

Ukeoppgaver 3: 13. - 17. sep (INF1000 - Høst 2010)

Forgreninger (kap. 4.1 - 4.2), **løkker** (kap. 4.3 - 4.4), **arrayer** (kap. 5.1 - 5.5 og 5.7), **tekster**, (og **filer**).

Mål

Øve på bruk av forgreninger, løkker, arrayer, og tekster.

Oppgaver til teoritimen

1. Enkel kalkulator med if-else og switch:

(a) Lag en kalkulator som støtter enkle regnestykker på formen: *tall operator tall*, hvor de tre elementene er adskilt med mellomrom. *Tall* er heltall, og *operator* er en av de fire regneartene: + - * /. Eksempel på kjøring av programmet:

```
Regnestykke: 4 + 5
Resultat: 9
```

Bruk `tast.inInt()` og `tast.inChar(" ")` for å lese tall og regneart fra tastaturet. Mellomrommet i parentesene til `inChar` angir at den skal betrakte mellomrom som et *skilletegn* før og etter regnearten. Dette er nødvendig å si fra til `inChar`, fordi den leser inn ett tegn av gangen. Lagre tallene og operator i passende variabler, og bruk **if-else**-setninger til å velge hvilken regneart skal utføres. *P.S. Det vil bli lagt ut løsningsforslag både med og uten bruk av easyIO.*

(b) Hvordan kan programmet endres for å bruke en **switch**-setning i stedet for if-else? (*Hint*: Se eksemplet på side 78 i læreboka, og husk å ta med `break`-setningene.)

(c) Hva slags divisjon får vi utført?, og hvordan kan vi endre programmet for å få utført den andre typen divisjon uten å endre deklarasjonen av de to innleste heltall?

2. Løkker: Hva blir skrevet ut?

Avgjør uten å bruke datamaskin hva som blir skrevet ut når følgende programsetninger utføres.

```
// (a)
int a = 10;
while (a < 20) {
    a += 4;
}
System.out.println("a = " + a);

// (b)
int sum = 0;
for (int b = 1; b < 6; b += 2) {
    sum += b;
}
System.out.println("sum = " + sum);

// (c)
int produkt = 1;
for (int c = 1; c < 4; c++) {
    produkt = produkt * c;
    System.out.println(produkt);
}

// (d)
for (int d = 3; d >= 1; d--) {
    for (int e = 1; e <= 3; e++) {
        System.out.println(d + e);
    }
}

// (e)
int teller = 0;
for (int ytre = 0; ytre < 3; ytre++) {
    teller++;
    for (int indre = 0; indre < 3; indre++) {
        teller++;
    }
}
System.out.println(teller);
```

3. Array med tall:

(a) Lag et program som ber bruker taste inn tre heltall og lagrer disse i en array kalt *tall*:

```
int[] tall = new int[3];
```

(b) Sum av array: Utvid programmet slik at det regner ut summen av tallene ved hjelp av en løkke, og skriver ut resultatet.

(c) Minste verdi: Utvid programmet slik at det finner og skriver ut det minste tallet i arrayen.

(d) Lave verdier: Legg til programkode som skriver ut alle verdiene i arrayen som er mindre enn 10.

(e) Søk: Legg til programkode som skriver ut en beskjed om verdien 5 finnes eller ikke finnes i arrayen.

4. 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 andre ledd ikke blir null.equals(..)
                if (navn[i] != null && navn[i].equals(andreNavn[j])) {
                    System.out.println(i + " " + j);
                }
            }
        }
    }
}
```

5. Innlesing og utskrift til fil: (eksempel side 50 i læreboka)

(a) Ta utgangspunkt i følgende program, fra side 50 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: Utskrift til fil vha. easyIO programmeres på samme måte som utskrift til skjerm, bare at man i tillegg angir filnavnet – f.eks. `new Out("minfil.txt")`, – og lukker filen til slutt (`utfil.close()`). Navnet på forbindelsen `skjerm` bør også endres for å gjøre det tydelig at man skriver til en `utfil`.

(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, men man må i tillegg angi filnavnet – f.eks. `new In("minfil.txt")`.

6. Flerdimensjonal array:

To-dimensjonale arrayer brukes på samme måte som en-dimensjonale, men har to sett med klammer, og gjennomgås best vha. to nestede for-løkker:

```
import easyIO.*;
class Array2D {
    public static void main(String[] args) {

        int[][] b = new int[3][4];

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 4; j++) {

                b[i][j] = i * j;

                System.out.println( <Hva mangler her?> );
            }
        }
    }
}
```

Arrayen kan illustreres slik:

	0	1	2	3
0				
1			X	
2				

KJØREEKSEMPEL:
`> java Array2D`
`b[0][0] = 0`
`b[0][1] = 0`
`b[0][2] = 0`
`b[0][3] = 0`
`b[1][0] = 0`
`b[1][1] = 1`
`b[1][2] = 2`
`b[1][3] = 3`
`b[2][0] = 0`
`b[2][1] = 2`
`b[2][2] = 4`
`b[2][3] = 6`

(a) Fullfør `println`-setningen slik at programmet gir utskriften vist under Kjøreeksempel.

(b) Hvor mange elementer er det plass til i arrayen `b` vist over?

(c) Hva er indeksene til elementet markert med "x" i figuren, og hvilken verdi får elementet utdelt i programmet?

Oppgaver til terminaltiden

1. Enkel kalkulator med if-else og switch:

(a) - (c) (Samme oppgaver som i [nr. 1](#) for teoritimen.)

(d) Test programmet på datamaskin og se hvilke feilmeldinger du får når du prøver å dele et tall på 0, både med og uten løsningen din på deloppgave (c) ovenfor.

2. Utskrift av oddetalls-array: kap. 5, oppg. 1 (side 97)

Skriv et program som inneholder en heltalls-array med følgende elementer: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19. Programmet skal inneholde en løkke som skriver ut indeksen og verdien for alle elementene i arrayen.

3. Sum av elementene i en array: kap. 5, oppg. 2 (side 97)

Vi bruker her samme array som i forrige oppgave: Beregn summen av elementene og skriv ut resultatet.

4. Søke etter tall i array:

(a) Utvid programmet fra forrige oppgave slik at det ber bruker taste inn et tall, og deretter skriver ut en beskjed om det inntastede tallet

finnes eller ikke finnes i arrayen.

(b) Legg også til en løkke som skriver ut alle verdiene i arrayen som er mindre enn det inntastede tallet.

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

(Samme oppgave som [nr. 4](#) for teoritimen.)

6. **Innlesing og utskrift til fil:** (eksempel side 50 i læreboka)

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

7. Begynn med [Oblig 2](#).

8. **Ukens nøtt:** (vanskelig!)

Lag et program som ber om 5 tall fra bruker, og deretter finner og skriver ut hvilke tall som er gjentatt blant disse. F.eks. hvis bruker tastet inn 6 6 3 6 3, så skal programmet gi meldingen: Tall som er gjentatt: 6 3

9. **Tips til Emacs:** (for spesielt interesserte)

Emacs-forkortelser: Legg til følgende fem linjer i din ~/ .emacs-fil:

```
(define-abbrev-table 'java-mode-abbrev-table '(
  ("psv" "public static void main(String[] args) {" nil 0)
  ("sop" "System.out.println" nil 0)
))
(abbrev-mode 1)
```

Deretter starter du Emacs på nytt. Når du skriver psv i et Java-program (etterfulgt av mellomrom eller linjeskift) så vil det nå bli utvidet til: public static void main(String[] args) {, og tilsvarende for sop etterfulgt av (. Legg gjerne til flere forkortelser. Mer info om filen ~/ .emacs kan du finne i [Ukeoppgaver 2](#).

Undo: For å angre siste redigering trykk *C-_* (dvs. *Ctrl-understrek*), eller klikk på ikonet med bilde av en «bøyd pil» øverst i Emacs-vinduet.

Copy/paste: For å kopiere tekst fra et hvilket som helst sted på skjermen til Emacs, start med å markere teksten vha. musa. Deretter flytter du mus-pekeren til det stedet i Emacs-vinduet der du vil lime inn teksten, og trykker musens midt-knapp (dvs. hjul-tasten) rett ned. Ferdig! Du trenger altså ikke trykke *Ctrl-c* eller *høyreklikk* > *Copy* for å velge teksten på Emacs, det er nok å markere det.

Cut/paste: Hvis du vil klippe bort tekst og flytte det til et annet sted i Emacs-vinduet: markér teksten; trykk *Delete*-tasten eller *C-w* for å klippe det bort; flytt tekst-markøren til ønsket sted; og trykk *Insert*-tasten eller *C-y* for å lime inn.

Splittet vindu: For å kunne se to filer samtidig kan du dele Emacs-vinduet i to ved å trykke *C-x 2* ("*C-*" står for *Ctrl*-tasten). For å gå tilbake til å vise én fil klikk med musa på ønsket del av splitt-vinduet og trykk *C-x 1*.

Flere vinduer: For å åpne et ekstra-Emacs-vindu slik at du kan se to filer samtidig enda lettere trykk *C-x 52*. Husk *C-x C-f* for å åpne en fil i det nye vinduet.

Innrykk: Du kan la Emacs sette riktig innrykk i hele programmet ved å trykke *C-x h* og deretter velge i menyene øverst: *Java* > *Indent Line or Region*.

Flere tips til Emacs kan du finne i [Emacs-oppgavene fra Forkurs i informatikk](#)

Løsningsforslag

Kommer...

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