



Prøveeksamen inf 1000
28. november 2013

Arne Maus og Ole Christian Lingjærde

Oppgave 1 (6 poeng)

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

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

Svar: Det er plass til $9 * 9 = 81$ double-verdier

Oppgave 1 (6 poeng)

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));  
}
```

Svar:

Vi setter $n=3$ og

$i = 0 \rightarrow n-i-1 = 3-0-1 = 2 \rightarrow \text{txt.charAt}(n-i-1) = \text{"E"}$

$i = 1 \rightarrow n-i-1 = 3-1-1 = 1 \rightarrow \text{txt.charAt}(n-i-1) = \text{"L"}$

$i = 2 \rightarrow n-i-1 = 3-2-1 = 0 \rightarrow \text{txt.charAt}(n-i-1) = \text{"O"}$

Dermed skrives følgende ut: **ELO**

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";` Lovlig `double`
- b) `double verdi2 = "String";` Ulovlig
- c) `double vekt1 = 5/2;` Lovlig `2`
- d) `double vekt2 = 5.0/2.0;` Lovlig `2.5`
- e) `int vekt3 = 5.0/2.0;` Ulovlig
- f) `double resultat = 5.2 + 14.6 * 3.7 + resultat;` Ulovlig
- g) `String svar1 = "2 + 3.14 = " + (2 + 3.14);` Lovlig `2 + 3.14 = 5.14`
- h) `boolean svar2 = true && false;` Lovlig `false`
- i) `boolean svar3 = true || false;` Lovlig `true`
- j) `boolean svar4 = (true && false) || (true || false);` Lovlig `true`

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");  
    }  
}
```

Svar: $5 * 20 = 100$ ganger

Oppgave 3 (9 poeng)

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");  
    }  
}
```

Svar: Når $k=0$ skrives "Arne" ut 20 ganger.

Når $k=1$ skrives "Arne" ut 19 ganger.

....

Når $k=4$ skrives "Arne" ut 16 ganger.

Totalt antall ganger: $16+17+\dots+20 = 90$

Oppgave 3 (9 poeng)

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");  
    }  
}
```

Svar: Når $k=0$ skrives "Arne" ut 5 ganger.

Når $k=1$ skrives "Arne" ut 4 ganger.

....

Når $k=4$ skrives "Arne" ut 1 gang.

Når $k>4$ skrives "Arne" ut 0 ganger.

Totalt antall ganger: $1+2+3+4+5 = 15$

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.


```
String[] konkatener (String[] tab, String s) {  
    String nytab = new String[tab.length];  
    for (int i=0; i<tab.length; i++) {  
        nytab[i] = tab[i] + s;  
    }  
    return nytab;  
}
```

Mulig tillegg: teste om `tab == null` og isåfall returnere, f.eks. ved å legge inn setningen `if (tab==null) return null;` øverst i metoden.

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 tilfredsstillter 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.

Hvordan går vi frem her?

Alternativ A:

- Vi lager oss først en liste over hvilke ulike verdier som forekommer i den opprinnelige arrayen.
- Så kan vi for hvert element i listen telle opp hvor mange ganger den forekommer i den opprinnelige arrayen.

Alternativ B:

- Vi starter med å anta at det er verdien øverst i arrayen som forekommer flest ganger (maxAnt ganger)
- Vi går gjennom arrayen
- For hvert element, teller vi hvor mange ganger det forekommer derfra og nedover
- Hvis antallet overstiger maxAnt, oppdaterer vi maxAnt og husker på hvilken verdi det var

```
int finnMode (int [] a) {
    int verdi, antall;
    int maxVerdi, int maxAnt = 0;

    for (int i=1; i<a.length; i++) {
        verdi =a[i-1];
        ant =1;
        // går gjennom resten av arrayen
        for (int j=i; j<a.length; j++) {
            if (a[j] == verdi) antall++;
        }
        if (antall > maxAnt) {
            maxAnt = antall;
            maxVerdi = verdi;
        }
    }
    return maxVerdi;
}
```

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.

```
void Oppgave6 () {
    In fil = new In("Ord.txt");
    HashMap<String,String> h1 =
        new HashMap<String,String> ();
    HashMap<String,String> h2 =
        new HashMap<String,String> ();

    while (fil.hasNext()) {
        String ord = fil.next();
        if(ord.length() == 1) h1.put(ord,ord);
        else if(ord.length() == 2) h2.put(ord,ord);
    }

    fil.close();

    System.out.println("Ord av lengde 1:" + h1.size());
    System.out.println("Ord av lengde 2:" + h2.size());
}
```

Oppgave 7 (8 poeng)

a) Skriv binærtallet 1110 som et desimaltall.

Svar: $1110 = 14$

b) Skriv desimaltallet 63 som et binærtall.

Svar: $63 = 111111$

c) Skriv binærtallet 11001 som en sum av to binærtall (hvor begge større enn binærtallet 10).

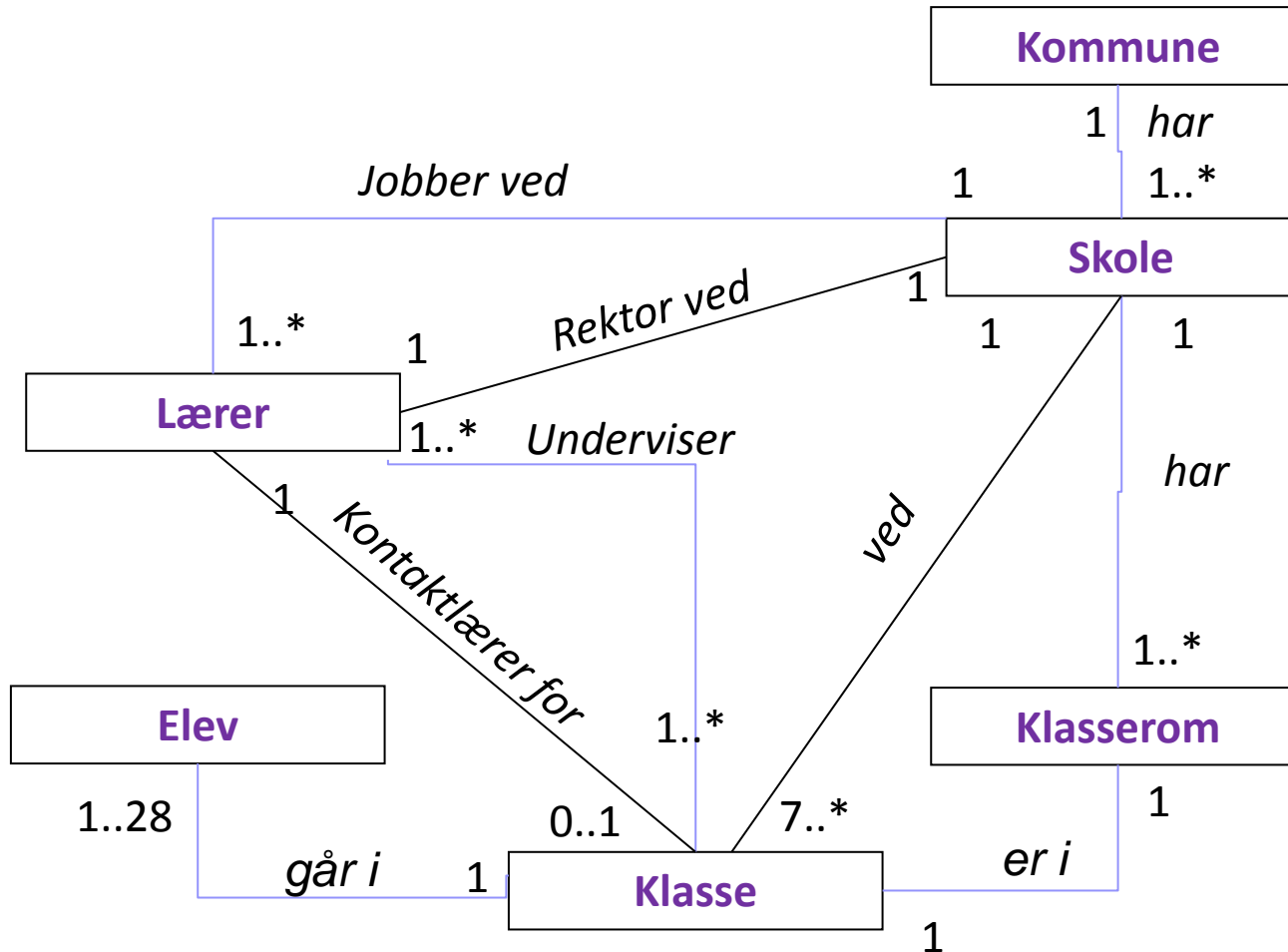
Svar: $11001 = 10000 + 1001$

d) Skriv desimaltall 20 som et heksadesimalt tall.

Svar: $20 \text{ (des)} = 14 \text{ (hex)}$

Oppgave 8

a)



b)

```
class Kommune {  
    String navn;  
    int aar;  
    int antSkoler;  
    Skole[ ] skolene;  
}
```

b)

```
class Skole {  
    String navn;  
    int antKlasserom;  
    Kommune minKommune;  
    Larer rektor;  
    Larer[] larere ;  
    Klasse [] klassene ;  
}
```

b)

```
class Klasserom {  
    String romNavn;  
    Klasse klasseHer;  
}
```

b)

```
class Klasse {  
    String navn;  
    Larer kontaktL;  
    Elev [] elevene;  
}
```

b)

```
class Elev {  
    String navn;  
    Klasse minKlasse;  
}
```

b)

```
class Lærer {  
    String navn;  
    Klasse kontakt = null;  
    Klasse[] mineKlasser;  
    Skole minSkole;  
}
```

c)

```
Kommune (String fil) {
    In f = new In(fil);
    navn = f.next();
    aar = f.inInt();
    antSkoler = f.inInt();
    skolene = new Skole[antSkoler];
    int i =0;
    for (i = 0; f.hasNext(); i++) {
        String navn = f.inWord(",");
        int antRom = f.inInt();
        skolene[i] = new Skole(navn, antRom, this);
    }
}
```



```
Klasse(Larer kl, String n, int antElever) {  
    elevene = new Elev[antElever];  
    navn = n;  
    kontaktL = kl;  
}
```

```
Elev(Klasse mK, String n) {  
    minKlasse = mK;  
    navn = n;  
}
```

```
Skole(String n, int ak, Kommune kom) {
    navn = n;
    antKlasserom = ak;
    minKommune = kom;

    String filN = minKommune.aar + navn + ".txt";
    In s = new In(filN);
    int antElever = s.inInt(",");
    rektor = new Larer(s.inWord(", "), this);
    int antLarere = s.inInt(",");
    larere = new Larer[antLarere];
    for (int i=0; i<antLarere; i++) {
        larere[i] = new Larer(s.inWord(", "), this);
    }
    int antKlasser = s.inInt();
    klassene = new Klasse[antKlasser];

    <.... se neste slide....>
}
```

```
for (int k = antKlasser-1 ; k >=0; k--) {
    int trinn = s.inInt() +1; // neste år
    char tegn = s.inChar();
    Larer kLarer = new Larer( s.inWord(",") ,this);
    int eAnt = s.inInt(",");

    if (trinn == 8) {
        for (int i = 0; i<eAnt; i++){
            s.readLine();
        }
    } else if (trinn > 1) {
        String kNavn = ""+trinn+tegn;
        klassene[k] =
            new Klasse(kLarer, ""+trinn+tegn, eAnt);
        for (int e=0; e<eAnt; e++) {
            klassene[k].elevene[e] = new
                Elev(klassene[k],s.inLine());
        }
    }
}
```

Oppgave 9

- a) Dette er klart ulovlig etter §11, at data innsamlet til ett formål nyttes til et annet formål. Elevene og foresatte har ikke gitt sitt samtykke (§8), og er heller ikke informert. Det er opplagt at salg av navnelister ikke er faglig nødvendig for å drive en barneskole (§8a,c,e eller f)

- b) Det nytter ikke å søke datatilsynet om lov (løyve) her - dette dreier seg ikke om sensitiv informasjon. Hvis derimot rektor spør hver elev og foresatte (§11c), kan han selge slike navnelister over de som skriftelig gir samtykke til det (foresatte), ikke de som enten ikke svarer eller sier nei.