

Kandidatnummer: _____

BOKMÅL

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i : INF1000 — Grunnkurs i objektorientert programmering
Eksamensdag : Mandag 1. desember 2008
Tid for eksamen : 14.30-17.30
Oppgavesettet er på : 13 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 218. Oversatt i tidsbruk vil 4 poeng tilsvare omlag 3 minutter og 10 poeng omlag 8 minutter (hvis vi regner at 218 poeng tilsvarer 3 timer). Pass på at du bruker tiden fornuftig.
- 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 `resultater`?

```
int[] [] resultater = new int[9][4];
```

Svar:36.....

Oppgave 2 (20 poeng)

Er disse programsetningene lovlige i Java?

- | JA | NEI | |
|-------------------------------------|-------------------------------------|--|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | <code>int x = 3/2;</code> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | <code>int x = (int)3.0/(int) 2.77;</code> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | <code>int i =4; double x = i*22.55;</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>double i =4; int x = i*22.55;</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>String int = "123";</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>int k = new int [7];</code> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | <code>String [][] XX = {"hei"};</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>double xy = new double(1.0 + 4);</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>while(boolean b == true) b = false;</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>new int[][] a = new int [9][10];</code> |

Oppgave 3 (16 poeng)

a) Hvor mange ganger blir "INF1000" skrevet ut her:

```
for (int i=2; i<9; i++) {  
    System.out.println("INF1000");  
}
```

Svar:7.... ganger

b) Hvor mange ganger blir "INF1000" skrevet ut her:

```
int i=1;  
while (++i < 100) {  
    System.out.println("INF1000 ");  
}
```

Svar:98..... ganger

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

```
for (int j=1; j<5; j=j+1) {  
    for (int i=j; i<j*2; i++) {  
        System.out.println("INF1000");  
    }  
}
```

Svar: ...10..... ganger

d) Hvor mange ganger blir "INF1000" skrevet ut her:

```
int k = 1;
while (k * k < 5 + 3 * k++) {
    System.out.println("INF1000");
}
```

Svar: ...4..... ganger

Oppgave 4 (5 poeng)

Anta at følgende kodelinjer utføres:

```
String res = "";
String s = "ab";
int p = 4;
res = res + (p++) + s + (p/2) + s.charAt(1);
res = res + s + s;
System.out.println(res);
```

Hva blir skrevet ut på skjerm?

Svar:4ab2babab.....

Oppgave 5 (20 poeng)

Skriv ferdig metoden under, som skal regne ut og returnere *samlet pris med moms* på en rekke varepriser som er angitt i arrayen **priser**. Du skal altså summere prisene du finner i denne arrayen, legge til moms på denne summen og til slutt returnere det svaret du da får. Momssatsen du skal bruke er 25%.

Svar:

```
double samletPrisMedMoms(double[] priser){

    double netto = 0;

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

    double brutto = netto * 1.25;

    return brutto;
}
```

Oppgave 6 (30 poeng)

Skriv ferdig programmet under ved å fullføre metoden `ordAnalyseAvBok`, som får som parameter navnet på en fil som inneholder teksten til en bok, og som finner ut hvilke ulike ord som er brukt i boka og hvor ofte hvert ord er brukt. Du skal bruke en `HashMap` og klassen `Ord` nedenfor. Metoden skal til slutt skrive ut på skjermen en liste over alle ordene i boka (hvert ord skal da bare forekomme en enkelt gang i denne listen, selv om det forekommer flere ganger i filen) og for hvert ord skal det skrives ut hvor mange ganger ordet forekommer i filen.

Hint: du kan bruke metoden `containsKey(String s)` i klassen `HashMap` for å teste om det ordet du akkurat har lest, er registrert tidligere i `HashMap`'en.

Med `easyIO` kan ordene i filen leses med kommandoen `inWord(" .,")`. Grunnen til parameteren `" .,"` er at vi ønsker at samme ordet skal returneres uavhengig av om det står like foran et punktum, et komma eller et vanlig mellomrom. Vi ønsker med andre ord å betrakte komma og punktum som skille tegn på lik linje med blanke tegn.

Svar:

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

class Oppgave6 {
    public static void main (String[] args) {
        Analyse ana = new Analyse();
        ana.ordAnalyseAvBok(args[0]);
    }
}

class Ord {
    String ord;
    int antall = 1;

    Ord (String ord) {
        this.ord = ord;
    }
}
```

```

class Analyse {

    void ordAnalyseAvBok(String filnavn) {
        In bok = new In(filnavn);
        HashMap <String,Ord> h = new HashMap <String,Ord>();

        while (bok.hasNext()) {
            String ordet = bok.inWord(" ,.");
            if (h.containsKey(ordet)) {
                h.get(ordet).antall++;
            } else {
                h.put(ordet, new Ord(ordet));
            }
        }

        for (Ord x : h.values()) {
            String s = x.ord + "\t(" + x.antall + " forekomster)";
            System.out.println(s);
        }
    }
}

```

Oppgave 7 (10 poeng)

```

int rest(int n, int m){
    while(n >= m) {
        n-=m;
    }
    return n;
}

```

Hva returneres fra metodekallet `rest(45,7)`?

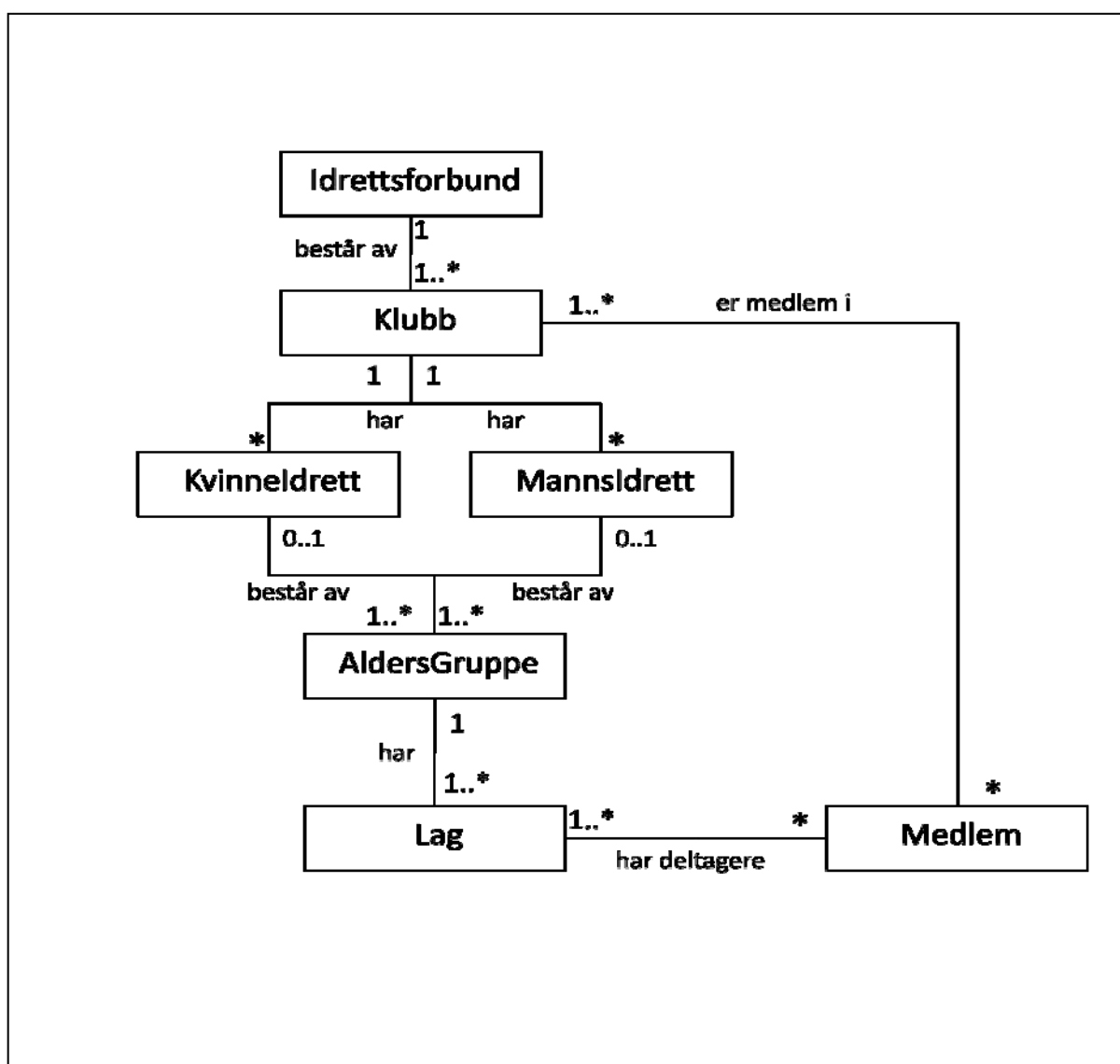
Svar:3.....

Oppgave 8 (20 poeng)

Idrettsforbundet i Ruritania består av en rekke klubber. Hver klubb har aktiviteter i en eller flere idretter. Man skiller mellom manns- og kvinne-idretter. Ikke alle klubber driver både manns- og kvinne-idrett i alle de idrettene de har. Idrettene er delt inn i aldersbestemte grupper (småjenter/smågutter, jenter/gutter, juniorer, ...), og det er ett eller flere lag i hver aldersgruppe som klubben har. Et lag kan bare ha deltagere fra én aldersgruppe. En person kan være medlem av en eller flere klubber og i hver klubb som en person er medlem av, er hun/han med i ett eller flere lag.

Tegn et UML-klassediagram med de følgende syv (Java-) klassene som kan brukes til å representere dette problemet: Idrettsforbund, Klubb, KvinneIdrett, MannsIdrett, AldersGruppe, Lag og Medlem. Gi navn på relasjonene mellom disse klassene slik det er beskrevet ovenfor og plassér antall på begge sider av hvert forhold. Gjør egne forutsetninger ved behov, men forklar isåfall disse.

Svar:



Oppgave 9 (60 poeng)

Firmaet Perfekt.no har et større varelager med tusenvis av varer. Oppgaven din nå er å lage et enkelt program som skriver ut regninger (fakturaer) på bestillinger som firmaet mottar over internett. Fakturaene skrives ut en gang per dag samtidig som varene pakkes. Perfekt.no har i varekatalogen sin alle de vareslagene som bestilles, men det kan hende at de ikke har stort nok antall på lager av en bestemt vare, og da må du ta hensyn til dette når du skriver faktura (se også nedenfor).

A) Varene som finnes på lageret ligger beskrevet på en fil: ”**Varelager.txt**”, med en linje for hver vare, og hvor hver opplysning på linjen er avsluttet med et semikolon(;). Formatet på en linje på fila er:

```
<Varenummer>;<Varenavn>;<antall på lager>;<pris per stykk uten moms>;
```

En linje i denne fila kan være:

```
1257;HPprinter;12;3450.50;
```

B) Kundene til Perfekt.no ligger på en annen fil ”**Kundene.txt**” (nye bestillere må alltid registrere seg som kunde før de kan komme med en bestilling). Det er én linje for hver kunde på fila, og hver opplysning avsluttes med et semikolon(;).

```
<KundeNr>;<Kundenavn>;<Kundeadresse>;
```

En linje på denne fila kan da være:

```
135;Per Petterson;Gågata 13 1444 Smalsund;
```

C) Bestillingene fra den dagen du skal kjøre faktura fra, ligger samlet på en tredje fil: ”**Bestillinger.txt**”, hvor det er én linje for hver bestilling av en bestemt vare fra en kunde. Det kan være én eller flere bestillinger (dvs. linjer) for hver kunde, men alle bestillingene fra en og samme kunde kommer etter hverandre på fila. En sliks bestillingslinje har følgende format:

```
<Kunde-nummer> ;<Varenummer>; <antall som bestilles>;
```

En linje i denne fila kan være:

```
135; 1257; 2;
```

(som betyr at Per Petterson har bestilt 2 stk. HP printere)

Oppgave:

Du skal først lese inn opplysningene i ”**Varelager.txt**” og ”**Kundene.txt**” og legge opplysningene i hver sin HashMap. Deretter skal du lese fila ”**Bestillinger.txt**” og skrive faktura for bestillingene for hver kunde – som for eksempel (se neste side):

Faktura

Per Petterson
Gågata 13 1440 Smalsund

2 stk. HP printer	a kr3450.50 = kr. 6901.00
+ moms 25%	= kr. 1725.25
SUM	=kr. 8626.25

(*Ikke* legg vekt på å få til en pen utskrift)

Regningen skal skrives ut på skjermen (enten med System.out.println eller med metodene i klassen Out fra easyIO).

Tips: Du oppdager at bestillingene for én kunde er slutt når kundenummeret på neste linje ikke er lik kundenummeret til den kunden du holder på med (eller at det er slutt på fila).

NB: Hvis du **ikke** har nok på lager av en vare, så skriver du faktura på det antallet du har på lageret og kan levere nå – og kommer med forklarende tekst på fakturaen at det resterende på bestillingen vil bli levert (og fakturert) senere.

Husk å trekke fra på varelageret (i HashMap'en) det antallet av hver vare du har levert og sendt faktura på. Når du er ferdig med alle bestillingene fra "**Bestilling.txt**", må du skrive ut den nye versjonen av filen "**Varelager.txt**" til disken hvor antallene på de varene du nå har solgt og sendt regning for, er trukket fra.

Svar:

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

class FakturaSystem {
    public static void main(String[] args) {
        Fakturaer f = new Fakturaer();
        f.sendRegninger();
    }
}

class Kunde {
    String nummer;
    String navn;
    String adresse;

    Kunde(String nummer, String navn, String adresse){
        this.nummer = nummer;
        this.navn = navn;
        this.adresse = adresse;
    }

    String fåNummer() {return nummer;}
    String fåNavn() {return navn;}
    String fåAdresse() {return adresse;}
}
```



```

class Vare {
    String nummer;
    String navn;
    int antall;
    double pris;

    Vare(String nummer, String navn, int antall, double pris) {
        this.nummer = nummer;
        this.navn = navn;
        this.antall = antall;
        this.pris = pris;
    }

    String fåNummer() {return nummer;}
    String fåNavn() {return navn;}
    int fåAntall() {return antall;}
    double fåPris() {return pris;}

    int taUt(int antall) {
        int tasUt = Math.min(antall, this.antall);
        this.antall = this.antall - tasUt;
        return tasUt; // Returner antall tatt ut
    }
}

class Fakturaer {
    HashMap <String,Vare> varelager = new HashMap<String,Vare>();
    HashMap <String,Kunde> kunder = new HashMap<String,Kunde>();

    Fakturaer() {
        // Les varelager fra fil
        In varefil = new In("Varelager.txt");
        while (varefil.hasNext()) {
            String nummer = varefil.inWord(" ");
            String navn = varefil.inWord(";");
            int antall = varefil.inInt(" ");
            double pris = varefil.inDouble(";");
            varelager.put(nummer, new Vare(nummer, navn, antall, pris));
        }
        varefil.close();

        // Les kunder fra fil
        In kundefil = new In("Kundene.txt");
        while (kundefil.hasNext()) {
            String nummer = kundefil.inWord(" ");
            String navn = kundefil.inWord(";");
            String adresse = kundefil.inWord(";");
            kunder.put(nummer, new Kunde(nummer, navn, adresse));
        }
        kundefil.close();
    }
}

```

```

void avsluttFaktura (double sum) { // Avslutt forrige faktura
    System.out.println("+ moms 25%    = kr " + sum * 0.25);
    System.out.println("SUM          = kr " + sum * 1.25 + "\n\n");
}

void sendRegninger() {
    In bestillingsfil = new In("Bestillinger.txt");
    boolean førsteKunde = true;
    double totalbeløp = 0.0;
    String kundeNr = "";

    // Les neste linje av bestillingsfila
    while (bestillingsfil.hasNext()) {
        String forrigeKundeNr = kundeNr;
        kundeNr = bestillingsfil.inWord(" ;");
        boolean nyKunde = !forrigeKundeNr.equals(kundeNr);
        String varenummer = bestillingsfil.inWord(" ;");
        int ønsketAntall = bestillingsfil.inInt(" ;");

        // Skriv ut overskrift og evt avslutt forrige
        if (nyKunde) {
            if (!førsteKunde) avsluttFaktura(totalbeløp);
            else førsteKunde = false;

            // Skriv overskrift til ny faktura
            Kunde k = kunder.get(kundeNr);
            System.out.println("Faktura\n" );
            System.out.println(k.fåNavn() + "\n" + k.fåAdresse() + "\n");
            totalbeløp = 0;
        }

        // Finn ut om nok av varen på lager, oppdater lagerstatus
        Vare v = varelager.get(varenummer);
        int kanLevere = v.taUt(ønsketAntall);
        double beløp = kanLevere * v.fåPris();
        totalbeløp = totalbeløp + beløp;

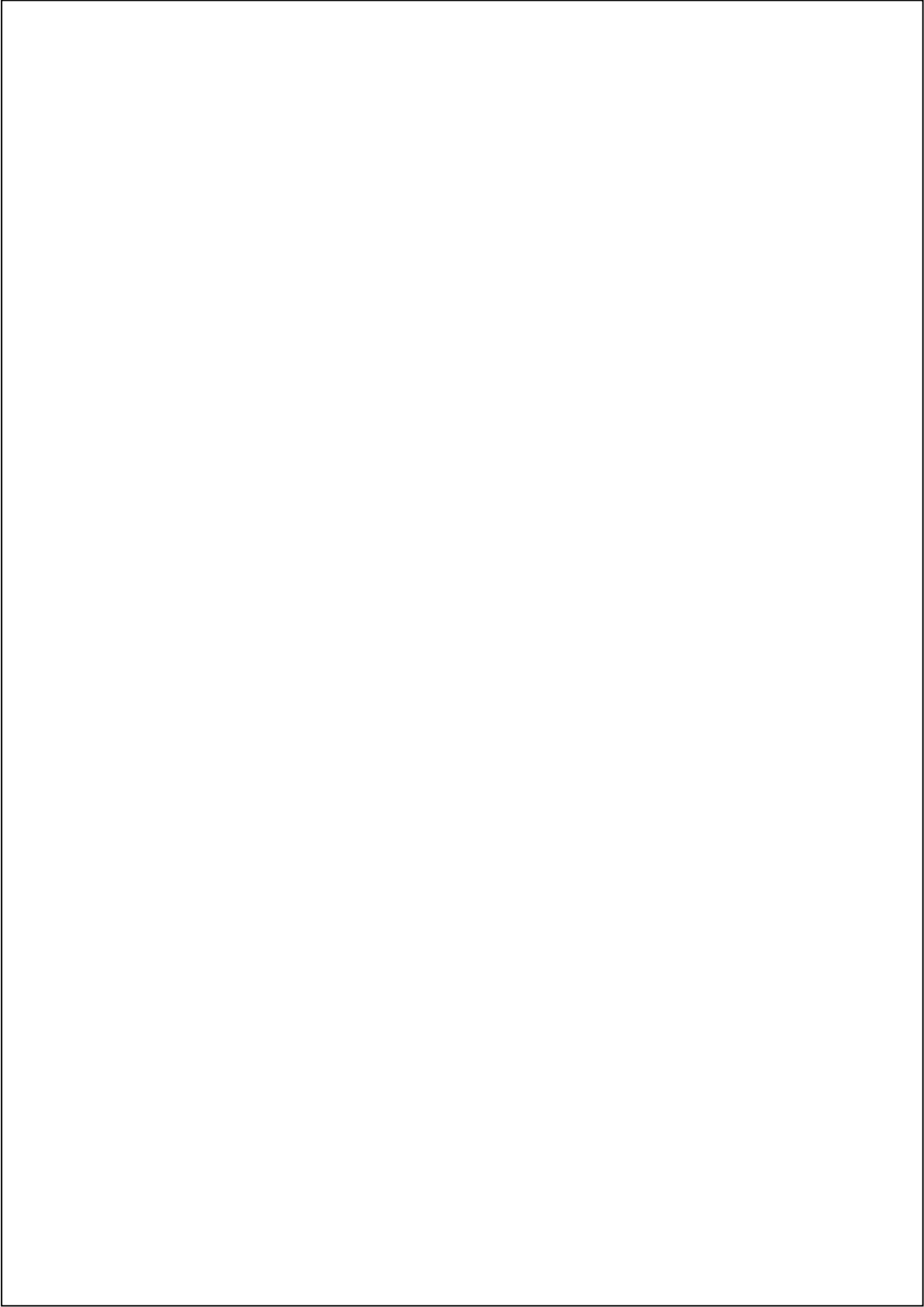
        // Skriv ut fakturalinje
        System.out.print(kanLevere + " stk. " + v.fåNavn() +
            " a kr. " + v.fåPris() + " = kr. " + beløp);
        if (kanLevere < ønsketAntall) {
            System.out.print("MANKO:" + (ønsketAntall-kanLevere) + "stk");
        }
        System.out.println();
    }

    // Avslutt siste faktura
    avsluttFaktura(totalbeløp);
    bestillingsfil.close();

    // Skriv ut oppdatert varelager
    Out var = new Out("Varelager.txt");
    for (Vare va : varelager.values()) {
        String s = va.fåNummer() + ";" + va.fåNavn() + ";";
        s = s + va.fåAntall() + ";" + va.fåPris() + ";";
        var.outln(s);
    }

    var.close();
} // end sendRegninger
} // end class Fakturaer

```



Oppgave 10 (15 poeng) (Litt vanskelig)

Anta at følgende program utføres (for definisjon av %-operatoren se evt. læreboka s.37):

```
class K {
    static K[] fem = new K[5];

    String s;

    K(String tekst, int i){
        s = tekst;
        fem[(i+2)%5] = this;
    }

    public static void main(String[] args) {
        String s = "";

        for( int i = 0; i < 5; i++) {
            s+= i;
            new K(s,i);
        }

        for( int i = 0; i < 5; i++) {
            System.out.println(K.fem[i].s);
        }
    } // end main
}
```

Hva skriver programmet ut på skjermen?

Svar:

```
0123
01234
0
01
012
```

Oppgave 12 (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:

Arno har blitt selger i bilbutikken UqbarElectricØkoCars og vil nå lage et utvidet kunderegister for å få fart på salget. Nå av vil han i tillegg til navn, adresse fra tidligere kunder, også utvide registeret med nye, potensielle kunder. Arno vet at de nye elektriske bilene som ikke slipper ut CO₂, appellerer særlig til høyt utdannede personer under 40 år med middels inntekt. Denne gruppa stemmer også ofte på de venstreorienterte partiene i Uqbar. Arno samler derfor inn all den offentlige tilgjengelige data han kan finne, som:

- Navn. (får du fra universitetets data-anlegg, bl.a lister studenter som har levert masteroppgave)
- Adresse får du fra Telefonlister på nettet (Gule Sider).
- Fødselsår og inntekt får du fra skattelisten som også er lagt ut på nettet

- Politisk partitilhøriget får han ved å samle inn lister fra de som ar stilt til valg for de ulike partiene i de siste kommevalgene. Han finner også noen store grupper av underskrivere av ulike miljøopprop og medlemmer av ulike grupper på Facebook.
- Siden Arno tror at de økobilene han selger ikke appellerer så godt til innvandrere, lager han en lang liste av navn som er typiske for denne gruppen og fjerner (med et lite program han har skrevet selv) alle med slike navn fra listene dine.

Du skal nå drøfte om dette er greie opplysninger som Arno har rett til å samle inn eller om han må søke Datatilsynet i Uqbar om lov til dette. Begrunn svaret med å vise til konkrete paragrafer som du mener er relevante for dette registeret og de opplysningene Arno ønsker å samle inn, og hvorfor de ulike paragrafene du nevner er relevante.

Systemet kommer klart inn under §2, fordi han registrerer sensitive opplysninger, f.eks politiske holdninger. Videre står det i §9 at det kan han bare gjøre hvis de registrerte samtykker (noe de ikke gjør her) – han henter alt fra nettet og lignende. Han registrerer også etnisk tilhørighet indirekte ved å fjerne innvandrere (og sitter igjen med en 'ren' hvit liste) - igjen §2.

Fordi han klart kommer inn under §2 må han søke Datatilsynet om tillatelse (konsesjon) §33 – noe de etter alle solemerker vil avslå fordi de ikke har spurt de registrerte om lov (§9) og fordi han generelt registrer opplysninger som bryter §8 i og med at det er vanskelig å begrunne at et bil-firma trenger å sitte på slike data uten at det er samtykke fra de registrerte.

Det er enkelt å se at han også bryter §11 a) fordi det ikke er tillatt etter §8 og §9. Han bryter også §11 c) siden opplysningene var samlet inn for helt andre formål og kan ikke nyttes til nye formål uten å spørre de registrerte.

Konklusjon: Systemet bryter §2, 8, 9 og 11.

*For å få 6 poeng bør det argumenteres for at systemet er ulovlig etter minst en av disse §-ene. En fullpoengsløsning bør nevne minst tre av disse §-ene. Husk at det ikke er nok å nevne en paragraf – det må også være en setning som forklarer **hvorfor** denne §-en er relevant.*

§2 nr.8: Sensitive opplysninger er opplysninger om:

- Rase, etnisk, poilitiske meninger ..., osv
- Om man er dømt, mistenkt .el.
- Helse- og seksuelle forhold
- medlemskap i fagforeninger

§ 8 Behandling av personopplysninger skal bare gjøres etter samtykke og være saklig begrunnet, må være nødvendig før:

- Oppfylle avtale med den registrerte
- Ivareta den registrertes interesser
- å utøve offentlig myndighet
- å ivareta en berettiget interesse som overstiger den registrertes interesse

§9 Behandling av sensitive opplysninger, bare hvis

- den registrerte samtykker
- fastsatt i lov
- g – h) nødvendig av helse, statistiske undersøkelser

§ 11. Grunnkrav til behandling av personopplysninger

Den behandlingsansvarlige skal sørge for at personopplysningene som behandles

- bare behandles når dette er tillatt etter § 8 og § 9,
- bare nyttes til uttrykkelig angitte formål som er saklig begrunnet i den behandlingsansvarliges virksomhet,
- ikke brukes senere til formål som er uforenlig med det opprinnelige formålet med innsamlingen, uten at den registrerte samtykker,
- er tilstrekkelige og relevante for formålet med behandlingen, og
- er korrekte og oppdatert, og ikke lagres lenger enn det som nødvendig ut fra formålet med behandlingen, jf. § 27 og § 28.

§ 31 - 32 Meldeplikt til Datatilsynet for registre (30 dager)

§ 33 Konsesjon for sensitive personopplysninger