

Inf 1000 høst 2005

Løsningsforslag ordinær eksamen

Oppgave 1 (10 poeng)

- a) $11 \times 3 = 33$
- b) 4
- c) 4
- d) $10 \times 9 = 90$

Oppgave 2 (16 poeng)

Er disse programsetningene lovlige i Java?

- | JA | NEI |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> <code>int i=1, j =i;</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>int[] x = new int[];</code> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> <code>int[] x = new int[0];</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>int i = new int [7];</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>double int = 14;</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>double[] x = new double x[5];</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <code>long [] x = new int[33];</code> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> <code>long [] a, b = new long[4];</code> |

Oppgave 3 (5 poeng)

Forskjellige

Oppgave 4 (15 poeng)

```
double matPrisMedMoms(double matPrisUtenMoms, boolean servert) {
    if( servert ) return matPrisUtenMoms*1.11;
    else return matPrisUtenMoms*1.25;
}
```

Oppgave 5 (25 poeng)

- | JA | NEI |
|--------------------------|---|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <pre>int k = 0; while (k < c.length) { snitt = c[k]/(double)c.length; }</pre> |

- x `int k = 0;
snitt = c[0];
while (k < c.length) {
 c[k++] += snitt;
}
snitt = snitt/c.length;`
- x `int k = 0;
while (k++ < c.length) {
 snitt = snitt + c[k-1];
}
snitt = snitt/(k-1);`
- x `int k = 0;
while (k < c.length) {
 snitt += c[k++]/c.length; // feil:int-divisjon
}`
- x `int k = 0;
while (k++ < c.length) {
 snitt += c[k-1]/(double)c.length;
}`

Oppgave 6 (15 poeng)

lnx = 10.

Oppgave 7 (15 poeng)

```
int størst(int a, int b, int c) {
    int maxab = a;
    if( a<b ) maxab = b;
    if( c>maxab ) return c;
    else return maxab;
}
```

Oppgave 8 (25 poeng)

```
String [] lagNyeNavn(String [] navn) {
    int fornavn=0, etternavn=1;
    String[] nye = new String[2];
    String[] første = navn[0].split(" ");
    String[] andre = navn[1].split(" ");
    String famnavn = andre[etternavn] + "-" + første[etternavn];
    nye[0] = første[fornavn] + " " + famnavn;
    nye[1] = andre[fornavn] + " " + famnavn;
    return nye;
}
```

Oppgave 9 (30 poeng)

```
class Karakter {
    int []karakterAntall = new int[6];

    Karakter (String fil) {
        In innfil = new In(fil);
        for(int i=0; i<6; i++) karakterAntall[i] = 0;
        while( !innfil.endOfFile() ){
            innfil.inInt(" ");
            char kar = innfil.inChar(" ");
            innfil.readLine();
            karakterAntall[(int)(kar-'A')]++;
        }
    }

    void prosentKarakterfordelig () {
        double total = 0;
        for(int i=0; i<6; i++)
            total += karakterAntall[i];
        System.out.println("Totalt sensurantall: "+ total);
        int stryk = karakterAntall[5];
        double bestått = total - stryk;
        System.out.println("Stryk: " + (stryk/total)*100 +" %");
        for(int i=0; i<6; i++)
            System.out.println((char)('A'+i) + ": "+
                (karakterAntall[i]/bestått)*100 );
    }
}
```

Oppgave 10 (30 poeng)

```
void sorter (int[] a) {
    for(int i=0; i<a.length-1; i++){
        int min = i+1;
        for(int j=min+1; j<a.length; j++)
            if( a[j] < a[min] ) min = j;
        if( a[min] < a[i] ){
            int temp = a[i]; a[i] = a[min]; a[min] = temp;
        }
    }
}
```

Oppgave 11 (25 poeng)

```
class PyramideTest {

    public static void main ( String [] args) {
        // skriv kode her som lager to Pyramide-objekter og
```

```

        // skriver ut deres høyde, lengde og volum
        Pyramide[] p = new Pyramide[2];
        p[0] = new Pyramide(30.1,30.1,22.9);
        System.out.println("Pyramide "+ 30.1 +" x "+ 30.1+" x "+22.9+"
            har volum "+ p[0].volum());
        p[1] = new Pyramide(30.1,30.1,2*22.9);
        System.out.println("Pyramide "+ 30.1 +" x "+ 30.1+" x "+2*22.9+"
            har volum "+ p[1].volum());
    } // end main
} // end class PyramideTest

```

```

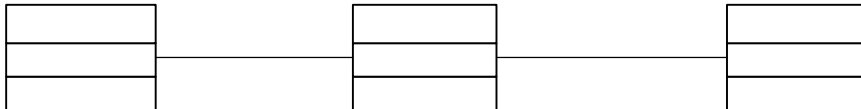
class Pyramide {
    double høyde, lengde, bredde;

    // skriv konstruktør her
    Pyramide( double lengde, double bredde, double høyde ){
        this.lengde = lengde;
        this.bredde = bredde;
        this.høyde = høyde;
    }

    // skriv objektmetode her som beregner og returnerer volumet
    double volum(){
        return 0.333*høyde*bredde*lengde;
    }
} // end class Pyramide

```

Oppgave 12 (20 poeng)



Oppgave 13 (25 poeng)

Det genereres rekursivt en trestruktur med 7 objekter og dybde 4. Utskriften følger en standard postfiks traversering av treet som (uten linjeskiftene) gir:

```

Antall:0
Antall:1
Antall:0
Antall:2
Antall:0
Antall:1
Antall:3

```

Oppgave 14 (50 poeng)

```

import java.util.*;
import easyIO.*;

```

```

class Salg{

```

```

String varenavn;
int antall;
double pris;

Salg(String v, int a, double p){
    varenavn=v; antall=a; pris=p;
}

void oppdater(int n){
    antall += n;
}

double omsatt(){
    return antall*pris;
}
}

class SalgSummator {

    In innfil = new In("Salg.txt");
    Out utfil = new Out("AntallSolgt.txt");
    HashMap<String,Salg> arkiv = new HashMap<String,Salg>();
    Salg mestSolgt = new Salg("DUMMY",0,0);
    Salg mestOmsatt= mestSolgt;

    public static void main(String[] args) {
        SalgSummator ordensmann = new SalgSummator();
    }

    SalgSummator(){
        lesFil();
        beregn();
        skrivFil();
    }

    void lesFil(){
        while( !innfil.endOfFile() ){
            String varenavn = innfil.inWord(" ");
            int antall = innfil.inInt(" ");
            double pris = innfil.inDouble();
            innfil.readLine();
            Salg aktivPost = arkiv.get(varenavn);
            if( aktivPost == null )
                arkiv.put( varenavn, new Salg(varenavn,antall,pris) );
            else aktivPost.oppdater(antall);
        }
    }

    void beregn(){
        double sum = 0;
        for( Salg s: arkiv.values() ){
            double omsatt = s.omsatt();
            sum += omsatt;
            if( s.antall > mestSolgt.antall )
                mestSolgt = s;
            if( s.omsatt() > mestOmsatt.omsatt() )

```

```
        mestOmsatt = s;
    }
    System.out.println("Totalt salg: "+ sum +" kr.");
    System.out.println("Mest solgt: "+ mestSolgt.varenavn +" (" +
        mestSolgt.antall +" )" );
    System.out.println("Mest omsatt: "+ mestOmsatt.varenavn +" (" +
        mestSolgt.omsatt() +" kr)" );
}

void skrivFil(){
    for( Salg s: arkiv.values() )
        utfil.outln(s.varenavn+" "+ s.antall);
    utfil.close();
}
}
```