

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Prøveeksamen i : INF1000 — Grunnkurs i objektorientert programmering

Prøveeksamensdag : Torsdag 28. november 2013

Tidspunkter: 10:15 Utdeling av prøveeksamen

14:00 Pizza-servering (1. etg Ole-Johan Dahls hus)

14:30 Gjennomgang av løsning (Auditorium: Simula)

Oppgavesettet er på : 5 sider

Vedlegg : Ingen

Tillatte hjelpemidler : Alle trykte og skrevne

- Kontroller at oppgavesettet er komplett, og les nøye gjennom oppgavene før du løser dem. Poengangivelsen øverst i hver oppgave angir maksimalt antall poeng.
- Du kan legge dine egne forutsetninger til grunn og gjøre rimelige antagelser, så lenge de ikke bryter med oppgavens "ånd". Gjør i så fall rede for disse forutsetningene og antagelsene.
- Til eksamen skal svarene skrives på gjennomslagspapir. Da må du huske å skrive hardt nok til at besvarelsen blir mulig å lese på alle gjennomslagsarkene, og ikke legge andre deler av eksamensoppgaven under når du skriver.
- Til eksamen skal du selv beholde underste arket etter levering av de to øverste til eksamensinspektøren. Nummerer sidene, og husk å skrive kandidatnummeret ditt på besvarelsen.

Oppgave 1 (6 poeng)

a) Hvor mange double-verdier er det plass til i denne arrayen:

```
double[][] vekt = new double[9][9];
```

b) Hva blir skrevet ut her?

```
String txt = "OLE";  
int n = txt.length();  
for (int i=0; i<n; i++) {  
    System.out.print(txt.charAt(n-i-1));  
}
```

Oppgave 2 (10 poeng)

Angi hvilke av disse programsetningene er lovlige i Java, og for de som er lovlige skriv hvilken verdi variabelen i venstre side av tilordningen får.

- a) `String verdi1 = "double";`
- b) `double verdi2 = "String";`
- c) `double vekt1 = 5/2;`
- d) `double vekt2 = 5.0/2.0;`
- e) `int vekt3 = 5.0/2.0;`
- f) `double resultat = 5.2 + 14.6 * 3.7 + resultat;`
- g) `String svar1 = "2 + 3.14 = " + (2 + 3.14);`
- h) `boolean svar2 = true && false;`
- i) `boolean svar3 = true || false;`
- j) `boolean svar4 = (true && false) || (true || false);`

Oppgave 3 (9 poeng)

a) Hvor mange ganger blir "Arne" skrevet ut her?

```
for (int k = 0; k < 5; k++) {
    for (int r = 0; r < 20; r++) {
        System.out.println("Arne");
    }
}
```

b) Hvor mange ganger blir "Arne" skrevet ut her?

```
for (int k = 0; k < 5; k++) {
    for (int r = k; r < 20; r++) {
        System.out.println("Arne");
    }
}
```

c) Hvor mange ganger blir "Arne" skrevet ut her?

```
for (int k = 0; k < 20; k++) {
    for (int r = k; r < 5; r++) {
        System.out.println("Arne");
    }
}
```

Oppgave 4 (15 poeng)

I programmer kan det være behov for å konkatenerer (slå sammen) to tekststrenger, slik som vi gjør når vi skriver `String t = s1 + s2;` der `s1` og `s2` er to `String`-variable. I denne oppgaven skal du lage en metode som gjør det samme, men mange ganger. Mer presist skal du lage en metode som har to parametre: en `String`-array og en `String`. Metoden skal konkatenerer hvert enkelt element i `String`-arrayen med den angitte `String`-verdien og returnere resultatet i en

ny String-array som er like lang som den opprinnelige. Hvis metoden heter **konkatener**, så skal f.eks. setningene

```
String[] ordliste = {"Karoline", "Anders", "Camilla"};
String[] resultat = konkatener(ordliste, " er flink");
```

resultere i at **resultat** blir en String-array av lengde 3 med verdiene "Karoline er flink", "Anders er flink" og "Camilla er flink", henholdsvis.

Oppgave 5 (25 poeng)

Skriv en metode med en int-array som parameter og som returnerer en int-verdi, og som skal finne ut hvilken verdi som forekommer flest ganger i arrayen og returnere denne. Hvis det er flere verdier i arrayen som tilfredsstiller betingelsen, skal metoden returnere med en av disse (vilkårlig hvilken). Eksempel: hvis arrayen som gis som parameter inneholder verdiene

2 6 6 3 8 1 3 9 14

så forekommer 3 og 6 to ganger, og metoden skal returnere en av dem (vilkårlig hvilken).

Oppgave 6 (25 poeng)

Anta at tekstfilen "Ord.txt" inneholder en rekke ord atskilt av blanke tegn (du kan anta at filen ikke inneholder noen spesialtegn som punktum, komma, kolon, osv). Skriv et fullstendig program som først leser filen og som deretter skriver ut på skjermen hvor mange ulike ord av lengde 1, lengde 2, lengde 3, og lengde 4 som forekommer i filen. Eksempel: dersom filen har følgende innhold:

```
it was a dark and stormy night
the rain fell in torrents except at occasional intervals
when it was checked by a violent gust of wind
which swept up the streets
```

så skal programmet skrive ut på skjermen følgende (teksten i parentes til høyre på hver linje er bare ment som en kommentar til deg og skal ikke være en del av utskriften):

```
Ord av lengde 1: 1      ("a")
Ord av lengde 2: 6      ("it", "in", "at", "by", "of", "up")
Ord av lengde 3: 3      ("was", "and", "the")
Ord av lengde 4: 6      ("dark", "rain", "fell", "when", "gust", "wind")
```

Du kan anta at ordene bare inneholder små bokstaver, og du kan anta at bare tekstlig helt like ord regnes som samme ord, slik at f.eks. "rain" og "rains" er to ulike ord. Bruk enten String-array eller HashMap til å lagre ordene du leser inn fra fil.

Oppgave 7 (8 poeng)

- Skriv binærtallet 1110 som et desimaltall.
- Skriv desimaltallet 63 som et binærtall.
- Skriv binærtallet 11001 som en sum av to binærtall (hvor begge større enn binærtallet 10).
- Skriv desimaltallet 20 som et heksadesimalt tall.

Oppgave 8 (80 poeng)

En norsk kommune har et antall barneskoler hvor det undervises skolebarn på 7 klassetrinn (klassetrinn 1 til klassetrinn 7). Et barn har rett til å være elev på en bestemt skole ut fra hvor barnet bor; kommunene er da delt opp i like mange skolekretser som det er barneskoler. Et unntak er en livssynsskole som for eksempel Steinerskolen, som tar opp elever fra hele kommunen. En skole har et visst antall klasserom, og kan ikke ha flere klasser enn antall klasserom. Det kan være flere klasser på hvert trinn (f.eks. klasse 4A og 4B). I en klasse kan det maksimalt være 28 elever. Hver skole har en rektor og hver klasse har en kontaktlærer og ingen lærer er kontaktlærer i mer enn en klasse. Som du husker selv fra skolen kan det være mer enn en lærer pr time i en klasse og ikke samme lærere hele tiden i en bestemt klasse.

Du skal nå lage de aller første delene av et veldig forenklet administrasjonssystem for skoler, elever, rektorer og lærere.

- Tegn et UML-diagram over forholdet mellom klassene: Kommune, Skole, Klasserom, Klasse, Elev og Lærer. Skriv navn og antall på alle forholdene i diagrammet.
- Deklarer Java-klasser for ulike klassene i UML-diagrammet, og legg inn de objektvariablene du synes klassene skal ha ut fra beskrivelsen ovenfor. Blant annet har de fleste av objektene et navn.
- Til å lette arbeidet med å starte systemet, er det utarbeidet en fil **skolene2012.txt** (en for hver kommune) som beskriver kommunens skoler. De tre første linjene inneholder kommunenavnet, skoleåret filen gjelder for (skoleåret 2011/2012 kalles 2012), og antall skoler i kommunen. Deretter er det en linje per skole; denne inneholder først skolens navn og så antall klasserom. Hvis vi velger Nesodden som kommunen vi lager system for i 2013, vil fila vi leser for forrige år (2012) se slik ut:

```
Nesodden
2012
7
Berger, 14
Nesoddtangen , 25
Bjørnemyr, 10
Myklerud , 15
Jar, 7
Fjellstrand, 9
Nesodden Steinerskole, 14
```

Skriv en konstruktør i klassen Kommune som leser denne fila (uansett hvilke kommune dette gjelder) og oppretter det nødvendige antall skoler og klasserom.

- d) (Kan puffes) For videre å lette arbeidet før oppstart av et nytt skoleår (her 2013) er det for hver skole en fil som beskriver antall elver, rektors navn, alle lærernes navn og så for hver klasse, de elevene som gikk der i fjor. Vi antar at ingen flytter i sommerferien. For Berger Skole ved avslutningen i juni 2012 heter filen: Berger2012.txt , for Jar Skole: Jar2012.txt, .. osv. . Husk når du leser denne filen, så skal elevene i 7. klasse gå over til ungdomskolen (som ikke er en del av dette systemet), og de andre skal rykke opp en klasse, slik at de som for eksempel gikk i klasse 2A i 2012 skal nå i 2013 gå i klasse 3A, osv.

Fila inneholder først en linje med rektors navn, antall lærere på skolen, og alle lærernes navn (opplysningene på linjen er atskilt med komma) og en linje med antall klasser på skolen (i eksemplet nedenfor har skolen 14 klasser). Deretter kommer det informasjon om alle klassene på skolen. Informasjonen om en enkelt klasse består av en innledende linje med klassens navn (f.eks. 7A), kontaktlærerens navn (f.eks. Geir By), og antall elever i klassen (f.eks. 14), fulgt av en rekke linjer som gir elevenes navn. De fem første linjene for Jar Skole kunne da være:

```
Snillfrid Klem, 19, Ola Busk, Geir By, <+17 flere lærernavn >  
14  
7A,Geir By,14  
Stein Basse Nesland  
Omar Pamuk
```

.....

Skriv kode der denne filen leses i konstruktøren til det tilhørende Skole-objektet for den skolen som omtales (Hint: Leser du fila med klassen In fra easyIO, vil inWord(",") og inInt(",") lese alt til neste komma på en linje som henholdsvis en String eller heltall. Metoden inLine() leser alt på en linje – for eksempel et navn).

Oppgave 9 (20 poeng)

Det er harde tider i skolen, og en av rektorene i oppgave 8 er på desperat jakt etter nye inntekter til skolen sin (ikke til seg selv). Han tenker seg å selge klasselister (navn + adresse) til næringsdrivende som selger 'skikkelige' produkter (klær, plater, leker, mobiltelefoner o.l) slik at de kan sende reklame hjem til elevenes foreldre og barna selv.

- Er dette tillatt etter personvernloven?
- Hvis rektor søker konsesjon, er det da tillatt?

Begrunn svarene med henvisning til relevante paragrafer i loven.