easyIO.*;

class Utskrift {
    public static void main(String[] args) {
        Out skjerm = new Out();
        skjerm.outln('A');
        skjerm.outln("Canis familiaris betyr hund");
        skjerm.outln(15);
        skjerm.outln(3.1415, 2); // Bruk to desimaler
    }
}
```

*Hint:*

- **EasyIO:** Utskrift til fil programmeres nesten på samme måte som utskrift til skjerm, bare at man angir filnavn i klargjøringen av "new Out", bruker et annet navn på forbindelsen (f.eks. "utfil" i stedet for "skjerm"), og lukker filen til slutt. Mer detaljert: Klargjør for utskrift til fil med f.eks. "Out utfil = new Out("minfil.txt");" Deretter bruker du utfil.out/outln for utskrift til fil på samme måte som du ville brukt skjerm.out/outln for utskrift til skjerm. Til slutt lukker du filen med utfil.close();. Hvis man glemmer close blir ingenting lagret i filen.
- **PrintWriter/printf:** **NB!** Dette er bare ment for spesielt interesserte som vil prøve uten EasyIO. Klargjør for utskrift til fil med "import java.io.\*;" og "PrintWriter utfil = new PrintWriter(new File("minfil.txt"));". Deretter bruker du utfil.print/println/printf for utskrift til fil på samme måte som du ville brukt System.out.print/println/printf for utskrift til skjerm. Til slutt utfil.close(); I tillegg bør du ha "try{" i begynnelsen av metoden (før setningen med "new PrintWriter") og følgende på slutten av metoden (du vil lære mer om dette i INF1010, foreløpig er det nok at du bare skriver det helt likt): "}" catch (Exception e) { e.printStackTrace(); }".

(b) Utvid programmet slik at det også leser innholdet i filen som ble opprettet i del (a), lagrer det i fire passende variabler (den første skal være av typen char, osv.), og skriver til slutt verdiene i variablene ut på skjerm.

*Hint:* Innlesing fra fil programmeres som innlesing fra tastatur, bare at man angir filnavnet under klargjøring, f.eks. "In innfil = new In("minfil.txt");", og bruker gjerne et annet navn på forbindelsen (f.eks. "innfil" i stedet for "tast"). Hvis du vil prøve uten EasyIO klargjør med "Scanner innfil = new Scanner(new File("minfil.txt"));" og legg try- og catch rundt metoden (se hintet i del (a)).

7.  **Lese to arrayer fra fil:** kap. 5, oppg. 3 (side 99)

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 62 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.

8. **Finn feil:**

Når vi prøver å kompilere 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> forventet
    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);
                       ^
```

9. **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(doble_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));`

10. **Mer om metoder**

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

```

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

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

class TestMetoder {
    Scanner tast = new Scanner(System.in); // Tilsvare: 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(); // Tilsvare: ... = tast.inInt();
        int tall2 = tast.nextInt(); // Tilsvare: ... = 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).
    // . . .
}

```

## 11. Fibonacci-tallene

(a) **Løkker:** Lag et program som skriver ut de 15 første tall i [Fibonaccifølgen](#). Følgen er definert ved at de to første tall er 0 og 1, og hvert neste tall er summen av de to foregående. Utskriften skal bli:

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

Mer info: Fibonacci-tallene forekommer mye i [naturen](#), bl.a. i tregrener, blomster, kongler og [kaniner](#).

(b) **Metoder:** Endre strukturen i Fibonacci-programmet du lagde [ovenfor](#) slik at det følger malen fra Oblig 2 (vist også i oppgave [nr. 8](#) fra forrige uke), med en kontrollklasse øverst, etterfulgt av en hjelpeklasse for metodene. Bruk tre metoder i hjelpeklassen: en ordreløkke som skriver ut følgende meny, og en metode for hvert av de 2 menyvalgene:

1. Skriv ut de 15 første tall i Fibonaccifølgen
2. Test om et tall hører til følgen

Metoden for ordre 2 ber bruker taste inn et tall, og går så i en løkke som genererer Fibonacci-tallene frem til det bruker-inntastede tallet. Deretter gis det melding til bruker om tallet hørte til følgen eller ikke.