



Det matematisk-naturvitenskapelige fakultet

Eksamen i INF1000 — Grunnkurs i objektorientert programmering

Eksamensdag: Prøveeksamen tirsdag 23. november 2010

Tid for eksamen: 12.00–16.00

Oppgavesettet er på 18 sider.

Vedlegg: Ingen

Tillatte hjelpemidler: Alle trykte og skrevne

Kontroller at oppgavesettet er komplett før du begynner å besvare spørsmålene.

- Les nøye gjennom hver oppgave før du løser den. For hver oppgave er det angitt det maksimale antall poeng du kan få hvis du svarer helt riktig. Summen av poengene er 100, slik at f.eks 5 poeng tilsvarer 12 minutter, 10 poeng tilsvarer 24 minutter, osv. (hvis du regner med å komme igjennom alt). Pass på at du bruker tiden din riktig. Setter du av 20 minutter til 10 poeng, får du 40 minutter til å se over alt til slutt!
- 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. Det er satt av plass i oppgavesettet slik at du kan skrive inn svarene dine.
- Får du likevel ikke plass nok i feltene i oppgaveteksten, kan du fortsette på eget ark.
- 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.*

Innhold

1 Løkker (5 poeng)	side 3
2 Array og løkker (5 poeng)	side 4
3 Java-syntaks (5 poeng)	side 5
4 String (5 poeng)	side 5
5 Metoder (10 poeng)	side 6
6 UML (10 poeng)	side 8
7 Klasser, objekter og array (15 poeng)	side 9
8 Sortering (15 poeng)	side 11
9 Klasser og objekter (25 poeng)	side 13
10 Personopplysningsloven (5 poeng)	side 17

Lykke til!

Oppgave 1 Løkker (5 poeng)

Hvor mange ganger blir "Eksamen" skrevet ut?

1a

```
for (int i=10; i>=5; i--) {  
    for (int j=0; j<6; j++) {  
        System.out.println("Eksamen");  
    }  
}
```

Svar: _____

1b

```
for (int i=5; i>0; i--) {  
    for (int j=0; j<i; j++) {  
        System.out.println("Eksamen");  
    }  
}
```

Svar: _____

1c

```
for (int i=0; i<5; i++) {  
    int k = 5-i;  
    while (k > 0) {  
        System.out.println("Eksamen");  
        k--;  
    }  
}
```

Svar: _____

1d

```
for (int i=0; i<10; i+=2) {  
    for (int k=3; k>-1; k--) {  
        System.out.println("Eksamen");  
        k--;  
    }  
}
```

Svar: _____

(Fortsettes på side 4.)

Oppgave 2 Array og løkker (5 poeng)

Gitt følgende array:

```
double [][] tall = new double [6] [5];
```

2a

Hvor mange double-verdier er det plass til i arrayen?

Svar: _____

2b

Vil følgende java-kode skrive ut innholdet i **alle** plassene i arrayen?

Ja Nei

```
for (int i=0; i<tall.length; i++) {  
    for (int j=0; j<tall[i].length; j++) {  
        System.out.println(tall[i][j]);  
    }  
}
```

```
for (int i=tall.length; i>0; i--) {  
    for (int j=tall[i-1].length; j>0; j--) {  
        System.out.println(tall[i-1][j-1]);  
    }  
}
```

```
for (int i=0; i<tall.length; i++) {  
    for (int j=4-i; j>=0; j--) {  
        System.out.println(tall[i][j]);  
    }  
}
```

(Fortsettes på side 5.)

Oppgave 3 Java-syntaks (5 poeng)

Er disse uttrykkene lovlige programsetninger i Java?

- | Ja | Nei | |
|--------------------------|--------------------------|--|
| <input type="checkbox"/> | <input type="checkbox"/> | <code>int i = false;</code> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>enum E {A,B,C;}</code> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>boolean ok = 3 > 2;</code> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>double tall = new Student("Lise");</code> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>String[] navn = { { "Per", "Lise"}, { "Geir", "Liv"} };</code> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>String[] ord = { "stein", "saks", "papir"};</code> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>boolean riktig = (new double[3]).length > 3;</code> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>class A{}</code> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>double pris = 124.99 kroner;</code> |
| <input type="checkbox"/> | <input type="checkbox"/> | <code>String tegn = 'a';</code> |

Oppgave 4 String (5 poeng)

Anta at følgende programkode kjøres:

```
String[] navn = { "Per", "Per" };
if (navn[0].equals(navn[1])) {
    navn[0] += navn[1];
    navn[1] += navn[0];
}
if (navn[0] == navn[1]) {
    System.out.println("Pekerlikhet");
} else {
    System.out.println("Ikke pekerlikhet");
}
if (navn[0].equals(navn[1])) {
    System.out.println("Like");
} else {
    System.out.println("Ulike");
}
```

Hva skrives ut? Skriv svaret her:

(Fortsettes på side 6.)

Oppgave 6 UML (10 poeng)

En bilforretning skal lage et Java-program for å holde orden på biler, selgere, kunder og salg. Bilforretningen kan ha flere selgere og flere kunder. I tillegg skal forretningen ha en oversikt over hvilke biler som er til salgs. Et salg er en avtale mellom én selger, én kunde og én bil. En selger kan selge flere biler, en kunde kan kjøpe flere biler, men en bil kan kun selges én gang.

Tegn et UML klassediagram for Java-programmet. Gi navn til klassene, tegn forhold mellom dem og sett på riktig antall på hver side av forholdene. Du behøver *ikke* skrive inn variable og metoder i klassene i diagrammet.

Tegn UML-diagrammet her:



(Fortsettes på side 9.)

Oppgave 7 Klasser, objekter og array (15 poeng)

Gitt følgende Java-klasse:

```
class Person{
    String navn;
    int tlfnr; // tlfnr er 0 hvis personen ikke har telefon

    Person(String navn){
        this.navn = navn;
        this.tlfnr = 0;
    }

    Person(String navn, int tlfnr){
        this.navn = navn;
        this.tlfnr = tlfnr;
    }

    boolean harTlf(){
        return( tlfnr != 0 );
    }
}
```

Lag et fullstendig program, inkludert en main-metode, som bruker Person-klassen og som

- oppretter en array med lengde 1000 i konstruktøren til **class PersonArray** der elementene skal være Person-objekter,
- implementerer en settInn-metode for å sette inn Person-objekter i arrayen
- oppretter et personobjekt for “Ole Olsen” med telefonnr 45000235 og legger inn i arrayen,
- oppretter et personobjekt for “Kari Karlsen” uten telefon og legger inn i arrayen, og
- går i en løkke gjennom alle objektene i arrayen og skriver ut navnene til de som ikke har telefon. Du skal bruke objektene harTlf()-metoder når du løser oppgaven.

I dette tilfellet skal programmet skrive ut **Kari Karlsen**, men det skal virke også om vi legger flere Person-objekter inn i arrayen. Du kan anta at vi aldri vil ta objekter ut av arrayen.

(Skriv inn svaret ditt på neste side...)

(Fortsettes på side 10.)

```
class PersonArray {
```

```
}
```

(Fortsettes på side 11.)

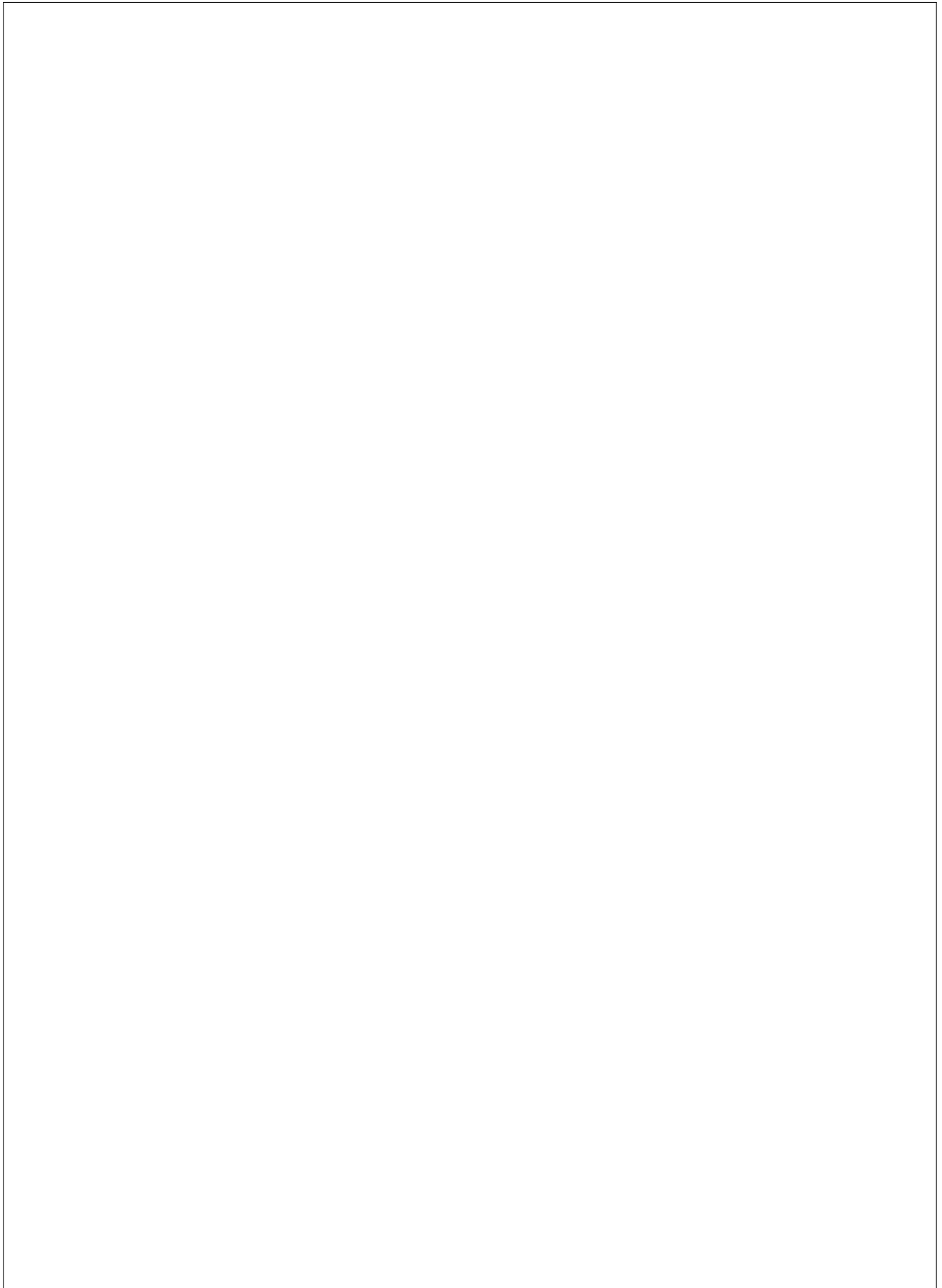
Oppgave 8 Sortering (15 poeng)

Vi skal arbeide videre på `PersonArray`-klassen i Oppgave 7. Fremdeles skal vi anta at det er maksimalt 1000 kunder som ligger i arrayen. Men nå skal vi også anta at `PersonArray`-klassen er et arkiv over ansatte i et firma som tilbyr firmatelefon der alle *telefonnumrene* ligger i intervallet 45000000, ..., 45001999 og at alle personer har forskjellig telefonnummer.

Du skal nå skrive en metode som returnerer en array over alle `Person`-objekter fra `PersonArray`-klassen som har telefon. Arrayen du skal returnere skal være *sortert etter stigende telefonnummer*. Når du implementerer denne metoden, skal du utnytte det vi vet om intervallet som telefonnumrene ligger i ved å bruke følgende algoritme:

- Opprett en array `temp` med plass til 2000 `Person`-objekter.
- Gå gjennom arrayen med `Person`-objekter i `PersonArray`-klassen og legg alle som har telefon inn i `temp` slik at `temp[i].tlfnr == 45000000 + i`. (*Hint*: Legg `Person`-objekt p inn på posisjon `temp[p.(tlfnr-45000000)]`.)
- Mens du gjør dette, hold orden på hvor mange objekter som ligger i `temp`. La en `int`-variabel `antall` lagre dette antallet.
- Gå gjennom `temp` fra indeks 0 og utover. En del av plassene i `temp` vil være `null`, men en del vil også referere til `Person`-objekter. Skriv `Person`-objektene over til en array med `Person`-objekter med lengde `antall` i den rekkefølge de besøkes når du går nedover `temp`. Arrayen du skriver over til fra `temp` vil være korrekt sortert, og metoden returnerer arrayen til slutt.

(Fortsettes på side 12.)



(Fortsettes på side 13.)

Oppgave 9 Klasser og objekter (25 poeng)

I denne oppgaven skal du skrive to klasser, `Medlem` og `Idrettslag`, som skal brukes av et idrettslag til å holde orden på hvilke medlemmer som har betalt kontigent. I klassen `Medlem` skal medlemsnummer, navn og telefonnr lagres som `String`-verdier. Videre skal det lagres informasjon om hvorvidt kontigenten er betalt. Klassen skal inneholde en konstruktør med parametre `String medlemsnr`, `String navn` og `String tlf`, og som initialiserer de respektive verdiene i objektet.

I klassen `Idrettslag` skal det være datastruktur for å lagre `Medlem`-objekter. Klassen skal i tillegg inneholde følgende metoder:

- En konstruktør som initialiserer datastrukturen.
- `void leggTilMedlem(Medlem m)`
Skal legge til `Medlem m` i datastrukturen.
- `void slettMedlem(String medlemsnr)`
Skal slette medlemmet med medlemsnr `medlemsnr`. Hvis medlemmet ikke finnes, skal det ikke gjøres noen ting.
- `Medlem hentMedlem(String medlemsnr)`
Skal sjekke om det finnes et medlem med medlemsnr `medlemsnr`. Hvis medlemmet finnes, skal riktig `Medlem`-objekt returneres. Hvis ikke, skal **null** returneres.
- `void sok(String tekst)`
Skal søke gjennom alle medlemmene og sjekke om *enten* navn *eller* telefonnr er tekstlig lik søkestrengen `tekst`. De medlemmer som har navn eller telefonnr lik `tekst` skal skrives ut på skjermen med medlemsnr, navn og telefonnr.
- `void registrerBetaling(String medlemsnr)`
Skal registrere at medlemmet med `medlemsnr` har betalt kontigenten. Hvis medlemmet ikke finnes, skal en passende feilmelding skrives ut på skjermen.
- `void skrivPurreliste()`
Skal skrive ut en liste på skjermen med medlemsnr, navn og telefonnr til de medlemmene som *ikke* har betalt kontigenten.
- `void harLikeTlfnr()`
Skal sjekke om noen medlemmer har like telefonnr. Hvis ingen medlemmer har like telefonnr, skal en passende melding skrives ut på skjermen. Hvis det finnes medlemmer med samme telefonnr, skal det skrives ut en linje med telefonnr og deretter medlemsnr og navn på de medlemmene som har dette telefonnr. Det skal skrives ut én slik linje for hvert telefonnr som deles av flere medlemmer.

Merk at du kun skal skrive Java-kode for det som er nevnt i teksten over. Du skal *ikke* skrive hverken `main`-metoden eller ordreløkke. *Du finner plass til å besvare oppgaven på de følgende sidene.*

(Fortsettes på side 14.)

```
class Medlem {
```

```
}
```

```
class Idrettslag {
```

(Fortsettes på side 15.)

(Fortsettes på side 16.)

}

(Fortsettes på side 17.)

Oppgave 10 Personopplysningsloven (5 poeng)

Petter Sprett er leder i den lokale idrettsforeninga Varild og samtidig eier av FriskSport, en av stedets to sportsbutikker. Petter vil gjerne bruke sin posisjon som leder av sportsklubben til å både gi medlemmene et bedre tilbud og hjelpe på omsetningen i sin butikk med å lage et nytt datasystem for Varild. I første runde registrerer han navn, adresse, alder, kontingentbetaling og idrettsgren de driver med for alle medlemmene i Varild. Han bruker så medlemsregisteret til å sende ut følgende.

- a) Et brev til alle medlemmer av Varild om at de får 8 % rabatt ved å handle i FriskSport ved framvisning av medlemskort.

Etter ett år finner han ut at for å drive kampanje for utvalgte varer han får inn butikken, bør han utvide datasystemet i Varild. Det har også kommet en rekke innvandrere til stedet. Mange av disse har meldt seg inn i Varild, og Petter vil gjerne finne sportsprodukter som dekker deres behov. Han registrerer nå i tillegg hvilket land medlemmene er født i og kjønn for bedre å kunne sende ut brev til bare de medlemmene som er interessert i hans produkter (særlig tenker han på det cricketutstyret han nå selger).

- b) Som den lovlidige (?) innbygger Petter er, sender han nå en melding til Datatilsynet om utvidelsen av datasystemet, og begynner så å sende ut kampanjebrev fra butikken til noen av medlemmene om utvalgte sportsprodukter med spesialpriser (cricketutstyr, sportshjåber, slalåmutstyr osv.). Ved å sende ut slik målrettet reklame sparer Petter porto og medlemmene får bare den reklamen som passer for dem.

Vurdér om det han gjør under a) og b) er lovlig i følge *Personopplysningsloven*. Hvis Petter har brutt noen av paragrafene, så nevne hvilke paragrafer han har brutt og hvorfor.

(Fortsettes på side 18.)

