

Kandidatnummer: \_\_\_\_\_

BOKMÅL

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet – eksamen

Eksamen i : INF1000 — Grunnkurs i objektorientert programmering  
Eksamensdag : Mandag 3. desember 2007  
Tid for eksamen : 09 – 12  
Oppgavesettet er på : 12 sider  
Vedlegg : Ingen  
Tillatte hjelpemidler : Alle trykte og skrevne

- Les *nøye* gjennom hver oppgave før du løser den. For hver oppgave er angitt det maksimale antall poeng du kan få hvis du svarer helt riktig. Summen av poengene er 252, slik at f.eks 4 poeng tilsvarer omlag 3 minutter og 10 poeng omlag 7 min. (hvis du regner med å komme igjennom alt). Pass på at du bruker tiden din riktig.
- Kontroller også 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 så fall rede for forutsetningene og antagelsene du gjør.
- Dine svar *skal* skrives på disse oppgavearkene, og *ikke* på separate ark. Dette gjelder både spørsmål med avkrysnings svar 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.

### Oppgave 1 (2 poeng)

a) Hvor mange int-verdier er det plass til i arrayen 'hester'?

```
int[][] hester = new int[10][5];
```

Svar: .....

## Oppgave 2 (20 poeng)

Er disse programsetningene lovlige i Java?

- | JA                       | NEI  |
|--------------------------|--|
| <input type="checkbox"/> | <input type="checkbox"/> <code>float x = 3 / 2;</code>                         |
| <input type="checkbox"/> | <input type="checkbox"/> <code>int x = 3 / 2;</code>                           |
| <input type="checkbox"/> | <input type="checkbox"/> <code>int[]boolean[] s = new int[8]boolean[9];</code> |
| <input type="checkbox"/> | <input type="checkbox"/> <code>int i =1, j=i+3, while( j--&gt;0) i = j;</code> |
| <input type="checkbox"/> | <input type="checkbox"/> <code>long[] boolean = new boolean[12];</code>        |
| <input type="checkbox"/> | <input type="checkbox"/> <code>boolean[] x = new boolean[0];</code>            |
| <input type="checkbox"/> | <input type="checkbox"/> <code>int XX = (int) (3.1415);</code>                 |
| <input type="checkbox"/> | <input type="checkbox"/> <code>double x = (int) 33.5;</code>                   |
| <input type="checkbox"/> | <input type="checkbox"/> <code>String[3] a = {"1", "2", "3"};</code>           |
| <input type="checkbox"/> | <input type="checkbox"/> <code>int[][] a = new int[][10];</code>               |

## Oppgave 3 (16 poeng)

a) Hvor mange ganger blir "INF1000" skrevet ut av følgende løkke:

```
for (int i=1; i<6; i++) {  
    System.out.println("INF1000");  
}
```

Svar: ..... ganger

b) Hvor mange ganger blir "INF1000" skrevet ut av følgende løkke:

```
int k = 1;  
while (++k < 10)  
    System.out.println("INF1000 ");
```

Svar: ..... ganger

c) Hvor mange ganger blir "INF1000" skrevet ut av følgende løkker:

```
for (int j=1; j < 10 ; j = j+1 ) {  
    for (int i = j; ++i < j+1; j = i) {  
        System.out.println("INF1000");  
    }  
}
```

Svar: ..... ganger

d) Hvor mange ganger blir "INF1000" skrevet ut av følgende løkke:

```
int k = 1;  
while (++k +1 < 9 )  
    System.out.println("INF1000");
```

Svar: ..... ganger

#### Oppgave 4 (4 poeng)

Anta at følgende kodelinjer utføres:

```
int en = 1, to=2, tre=3;
int fire = en++ + ++to - --tre;
int svar = fire + 3*tre + 2*to + 1*en;
```

Hva er verdien til variabelen `sva`r rett etterpå?

Svar: .....

#### Oppgave 5 (20 poeng)

Skriv ferdig metoden under, som beregner og returnerer volumet  $V$  av en kule ut fra radius  $r$  i kula og  $\pi$ . Formelen du skal bruke er:  $V = \frac{4}{3}\pi r^3$ , og hvor  $\pi = 3.1415926$  og  $r$  er radius i kula. (*Hint*: Pass på at du ikke tar feil av heltalls- og flyttallsdivisjon.)

Svar:

```
double kulevolum( double radius ){
}
}
```

#### Oppgave 6 ( 20 poeng)

Skriv ferdig metoden under, som med utgangspunkt i en array med priser for et fat med olje for flere av dagene i en måned, finner gjennomsnittsprisen den måneden og returnerer denne gjennomsnittsprisen. Problemet er at for noen av dagene mangler det priser og da står det -1.0 på vedkommende plass i stedet for en virkelig pris (som vi antar alltid er et positivt tall). Gjennomsnittsprisen er definert som summen av alle virkelige priser delt på antallet slike priser som er oppgitt i arrayen. Prisene i arrayen er gitt som flyttall i US dollar, og svaret ditt om gjennomsnittsprisen skal være i norske kroner. Du skal regne med en dollarkurs på kr. 5,28 (dvs. 1 US dollar er verdt kr. 5,28).



## Oppgave 8 (15 poeng)

```
int rest(int n, int m){
    int k = 0;
    while(n >= m) {
        n-=m;
    }
    while ( n-- > 0) {
        k++;
    }
    return k;
}
```

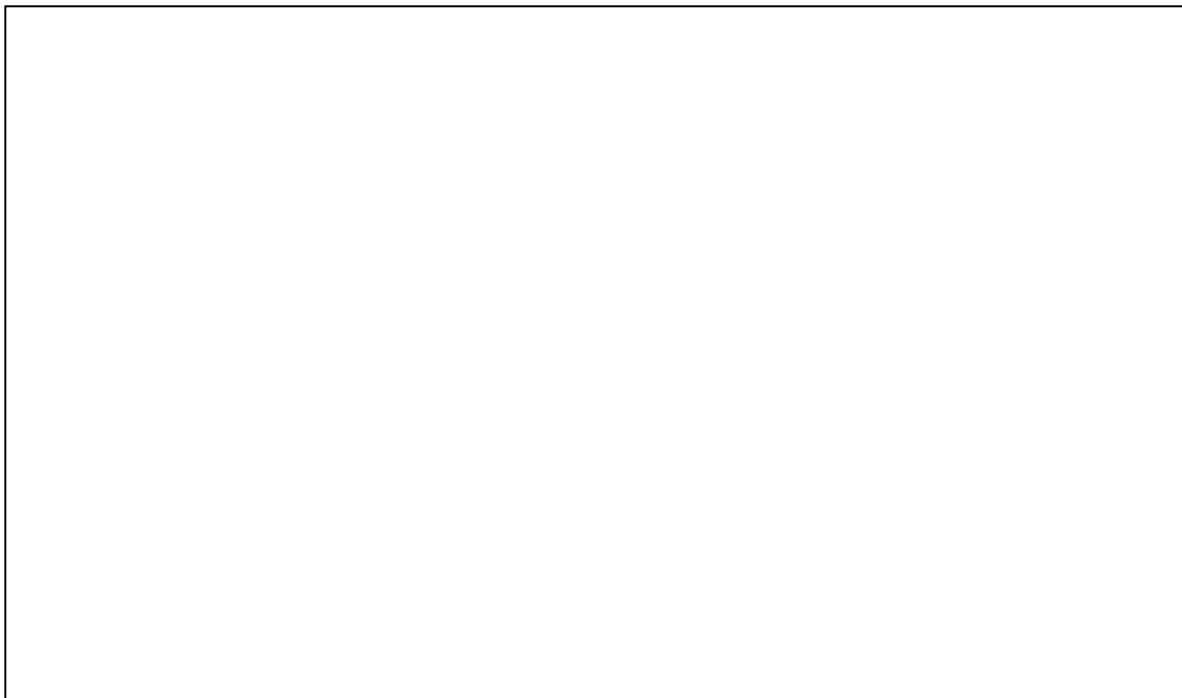
Hva returneres fra metodekallet `rest(41,7)`?

Svar: .....

## Oppgave 9 (20 poeng)

Det nærmer seg jul og Julenissen har fått det veldig travelt. Han har også mistet helt oversikten over alle verdens barn og deres ønsker. Det er noe han trenger når julekvelden kommer. Alvene har fått i oppdrag å lage et datasystem for Julenissen som holder orden på informasjonen Julenissen trenger Julekvelden. Din oppgave er å hjelpe alvene med å tegne et UML-klassediagram for Julenissens datasystem. Hele verden består av 5 verdensdeler. Julenissen klarer ikke å besøke alle verdensdelene selv (Beklager hvis vi ødelegger julegleden for noen!) så på julekvelden er det fra 10-100 alver som dekker hver verdensdel for Julenissen. I hver verdensdel bor det ingen eller flere snille barn. Et snilt barn behøver ikke ha noen ønsker, men Julenissen oppfyller ikke mer enn 5 ønsker. Det er ikke plass til slemme barn i Julenissens datasystem. Tegn et UML-klassediagram med de 5 (Java-)klassene som kan brukes til å representere dette problemet. Gi navn både på disse klassene og på relasjonene mellom dem og plasser antall på begge sider av forholdet mellom disse klassene.

Svar:



## Oppgave 10 (60 poeng)

Alle studentene ved Universitetet i Tlön i landet Uqbar går opp i fra 1 til 5 eksamner hvert semester, og siden alle studentene har ulike innloggingsnavn på universitetets dataanlegg, registreres alle studentenes eksamensresultater (A-E og F) med dette brukernavnet (eks. Studenten Arno Mår har brukernavnet 'arnom'). Alle resultatene for én eksamen registreres på en fil (også de som får F). Navnet på denne fila er kursnavnet etterfulgt av '.res'(eks. eksamensresultatene for INF1000 blir lagret på filen "**INF1000.res**" for MAT1000 på fila: "**MAT1000.res**", osv). På en slik fil er det en linje per student som gikk opp til eksamen på formatet:

**<brukernavn> <karakter>**

Eks: Hvis Arno Mår fikk E på INF1000 og strøk på MAT1000, vil det være en linje på fila: "**INF1000.res**" som følger:

**arnom E**

og en linje på fila: "**MAT1000.res**" som følger:

**arnom F**

Navnene på alle eksamensfilene for inneværende semester ligger på en egen fil: "**AlleEksamene.txt**" med en linje for hvert filnavn (f.eks **MAT1000.res**).

Oppgaven til det systemet du skal lage er å sende *en eneste e-post* til hver student som har tatt eksamen dette semesteret med *alle* eksamensresultatene til den studenten det semesteret. Da skal Arno Mår motta en e-post med tittel "Dine 2 eksamensresultater dette semesteret" og selve meldinga i e-posten vil være: "Hei arno, du fikk E i INF1000, og du fikk F i MAT1000. Hilsen eksamenssystemet ved Universitetet i Tlön".

Til å hjelpe seg å sende e-post har noen laget pakken 'easyEpost', som bla. inneholder en klasse **Epost** som har en metode (som ikke er static):

```
boolean sendEnEPost(String brukernavn,String tittel,String melding)
```

som opplagt sender en e-post med tittel: **tittel** og innhold/melding: **melding** til: **brukernavn** (metoden legger selv på det lokale tillegget i adressen slik at dette blir en fullstendig e-postadresse: brukernavn@ifi.uitl.uq ). Returverdien fra metoden er om e-postsystemet fant at denne brukeren eksisterte eller ikke. Systemet ditt skal skrive ut en feilmelding på skjermen hvis det får et feilaktig/ikke-eksisterende brukernavn.

Vi setter få krav til løsningen din uten at den minst skal ha to klasser: **Eksamenssystem** og **SendEposter** (men sannsynligvis flere). Programmet går først i en løkke og for hver eksamensfil på "**AlleEksamene.txt**" kaller en metode som leser denne og som for hver student på denne eksamensfila, greier å bevare eksamensresultatet og *hvilken eksamen og hvilken student* dette gjelder. Når så denne metoden er kalt for alle eksamene, skal en annen metode kalles som for hver student som er registret med eksamen dette semesteret, først bygger opp en melding til vekommende og så kaller **sendEnEPost**-metoden i klassen **Epost** for å få sendt denne meldinga.

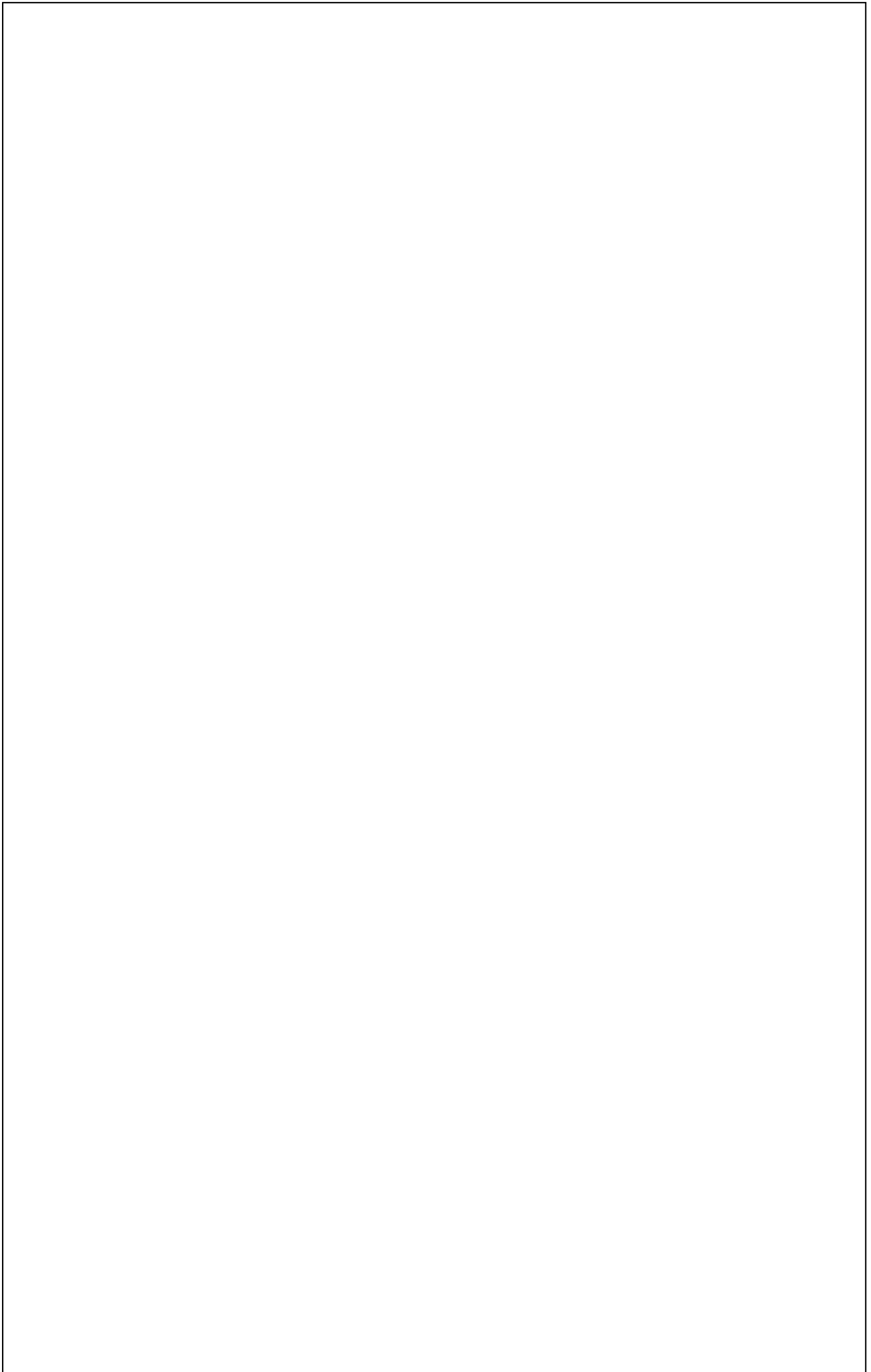
Svar:

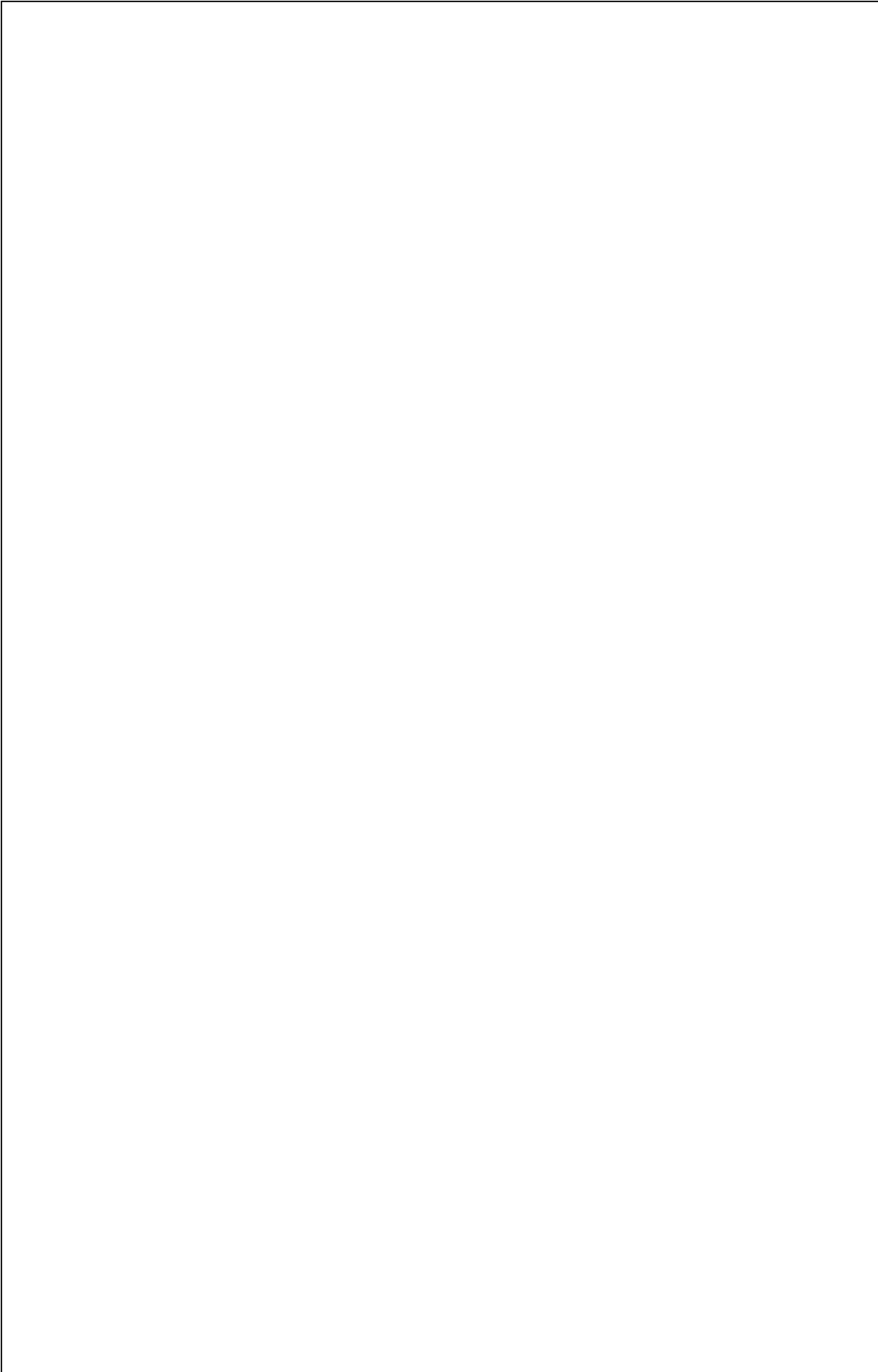
```
import easyIO.*;
import easyEpost.*;
import java.util.*;

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

class SendEposter {

    SendEposter (String eksamensfil) {
```





## Oppgave 11 (15 poeng) Litt vanskelig

Anta at følgende program utføres:

```
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();
    }
}
```

Tegn først opp for deg selv og finn ut hvor mange objekter av klassen **Tre** vi får laget av de ulike setningene i main og hvilke verdier klassevariablene til disse har. Svar så på spørsmålet: hva skriver programmet ut på skjermen?

Svar: .....

## Oppgave 12 (30 poeng)

Du skal skrive to metoder **finnMinIndeks** og **sorter** som tilsammen sorterer en array **a** etter følgende prinsipp: Vi finner først indeksen i arrayen **a** hvor det minste elementet i **a** er. Så lar vi det elementet og elementet i **a[0]** bytte plass. Så finner vi det minste elementet i resten av **a** – dvs i plass 1,2,..., og lar det elementet og **a[1]** bytte plass,...osv til hele arrayen **a** er sortert. Brukeren får da sortert en array ved å kalle på metoden **sorter**.

N.B. Du får ingen poeng for å skrive av sorteringsmetoden i læreboka.

Svar:

```
int finnMinIndeks (int[] a, int start, int end) {
```

```
}
```

```
void sorter (int[] a){
```

```
}
```

### Oppgave 13 (20 poeng)

Landet Uqbar har innført akkurat samme "Lov om behandling av personopplysninger" som i Norge og du skal bruke denne loven til å vurdere følgende:

Eksamenssystemet ved Universitetet i Tlön i Uqbar vil nå bruke de data som ligger på filene i Oppgave 10 til å lage et rapporteringssystem til Lånekassa i Uqbar om hvilke studenter som ikke har god nok studieprogressjon ved **ikke** å stå i minst 3 eller flere eksamener dette semesteret. De vil for hver slik student sende en epost til Lånekassa om hvor få eksamener disse studentene har tatt. Du skal **ikke** skrive kode for det, men derimot skrive en vurdering om det er tillatt for Universitetet i Tlön i Uqbar etter bestemmelsene i " Lov om behandling av personopplysninger" å sende slike beskjeder om studenter som ikke jobber bra nok til Lånekassa, og hvor studentene vil få redusert lån neste semester. Begrunn svaret med å vise til konkrete paragrafer som du mener er relevante, og hvorfor de er det: