

Løsningsforslag
INF1000
3 desember 2007

Oppgave 1

Svar: 50

```
// Løsning til Oppgave 2

class Oppgave2 {
    public static void main(String[] args) {
        // Lovlig, men ikke med i oppgaven
        int i[], j;

        // Her er oppgaven:

        // Lovlig
        float x = 3 / 2;

        // Lovlig
        int x = 3 / 2;

        // Ikke lovlig
        int[] boolean[] s = new int[8]boolean[9];

        // Ikke lovlig
        int i =1, j=i+3, while( j-->0) i = j;

        // Ikke lovlig
        long[] boolean = new boolean[12];

        // Lovlig
        boolean[] x = new boolean[0];

        // Lovlig
        int XX = (int) (3.1415);

        // Lovlig
        double x = (int) 33.5;

        // Ikke lovlig
        String[3] a = {"1", "2", "3"};

        // Ikke lovlig
        int[][] a = new int[][10];
    }
}
```

```
class Oppgave3{
    public static void main(String args[]){

        // Denne: 5 ganger
        for (int i=1; i<6; i++) {
            System.out.println("INF1000");
        }
        System.out.println("* * * * *");

        // Denne: 8 ganger
        {
            int k = 1;
            while (++k < 10)
                System.out.println("INF1000");
        }
        System.out.println("* * * * *");

        // Denne: 0 ganger
        {
            for (int j=1; j < 10 ; j = j+1 ) {
                for (int i = j; ++i < j+1; j = i) {
                    System.out.println("INF1000");
                }
            }
        }
        System.out.println("* * * * *");

        // Denne: 6 ganger
        {
            int k = 1;
            while (++k +1 < 9 )
                System.out.println("INF1000");
        }
    }
}
```

```
class Oppgave4{
    public static void main(String args[]){
        int en = 1, to=2, tre=3;
        int fire = en++ + ++to - --tre;
        int svar = fire + 3*tre + 2*to + 1*en;
        System.out.println("svar=" + svar);

        // svar=16
    }
}
```

```
class Oppgave5{
    double kulevolum( double radius ){
        // Husk: Passe på at det blir flyttall.
        // Ellers, at det er riktig formel, ^3, osv.
        return 4.0/3.0 * 3.1415926 * radius * radius * radius;
    }

    public static void main(String args[]){
        double test1 = 1.0;
        double test2 = 2.0;
        Oppgave5 o = new Oppgave5();
        System.out.println("test1=>" + o.kulevolum(test1));
        System.out.println("test2=>" + o.kulevolum(test2));
    }
}
```

```
class Oppgave6{
    double gjennomsnittOljePrisINorskeKroner( double [] oljepris ){
        double usd = 0.0;
        int funnet=0;
        for(int i=0;i<oljepris.length;i++){
            // Ikke ta med de som er -1.0
            if(oljepris[i]>-0.5){
                usd+=oljepris[i];
                funnet++;
            }
        }
        // Huske på å dele på riktig antall
        return usd*5.28/funnet;
    }

    public static void main(String args[]){
        double[] januar = new double[]{70.0, 80.0, -1.0};
        double[] februar = new double[]{100.0, -1.0, -1.0};
        Oppgave6 o = new Oppgave6();
        System.out.println("januar=>" +
            o.gjennomsnittOljePrisINorskeKroner(januar));
        System.out.println("februar=>" + o.gjennomsnittOljePrisINorskeKroner(
            februar));
    }
}
```

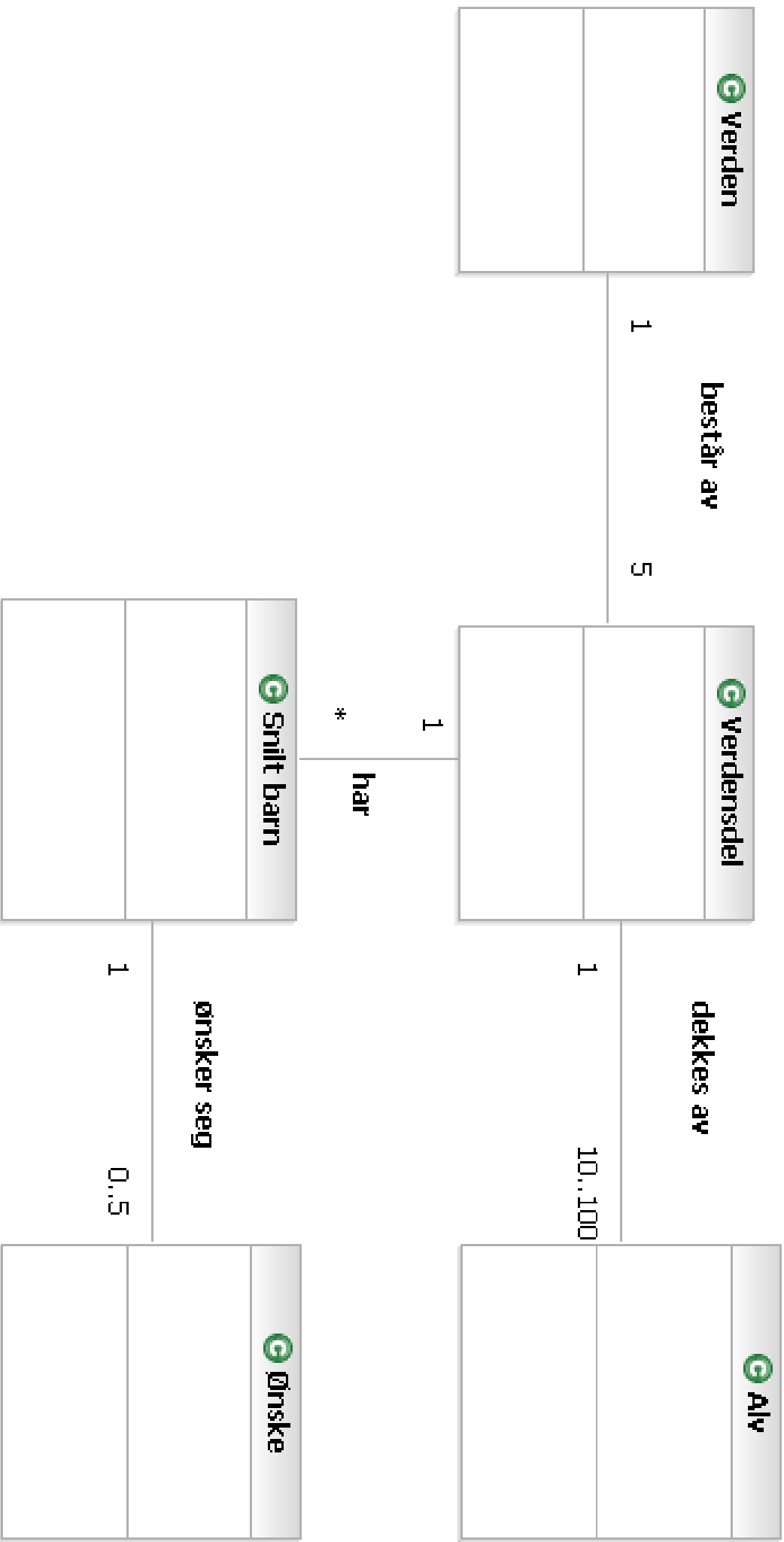
```
class Oppgave7{
    public static void main(String args[]){
        String s = "nitternittini";
        String t = s.substring(6,13);
        int pos = t.indexOf("ni");
        System.out.println("t=" + t);
        System.out.println("pos=" + pos);

        // t=nittini
        // pos=0
    }
}
```



```
class Oppgave8{
    int rest(int n, int m){
        int k = 0;
        while(n >= m) {
            n-=m;
        }
        while ( n-- > 0) {
            k++;
        }
        return k;
    }
    public static void main(String args[]){
        Oppgave8 o = new Oppgave8();
        System.out.println("rest(41,7)=>" + o.rest(41,7));

        // rest(41,7) => 6
    }
}
```



```

import easyIO.*;
import java.util.*;
//import easyEpost.*;

class Eksamenssystem {
    public static void main(String [] args) {
        SendEposter s = new SendEposter("AlleEksamene.txt");
        s.sendMeldinger();
    }
} // end class Eksamenssystem

class SendEposter {
    HashMap <String,Student> studenter = new HashMap<String, Student>();

    SendEposter (String eksamensfil) {
        In les = new In(eksamensfil);
        while(les.hasNext()){
            String filnavn = les.inWord();
            lesResultater(filnavn);
        }
    } // end constructor

    void lesResultater(String filnavn){
        In les = new In(filnavn);
        String emne = filnavn.substring(0, filnavn.length()-3);
        while(les.hasNext()){
            String brukernavn = les.inWord();
            String resultat = les.inWord();
            Student stud = studenter.get(brukernavn);
            if(stud == null){
                stud = new Student(brukernavn);
                studenter.put(brukernavn, stud);
            }
            stud.leggTilResultat(emne, resultat);
        }
    } // end lesResultater

    void sendMeldinger(){
        Epost epost = new Epost();
        Iterator<Student> it = this.studenter.values().iterator();
        while(it.hasNext()){
            Student student = it.next();
            String tittel = "Dine "+student.ant+" eksamensresultater dette semesteret.";
            String melding = "Hei, " + student.brukernavn + "\n";
            melding += student.hentResultater();
            melding += "Hilsen eksamenssystemet ved Universitetet i Tlön.";
            if( ! epost.sendEnEPost(student.brukernavn, tittel, melding)){
                System.out.println("Fant ikke studenten " + student.brukernavn + ". Epost ikke sendt!");
            }
        }
    } //end sendMeldinger
} // end class SendEposter

class Student{
    String[] emner= new String[5];
    String[] resultater=new String[5];
    String brukernavn;
    int ant=0;

    Student(String brukernavn){
        this.brukernavn = brukernavn;
    }

    void leggTilResultat(String emne, String resultat){
        this.emner[ant] = emne;
    }
}

```

```
        this.resultater[ant] = resultat;
        this.ant++;
    }

    String hentResultater() {
        String res = "";
        for(int i=0; i<ant; i++){
            res += " " + resultater[i] + " i emnet " + emner[i] + "\n";
        }
        return res;
    }
} // end class Student

/*
 *
 * Studentene skal ikke skrive det som står herfra og ned
 *
 */
class Epost{
    boolean sendEnEPost(String brukernavn, String tittel, String melding) {
        System.out.println("Sender epost");
        System.out.println("*****");
        System.out.println("Til: " + brukernavn);
        System.out.println("Tittel: " + tittel);
        System.out.println("Melding: \n" + melding);
        System.out.println("*****");
        return true;
    }
}
```

```

import easyIO.*;
import java.util.*;
//import easyEpost.*;

public class EksamenssystemAm{
    public static void main(String [] args) {
        SendEposter s = new SendEposter("AlleEksamene.txt");
        s.sendMeldinger();
    } // end class EksamenssystemAm

class SendEposter {
    HashMap <String,Student> studenter = new HashMap<String, Student>();

    SendEposter (String eksamensfil) {
        In les = new In(eksamensfil);
        while(les.hasNext()){
            String filnavn = les.inWord();
            lesResultater(filnavn);
        } // end constructor

    void lesResultater(String filnavn){
        In les = new In(filnavn + ".res");

        while(les.hasNext()){
            String brukernavn = les.inWord();
            String resultat = les.inWord();
            String res = " "+resultat+" i "+ filnavn +"\n";
            Student stud= studenter.get(brukernavn);
            if(stud == null){
                stud = new Student(brukernavn);
                studenter.put(brukernavn, stud);
            }
            stud.resultater +=res;
            stud.ant++;
        } // end lesResultater

    void sendMeldinger(){
        Epost epost = new Epost();
        Iterator it = this.studenter.values().iterator();
        while(it.hasNext()){
            Student student = (Student) it.next();
            String tittel = "Dine "+student.ant+" eksamensresultater dette semesteret.";
            String melding = "Hei, " + student.brukernavn + "\n";
            melding += student.resultater;
            melding += "Hilsen eksamenssystemet ved Universitetet i Tlön.";
            if (! epost.sendEnEPost(student.brukernavn, tittel, melding)) {
                System.out.println("FEIL: student:" + student.brukernavn + " finnes ikke");
            }
        }
    } //end sendMeldinger
} // end class SendEposter

class Student{
    String resultater="";
    String brukernavn;
    int ant=0;

    Student(String brukernavn){
        this.brukernavn = brukernavn;
    }
} // end class Student

```

```
/*
 *
 * Studentene skal ikke skrive det som står herfra og ned
 *
 */
class Epost{
    boolean sendEnEPost(String brukernavn,String tittel,String melding) {
        System.out.println("Sender epost");
        System.out.println("*****");
        System.out.println("Til: " + brukernavn);
        System.out.println("Tittel: " + tittel);
        System.out.println("Melding: \n" + melding);
        System.out.println("*****");
        return true;
    }
}
```

```

class Tre {
    Tre venstre;
    Tre hoyre;
    String tekst;
    Tre(String tekst){
        this.tekst = tekst;
    }
    Tre(String tekst, Tre venstre, Tre hoyre){
        this.tekst = tekst;
        this.venstre = venstre;
        this.hoyre = hoyre;
    }
    void skrivUt(){
        if(this.venstre != null)
            this.venstre.skrivUt();
        System.out.print(" " + this.tekst + " ");
        if(this.hoyre != null)
            this.hoyre.skrivUt();
    }
    public static void main(String[] args) {
        Tre a = new Tre("A");
        Tre b = new Tre("B");
        Tre c = new Tre("C", a, b);

        Tre d = new Tre("D");
        Tre e = new Tre ("E", c, d);
        c.skrivUt();
        System.out.print(" *** ");
        e.skrivUt();

        // Resultat:
        // A C B *** A C B E D

        // Tre:
        //      E
        //     / \
        //    C   D
        //   / \
        //  A  B

        // Bare en tom linje
        System.out.println();
    }
}

```

```
class Sortering{
    int finnMinIndeks (int[] a, int start, int end) {
        int min = a[start];
        int indeks = start;
        for(int i=start;i<=end;i++){
            // Både oppdatere indeks og min.
            if(a[i] < min){
                indeks=i;
                min=a[i];
            }
        }
        // Ikke glemme return
        return indeks;
    }
    void sorter (int[] a){
        for(int i=0;i<a.length;i++){
            int minI = finnMinIndeks(a, i, a.length-1);
            int min = a[minI];
            a[minI] = a[i];
            a[i] = min;
        }
        // Ingen return eller print.
    }
    public static void main(String [] args) {
        int[] test = new int [] {3,5,7,12,54,12,1,1,9,34,54,23,1,68,92,3,5,78,0,1};
        for(int i=0;i<test.length;i++){
            System.out.print(test[i] + " ");
        }
        System.out.println();
        Sortering s = new Sortering();
        s.sorter(test);
        for(int i=0;i<test.length;i++){
            System.out.print(test[i] + " ");
        }
        System.out.println();
    }
}
```


Løsning Oppgave 13

Systemet de foreslår rammes **ikke** av §2 – dette r ikke sensitiv informasjon, og Univeristetet i Tlön har klart lov etter §8 (a) Oppfylle avtale med den registrerte, c) Ivareta den registrertes interesser, og e) å utøve offentlig myndighet) til å samle inn disse data for eksamensformål.

Den desidert mest relevante paragrafen her er §11 *Grunnkrav til behandling av personopplysninger*, og punkt b) og særlig pkt. c):

- b) bare nyttes til uttrykkelig angitte formål som er saklig begrunnet i den behandlingsansvarliges virksomhet,
- c) ikke brukes senere til formål som er uforenlig med det opprinnelige formålet med innsamlingen, uten at den registrerte samtykker,

Det er rimelig å si at uten at Universitetet søker og får tillatelse av Datatilsynet og dette også offentliggjøres for studenter som tar kurs på Universitetet, så bryter dette foreslåtte systemet punktene b) og c) i §11 og er derfor ikke lovlig.

Hvis man vil hevde at dette tenkte systemet er lovlig bør man vise til §8 e) om å utøve offentlig myndighet, men det er nok en noe svakere besvarelse.