

FASIT/LØSNINGSFORSLAG (kommentarene er for læringsformål og avspeiler ikke nødvendigvis hva som ble vektlagt ved sensuren denne gang). Forslaget er laget av Ole Christian Lingjærde.

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i :	INF1000 — Grunnkurs i objektorientert programmering
Eksamensdag :	Torsdag 2. desember 2004
Tid for eksamen :	09.00 – 12.00
Oppgavesettet er på :	10 sider
Vedlegg :	Ingen
Tillatte hjelpemidler :	Alle trykte og skrevne

(Trykkfeil som ble oppdaget etter eksamen er korrigert nedenfor. Ved sensuren ble det tatt hensyn til slike forhold).

- Les gjennom hele oppgaven før du begynner å løse oppgaven. Kontroller at oppgavesettet er komplett før du begynner å besvare det. Dersom du savner opplysninger i oppgaven, kan du selv legge dine egne forutsetninger til grunn og gjøre rimelige antagelser, så lenge de ikke bryter med oppgavens "ånd". Gjør i såfall rede for forutsetningene og antagelsene du gjør.
- Dine svar skal skrives på disse oppgavearkene, og ikke på separate ark. Dette gjelder både spørsmål med avkryssningssvar og spørsmål hvor du bes om å skrive programkode. I de oppgavene hvor det skal skrives programkode, anbefales det at du først skriver en kladd på eget ark før du fører svaret inn i disse oppgavearkene på avsatt plass.
- Noen av spørsmålene er flervalgsoppgaver. På disse oppgavene får du poeng etter hvor mange korrekte svar du gir. Du får ikke poeng hvis du lar være å besvare et spørsmål, eller dersom du krysser av begge svaralternativer.
- Hvis du har satt et kryss i en avkrysningsboks og etterpå finner ut at du ikke ønsket å krysse av der, kan du skrive "FEIL" like til venstre for den aktuelle avkrysningsboksen.
- Husk å skrive såpass hardt at besvarelsen blir mulig å lese på alle gjennomslagsarkene, men ikke legg andre deler av eksamensoppgaven under når du skriver.
- Med unntak av oppgave 7 behøver du ikke skrive kommentarer til programkoden.

Oppgave 1

Hvor mange doubleverdier er det plass til i hver av disse arrayene?

a) `double[][] tabellA = new double[9][9];`

Svar:⁸¹.....

b) `double[][] tabellB = new double[10][2];`

Svar:²⁰.....

c) `double[][] tabellC = new double[5][1];`

Svar:⁵.....

Oppgave 2

Er disse programsetningene lovlige i Java?

- | JA | NEI |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> <code>int[] a = null;</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>int x = new int[50];</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>int[] x = int[50];</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>int[] verdier = new verdier[50];</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>int[] verdier = new int;</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>double[] x = new int[5];</code> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> <code>String[] s = {"januar", "februar", "mars"};</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>double[] x = new double{1.0, 2.0, 3.0, 4.0, 5.0};</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>double[] x, []y;</code> |

Oppgave 3

Anta at følgende kodelinjer utføres:

```
int a = 33;
int b = 65;
a = a + b;
b = a + b;
boolean samme = (a == b);
```

Hva er verdien til variabelen `samme` rett etterpå?

Svar: `false`

Oppgave 4

Skriv ferdig metoden under, som med utgangspunkt i en pris med moms skal beregne og returnere pris uten moms. Momssatsen skal antas å være 24%.

Svar:

```
double prisUtenMoms(double prisMedMoms) {

    double momssats = 0.24;
    return prisMedMoms / (1 + momssats);

}
```

Oppgave 5

Anta at følgende array er deklart i en metode i et program:

```
int[] a = new int[100];
```

Anta videre at **a** er fylt opp med ulike verdier, og at verdien 0 forekommer nøyaktig en gang i arrayen. Vil alternativene under alltid gi som resultat at **a[k]** inneholder verdien 0?

- | JA | NEI | |
|-------------------------------------|-------------------------------------|--|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <pre>int k = 0; while (k < a.length && a[k] == 0) { k++; }</pre> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | <pre>int k = 0; while (k < a.length && a[k] != 0) { k++; }</pre> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <pre>int k = 0; do { k++; } while (a[k] != 0);</pre> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <pre>boolean funnet = false; int k = 0; while (k < a.length && !funnet) { if (a[k] == 0) { funnet = true; } }</pre> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <pre>int k = 0; for (int i = 0; i < a.length; i++) { if (a[k] == 0) { k = i; } else { k++; } }</pre> |

Oppgave 6

Anta at følgende setninger utføres:

```
int sum = 9;
int verdi = sum++;
sum = ++sum + verdi++;
verdi += sum * 10;
```

Hva er verdien til variabelen **verdi** like etter at setningene over er utført?

Svar: 210.....

Oppgave 7

Her skal du både skrive **javadoc**-kommentarer og programkode.

- Skriv ferdig metoden under som skal returnere med gjennomsnittet av verdiene i arrayen **data** (dvs summen av alle elementene i **data** delt på antallet slike elementer). Du kan anta at lengden av arrayen er minst 1.
- Skriv en kort **javadoc**-kommentar til metoden som sier hva den gjør (returnerer av verdi) og hva parameteren er.

Svar:

```

/**
 * Returnerer gjennomsnitt (aritmetisk middelvei) av tallene i parameter-arrayen.
 *
 * @param data en array med double-verdier
 * @return aritmetisk middelvei av tallene i parameter-arrayen
 */

double middelvei(double[] data) {

    double sum = 0.0;          // Vanlig feil: glemme å initialisere denne til 0

    for (int i = 0; i < data.length; i++) {
        sum += data[i];
    }

    return sum / data.length; // Vi trenger ikke å sjekke om data.length==0 siden
                             // det skal antas at lengden av arrayen er minst 1.
}

```

Oppgave 8

Gitt to variabler

```
int i1, i2;
```

som initialiseres til hver sin verdi. Skriv noen programsetninger som samlet sett bytter om innholdet av variablene **i1** og **i2**. Du kan godt deklare ekstre variable hvis du trenger det.

Svar:

```
int tmp = i1;
i1 = i2;
i2 = tmp;
```

Oppgave 9

Skriv ferdig metoden under som skal bytte om på elementene i arrayen `x` slik at elementene blir liggende i omvendt rekkefølge. Dersom f.eks. verdiene i `x` i utgangspunktet er sortert i stigende rekkefølge (`x[0] <= x[1] <= .. <= x[999]`) så skal altså metoden endre rekkefølgen slik at de etterpå er sortert i synkende rekkefølge. Du skal ikke deklarere noen flere arrayer for å løse oppgaven, men du kan gjerne deklarere noen enkle variabler hvis du ønsker det.

Svar:

```

void reverser (double[] x) {

    int n = x.length;

    for (int i = 0; i < n / 2; i++) {
        double tmp = x[i];
        x[i] = x[n - 1 - i];
        x[n - 1 - i] = tmp;
    }

    /* Merk 1: det er ikke nødvendig å behandle tilfellene x.length == oddetall
    og x.length == partall hver for seg (forsøk f.eks. å la x.length == 2 og
    deretter x.length == 3 i programkoden over, og overbevis deg om at
    det riktige skjer i begge tilfeller).

    Merk 2: ikke deklarer variable før du trenger dem. I programkoden
    over deklarerer double-variabelen tmp INNE i løkka, ikke før løkka.
    Dermed eksisterer variabelen tmp kun i den delen av programmet
    hvor variabelen brukes. Dette gjør gjenbruk lettere, reduserer sjansen
    for programmeringsfeil og er (i motsetning til hva mange tror) minst like
    effektivt som å deklarere tmp før løkka. Vil du vite mer om slike ting,
    se boka Effective Programming in Java,
    http://java.sun.com/developer/Books/effectivejava/

    */

}

```

Oppgave 10 (Vanskelig)

Skriv metoden under som finner og skriver ut på skjerm de to største verdiene i arrayen `v`. En verdi kan forekomme flere steder i arrayen, men du skal anta at ikke alle verdier i arrayen er like. Hvis f.eks. arrayen inneholder to forekomster av verdien 100, to forekomster av verdien 50, og resten av verdiene er 0, så skal metoden skrive ut at største verdi var 100 og nest største verdi var 50. Du trenger ikke å skrive ut hvor disse verdiene ligger i arrayen. Du kan anta lengden av arrayen er minst 2 og at den bare inneholder ikke-negative heltall.

```
void finnToStørste (int[] v) {

    int største = 0;

    for (int i=0; i<v.length; i++) {
        if (v[i] > største) {
            største = v[i];
        }
    }

    System.out.println("Største verdi er " + største);

    int neststørste = 0;

    for (int i=0; i<v.length; i++) {
        if (v[i] > neststørste && v[i] < største) {
            neststørste = v[i];
        }
    }

    System.out.println("Nest største verdi er " + neststørste);
}
```

/ Merk: det er fullt mulig å finne både største og nest største verdi i en enkelt løkke, men det krever litt mer omtanke. De fleste som forsøkte seg på dette på eksamen endte opp med å gjøre feil. De som fikk det til, fikk samme uttelling som de som valgte løsningen ovenfor. Oppgave: forklar hvorfor forslaget nedenfor IKKE alltid gir riktig løsning (ser du hvordan en ekstra setning får den til å virke?):*

```
int største = 0, neststørste = 0;
for (int i=0; i<v.length; i++) {
    if (v[i] >= største) {
        største = v[i];
    } else if (v[i] > neststørste) {
        neststørste = v[i];
    }
}
*/
```

Oppgave 11

I programmet nedenfor skal du lage en konstruktør til klassen Sirkel som har sirkelens radius som parameter. Du skal også lage en objektmetode i klassen Sirkel som regner ut arealet og returnerer denne verdien (du kan bruke formelen: $\text{areal} = 3.14 * \text{radius} * \text{radius}$). Du skal også skrive programkode i main som oppretter to sirkler, en med radius 10.4 og en med radius 244.99. Fra main skal du så kalle på arealberegningemetoden i hvert av de to objektene og skrive ut en linje for hver sirkel med radius og areal.

```
class Sirkel {
    double radius;

    // skriv konstruktør her

    Sirkel (double radius) {
        this.radius = radius;
    }

    // skriv objektmetode her som beregner og returnerer arealet
    double finnAreal() {
        return 3.14 * radius * radius;
    }

    // Vanlig feil: ha med radius som parameter i metoden over. Sirkelens radius
    // er tilgjengelig som en objektvariabel, og det er denne som skal benyttes.
} // end class Sirkel

class SirkelTest {

    public static void main ( String [] args) {
        // skriv kode her som lager to Sirkel-objekter og
        // skriver ut deres radius og areal som gitt ovenfor

        // Først lager vi to sirkel-objekter med radier som oppgitt i oppgaven. Merk at sirkel-
        // radiusen skal gis som argument når man oppretter objektet, og ikke settes inn
        // etterpå med s1.radius = ..... Glemmer man å gi sirkel-radiusen som argument
        // får man kompileringfeil.

        double radius1 = 10.4;
        double radius2 = 244.99;
        Sirkel s1 = new Sirkel (radius1);
        Sirkel s2 = new Sirkel (radius2);

        // Så finner vi arealet til sirklene. Nå skal vi ikke oppgi sirklenes radier,
        // siden disse allerede ligger inni Sirkel-objektene.

        double areal1 = s1.finnAreal();
        double areal2 = s2.finnAreal();

        // Til slutt skriver vi ut radius og areal for hver av sirklene.

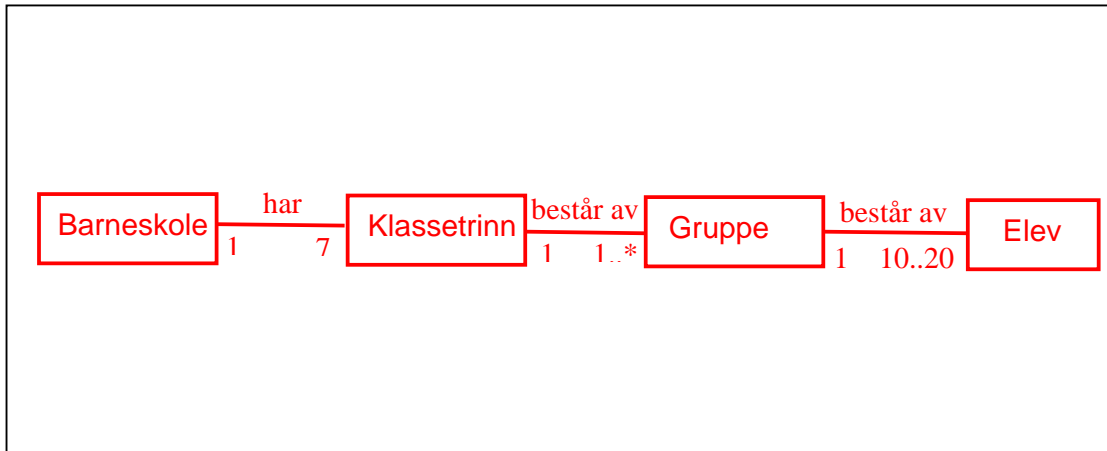
        System.out.println("Sirkel 1 har radius " + radius1 + " og areal " + areal1);
        System.out.println("Sirkel 2 har radius " + radius2 + " og areal " + areal2);

    } // end main
} // end class SirkelTest
```

Oppgave 12

På en barneskole finnes det 7 klassetrinn som hver igjen består av en eller flere grupper. Hver gruppe består så av fra 10 til 20 elever. Tegn et UML-klassediagram med de 4 (Java-)klassene som kan brukes til å representere dette problemet. Gi navn på disse klassene og plasser antall på forholdet mellom disse klassene.

Svar:



Oppgave 13

Anta at følgende program utføres:

```

class Tre {
    Tre venstre, høyre;
    static int ant = 0;

    Tre (int x) {
        ant++;
        if (x > 10 ) venstre = new Tre(x-10);
        if (x > 20 ) høyre = new Tre(x-20);
    }

    public static void main(String[] args) {
        ant = 0;
        Tre t1 = new Tre(11);
        System.out.println("1 " + ant);

        ant = 0;
        Tre t2 = new Tre(21);
        System.out.println("2 " + ant);

        ant = 0;
        Tre t3 = new Tre(31);
        System.out.println("3 " + ant);
    }
}
  
```

Tegn først opp for deg selv og finn ut hvor mange Tre-objekter vi får laget av de ulike setningene i main. Svar så på spørsmålet: hva skriver programmet ut på skjermen?

- 1) 2
- 2) 4
- 3) 7

Svar:

Oppgave 14

Anta at du har en tekstfil **Bok.txt** som f.eks. inneholder teksten i en roman. Lag et komplett, kompilerbart program som leser filen ord for ord og som teller opp antall forekomster av hvert enkelt ord i filen. Programmet ditt skal deretter opprette en ny fil **Ordforekomster.txt** og skrive resultatet av analysen til denne filen. Følgende opplysninger om **Bok.txt** skal skrives til denne filen: (a) antall ord totalt; (b) antall ulike ord; og (c) en liste over alle ord som forekommer minst 6 ganger. Hver linje av denne listen skal inneholde ett ord og antall forekomster av dette ordet, atskilt av blanke tegn.

Du kan anta at filen **Bok.txt** kun inneholder bokstaver og blanke tegn, dvs spesialtegn som komma, punktum, kolon, osv, forekommer ikke i teksten. Du kan også anta at alle ord i teksten er skrevet med små bokstaver, slik at du slipper problemer med stor forbokstav i begynnelsen av en setning, osv.

Hint: bruk en HashMap til å holde rede på ordene du leser i **Bok.txt** og antall forekomster av dem. Tenk deg nøye om hva du skal legge inn i HashMap'en. Husk at både nøkkelen og verdien du legger inn i en HashMap må være (pekere til) objekter. Du vil kanskje ønske å definere en klasse Ord hvor hvert objekt lagrer et ord og antall forekomster av ordet.

Svar:

```
import easyIO.*;           // For å få tilgang til klassene In og Out
import java.util.*;       // For å få tilgang til klassen HashMap

class BokMain {
    public static void main (String[] args) {
        BokAnalyse ba = new BokAnalyse();
        ba.lesFraFil("Bok.txt");
        ba.skrivTilFil("Ordforekomster.txt");
    }
}

class BokAnalyse {
    private HashMap alleOrd = new HashMap();
    private int antallOrdTotalt = 0;

    public void lesFraFil (String filnavn) {
        // Se neste side
    }

    public void skrivTilFil (String filnavn) {
        // Se neste side
    }
}

class Forekomster {
    private int antall = 1;

    public void nyForekomst() {
        antall++;
    }

    public int fåAntall() {
        return antall;
    }
}
```

(fortsetter på neste side)

```
void lesFraFil (String fnavn) {
    In innfil = new In(fnavn);

    while (!innfil.lastItem()) {
        String ord = innfil.inWord(); // Deklarer variabelen ord inni løkka, ikke før
        antallOrdTotalt++;

        if (alleOrd.containsKey(ord)) {
            Forekomster fo = (Forekomster) alleOrd.get(ord);
            fo.nyForekomst();
        } else {
            Forekomster fo = new Forekomster();
            alleOrd.put(ord, fo);
        }
    }
}

public void skrivTilFil (String filnavn) {
    Out utfil = new Out(filnavn);

    utfil.outln("Antall ord totalt: " + antallOrdTotalt);
    utfil.outln("Antall ulike ord: " + alleOrd.size());

    Iterator it = alleOrd.keySet().iterator();

    while (it.hasNext()) {
        String ord = (String) it.next();
        Forekomster fo = (Forekomster) alleOrd.get(ord);

        int antall = fo.fåAntall();
        if (antall > 5) {
            utfil.outln(ord + "\t" + antall);
        }
    }

    utfil.close();
}
```