Arne Maus, 10 des. 2012

INF1000-h2012-løsningsforslag

Oppgave 1 (2 poeng)

a) 10 b) 20

Oppgave 2 (8 poeng)

Riktige: a, b, c, f, h

Oppgave 3 (10 poeng)

N.B. To godkjente svar på e)

a) 16

b) 16

c) 9

d) 2

e) 3 – eller 0 (=kjører ikke, kompilereingsfeil – ekstrs parentes) // RETTELSE HER

Oppgave 4 (20 poeng)

```
double [] bmiBeregning (double[] hoyde, int [] vekt) {
      double[] bmi = new double[hoyde.length];
      for (int i=0; i < bmi.length; i++)
        bmi[i] = vekt[i]/(hoyde[i] * hoyde[i]);
       return bmi;
   }
```

Oppgave 5 (10 poeng)

Kan løses på mange måter:

```
double median1 (double a, double b, double c) {
      if ((a < b && b < c) ||(c < b && b < a) )return b;
      if ((b < a && a < c)|| (c < a && a < b) )return a;
      return c;
}
double median2 (double a, double b, double c) {
         if ( a < b) {
             if ( b < c) return b;
```

```
            else if (c < a) return a;else return c;
        } else {
            // vet nå b < a
            if (c < b) return b;
            else if (a < c) return a; else return c;
        }
    } // end median2

 double median3 (double a, double b, double c) {
        <legg a,b og c inn i en int [] num>
        < sorter num >
        < returner num [1]>
    } // end median2
```

Oppgave 6 (8 poeng) pos = 3, t= cdbabaabc

Oppgave 7 (9 poeng)|

a) 10, b) 100101 c) C7 // a) RETTET

Oppgave 8 (20 poeng)

Her godkjennes selvsagt at det er gjentagelsestegn på like høyresider i PettersLego-konstruktøren, og noen laget en formel for lengen som funksjon av antall knotter (også OK)

```
class PettersLego {
   Brikke[] mineBrikker = new Brikke[6];
   double h = 1.0, b= 1.6;
   PettersLego(){
      mineBrikker[0] = new Brikke(h,b,3.2,8);
      mineBrikker[1] = new Brikke(h,b,3.2,8);
      mineBrikker[2] = new Brikke(h,b,3.2,8);
      mineBrikker[3] = new Brikke(h,b,1.6,4);
      mineBrikker[4] = new Brikke(h,b,3.2,4);
      mineBrikker[5] = new Brikke(h,b,0.8,2);
   }

   double sumLengde() {
      double len=0.0;
      for (int i = 0; i < 6; i++)
      len+=mineBrikker[i].lengde;
      return len;
   }

}
```

```
class Brikke {
   double høyde, bredde, lengde;
   int antallKnotter;

   Brikke(double høyde, double bredde, double lengde, int antallKnotter) {
      this.høyde = høyde;
      this.bredde = bredde;
      this.lengde = lengde;
      this.antallKnotter = antallKnotter;
   }
}
```

Oppgave 9 (20 poeng)

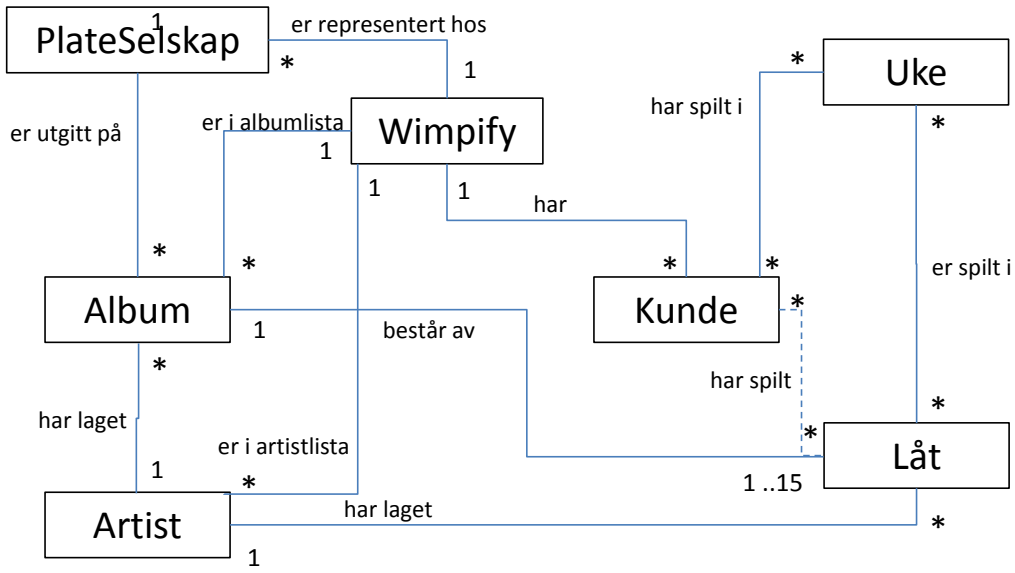
Merk at dette virker enten lengden er partall eller oddetall (for da skal midt-elementet ikke flyttes).

```
int [] snuArray(int [] a) {
        int temp;
        for (int i =0; i < a.length/2 ; i++) {
           temp = a[i];
           a[i] = a[a.length-i-1];
           a[a.length-i-1] = temp;
        }
       return a;
    }
```

Oppgave 10 (20 poeng)

Her er det egentlig naturlig å tegne en forbindelse mellom Kunde og Låt, men i programmet i neste oppgave er denne forbindelsen representert som Kunde->Uke->Låt (for å skrive regninger).

Merk også at det i programmet så er det ikke datastrukturer for alle mange-til-mange forbindelsene begge veier(for eksempel vet ikke Låt alle Ukene den er spilt – fordi vi ikke trenger det).



Oppgave 11 (50 poeng + 30 poeng for oppg. d))

Merk at i programmet trenger hver Kunde sitt eget Uke-objekt for et bestemt ukenummer og det igjen har et Låt-objekt for hver Låt som er spilt den uka (og spilles samme Låt en annen uke, blir det et nytt Låt-objekt for samme låt for denne kunden). Dette er vel egentlige vanskelige punktet i oppgaven.

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

class Oppgave11 {
   public static void main(String [] args) {
         new Wimpify().test();
    }
} // end Oppgave11

class Uke {
   int ukeNummer;
   HashMap <String, Lat> spilteLåter = new HashMap <String, Lat> ();

   Uke(int ukeNummer) {
      this.ukeNummer= ukeNummer;
   }
 }// end Uke

 class Lat{
    String låtNavn, artistNavn;
    int gangerSpilt=0;

    Lat(String navn, String artist) {
       låtNavn=navn;
       artistNavn=artist;
```

```
   }
} // end Lat (Låt)

class PlateSelskap{
   String navn;
   int bKontoNum;
   PlateSelskap(String navn, int bKontoNum) {
      this.navn = navn;
      this.bKontoNum= bKontoNum;
   }
 } // end PlateSelskap


class Wimpify{
   double prisKundeSpiltEnLåt =0.18,prisTiPlateselskapSpiltEnLåt=0.1;

   HashMap <String, Kunde>       kundeListe = new HashMap <String, Kunde> ();
   HashMap <String, PlateSelskap> selskaper  = new HashMap <String, PlateSelskap> ();
   HashMap <String, Album>       albumListe = new HashMap <String, Album> ();
   HashMap <String, Artist>      artistListe= new HashMap <String, Artist> ();

   In kunder = new In ("kunde.txt");
   In album  = new In ("album.txt");
   In avspilling = new In("avspillinger.txt");

   void test() {
     // oppg a)
     selskaper.put("EMI",   new PlateSelskap("EMI",123));
     selskaper.put("NAXOS", new PlateSelskap("NAXOS",456));
     selskaper.put("Grappa",new PlateSelskap("Grappa",789));

     // oppg b)
     lesKunder();

     // oppg c)
     lesAlbum();

     // oppg d
     lesAvspillinger();

   } // end utfør

   void lesKunder() {
      while (! kunder.endOfFile()) {
           Kunde k = new Kunde(kunder.inWord(";").trim(), kunder.inWord(";").trim());
           kundeListe.put(k.navn, k);
     }
   } // end lesKunder : oppg b


   void lesAlbum() {
      while (! album.endOfFile()) {
```

```
      String navn  = album.inWord(";").trim();
      String art   = album.inWord(";").trim();
      String selsk = album.inWord(";").trim();
      int ant      = album.inInt(" ;");
      Album a = new Album (navn,selsk, art);
      albumListe.put(navn,a);

      for (int i = 1; i <= ant; i++) {
         a.låter[a.antLåter++] = new Lat(album.inWord(";").trim(), art);
       }

       // legg in evt. ny Artist hos Arist og plateselskap
       Artist ar = artistListe.get(art);
       if(ar == null) {
          ar = new Artist (art);
          artistListe.put(ar);      // korreksjon
       }

       ar.mineAlbum.put(navn,a);

    } // end while album,txt
} // end lesAlbum: oppgc

void lesAvspillinger() {
   while (! avspilling.endOfFile()) {
       String kunde = avspilling.inWord(";").trim();
       int uke      = avspilling.inInt(" ;");
       String låt   = avspilling.inWord(";").trim();
       String album = avspilling.inWord(";").trim();

       Kunde k = kundeListe.get(kunde);
       Uke u = k.spilteUker.get(""+uke);

       if (u== null) {
          u = new Uke(uke);
          k.spilteUker.put(""+uke, u);
        }
       Lat l = u.spilteLåter.get(låt);
       if (l==null){
          l = new Lat(låt,albumListe.get(album).albumNavn);
          u.spilteLåter.put(låt,l);
       }
       l.gangerSpilt++;
    } // end les fil


    // les Kundene  og skriv regning
    for (Kunde k: kundeListe.values()) {

       for (Uke u: k.spilteUker.values()){
          double sumRegning =0.0;
```

```
            for (Lat l: u.spilteLåter.values()){
               sumRegning += l.gangerSpilt*prisKundeSpiltEnLåt;
            }
            k.sendEpost(k.navn, k.ePostadresse,u.ukeNummer, sumRegning);
          }
       } // end send regninger
   }// end lesAvspillinger: oppg d

} // end Wimpify

class Album{
  String albumNavn, plateSelskapsNavn,  artistNavn ;
  Album (String aln, String ps, String art) {
      albumNavn= aln;
      plateSelskapsNavn= ps;
      artistNavn= art;
   }

  Lat [] låter = new Lat [15];
  int antLåter =0;
} // end Album

class Kunde {
   String navn, ePostadresse ;
   HashMap <String, Liste> mineLister = new HashMap <String,Liste>();
   HashMap <String, Uke> spilteUker = new HashMap <String, Uke>();

   void sendEpost (String kundeNavn, String epostAdresse, int ukeNummer, double beløp) {
    System.out.println(" kundeNavn;"+kundeNavn+", epostAdresse:"+epostAdresse+
                    ", ukeNummer:"+ukeNummer+", beløp:"+beløp);
    }

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

class Artist{
   String navn;
   HashMap <String,Album> mineAlbum = new HashMap<String,Album>();
   Artist (String navn) {
      this.navn = navn;
   }
} // end Artist
```

Oppgave 12 (30 poeng)

a) Er ikke tillatt av flg. grunner:
   Forslag 1:
   - 2nr.8 – registrering av sensitive data (etnisk opphav) – trenger godkjenning fra datatilsynet etter søknad (§ 33)
   - §11 – nyttes vel til et annet formål enn navn ble samlet inn for .

   Forslag2:
   Det at andres navn kommer fram på dine web-sider er ikke tillatt (med mindre alle har godkjente dette da de tegnet kontrakt med Wimpify). §11 – navnene til Kunder brukes til et annet forhold enn det man kunne forvente da det ble samlet inn.
b) Forslag 2 kan fikses ved enten at de underskriver det når de tegner kontrakt, eller (bedre) at det bare opplyses for deg hvilket nummer du er på 'ukas streamer'-liste, hvor mange streaminger du har og hvor mange bestemann har (evt. hele lista på 10 beste) og hvor alle de andre er anonymisert som : <bruker nr i> har xxx streaminger denne uka for de 10 beste , og bare ditt navn kommer fram i klartekst.
c) Harald må søke om idé nr. 1 (og vill få avslag) på bevilling om det – han må nok enten sende Bollywood-filmtilbudene sine til alle kundene eller be kundene om å registrere seg selv som interessert i slike tilbud.