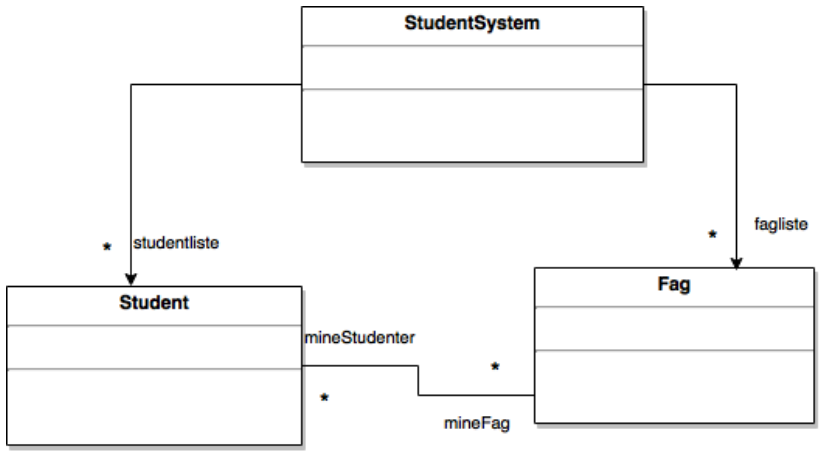
INF1000 - Løsning på seminaroppgaver til uke 9

Oppgave 1

UML-diagram:



Main

```
import java.util.Scanner;

public class Main{
  public static void main(String [] args) throws Exception{
    StudentSystem studentSys = new StudentSystem();
    studentSys.lesFraFil("emnestudenter.txt");
    studentSys.ordrelokke();
    studentSys.skrivTilFil("emnestudenter.txt");
  }
}
```

Fag

```
import java.util.HashMap;

public class Fag{
  private String fagkode;
  private HashMap<String, Student> mineStudenter = new HashMap<String, Student>();

  public Fag(String fagkode){
    this.fagkode = fagkode;
  }

  public void leggTilStudent(Student student){
    if(finnesStudent(student)){
      System.out.println(student.toString() + " tar allerede " + fagkode);
    }
    mineStudenter.put(student.toString(), student);
  }

  public String toString(){
    return this.fagkode;
  }

  public void fjernMegFraStudenter(){
    for(Student studentSomTarFaget : mineStudenter.values()){
      studentSomTarFaget.fjernFagFraStudent(this);
    }
  }

  public HashMap<String, Student> listeAvStudenter(){
    return mineStudenter;
  }

  public void skrivUtMineStudenter(){
    for(Student student : mineStudenter.values()){
      System.out.println(student.toString());
    }
  }

  public boolean finnesStudent(Student student){
    if(mineStudenter.containsValue(student)){
      return true;
    } return false;
  }

  public void fjernStudent(Student student){
```

```
    if(finnesStudent(student)){
      mineStudenter.remove(student);
    } else {
      System.out.println(student.toString() + " har ikke meldt seg opp i " + fagkode);
    }
  }

  public int antStudenter(){
    return mineStudenter.size();
  }
}
```

Student

```
import java.util.HashMap;

public class Student{
  private String navn;
  private HashMap<String, Fag> mineFag = new HashMap<String, Fag>();

  public Student(String navn){
    this.navn = navn;
  }

  public void leggTilFag(Fag fag){
    if(tarFag(fag)){
      System.out.println(navn + " tar allerede " + fag.toString());
      return;
    }
    mineFag.put(fag.toString(), fag);
  }

  public String toString(){
    return navn;
  }

  public void skrivUtMineFag(){
    for(Fag etFag : mineFag.values()){
      System.out.println(etFag.toString());
    }
  }

  public boolean tarFag(Fag fag){
    if(mineFag.containsValue(fag)){
      return true;
```

```
    } return false;
  }

  public void meldMegAvAlleFag(){
    for(Fag fagJegTar : mineFag.values()){
      fagJegTar.fjernStudent(this);
    }
  }

  public void fjernFagFraStudent(Fag fag){
    if(!tarFag(fag)){
      System.out.println(navn + " har aldri vaert meldt opp i " + fag.toString());
      return;
    }
    mineFag.remove(fag);
  }

  public int antFag(){
    return mineFag.size();
  }
}
```

StudentSystem

```
import java.util.HashMap;
import java.io.File;
import java.util.Scanner;
import java.io.PrintWriter;

public class StudentSystem{

  private HashMap<String, Fag> fagliste = new HashMap<String, Fag>();
  private HashMap<String, Student> studentliste = new HashMap<String, Student>();
  private Scanner tastatur = new Scanner(System.in);

  public void lesFraFil(String filnavn) throws Exception{
    Scanner fil = new Scanner(new File(filnavn));
    String linje;
    Fag fag = null;
    while(fil.hasNextLine()){
      linje = fil.nextLine();
      //Sjekker om linje er fagkode ved forekomst av *
      if(linje.charAt(0) == '*'){
        //Oppretter nytt fag. Fjerner * ved bruk av substring.
        fag = new Fag(linje.substring(1));
        fagliste.put(fag.toString(), fag);
```

```
    } else {
      Student student;
      //Sjekker om student finnes.
      if(studentliste.containsKey(linje)){
        student = studentliste.get(linje);
      } else {
        student = new Student(linje);
        studentliste.put(student.toString(), student);
      }
      student.leggTilFag(fag);
      fag.leggTilStudent(student);
    }
  }
}

public void meny(){
  System.out.println("************STUDENTSYSTEM**********");
  System.out.println("1: Legg til ny student.");
  System.out.println("2: Legg til nytt fag.");
  System.out.println("3: Skriv ut alle studenter som tar et spesifikt fag.");
  System.out.println("4: Skriv ut alle fag en spesifikk student tar");
  System.out.println("5: Meld opp en student til et emne.");
  System.out.println("6: Meld av en student til et emne.");
  System.out.println("7: Fjern en student fra systemet.");
  System.out.println("8: Fjern et fag fra systemet.");
  System.out.println("9: Finne ut hvilket fag som blir tatt av flest studenter.");
  System.out.println("10: Finne ut hvilken student som tar flest fag.");
  System.out.println("11: Skriv ut alle fag og hvilke studenter som tar de.");
  System.out.println("0: Avslutt.");
}


public void ordrelokke(){
  int inputFraBruker = -1;

  while(inputFraBruker != 0){
    if(inputFraBruker == 1){
      leggTilNyStudent();
    } else if(inputFraBruker == 2){
      leggTilNyttFag();
    } else if(inputFraBruker == 3){
      skrivUtAlleStudenterSomTarFag();
    } else if(inputFraBruker == 4){
      skrivUtAlleFagSomStudentTar();
    } else if(inputFraBruker == 5){
      meldOppStudentTilEmne();
```

```
    } else if(inputFraBruker == 6){
      meldAvStudentTilEmne();
    } else if(inputFraBruker == 7){
      slettStudent();
    } else if(inputFraBruker == 8){
      slettFag();
    } else if(inputFraBruker == 9){
      finnMestPopulaereFag();
    } else if(inputFraBruker == 10){
      finnMestArbeidsommeStudent();
    } else if(inputFraBruker == 11){
      skrivUtAlleFagMedTilhorendeStudenter();
    }
    meny();
    inputFraBruker = Integer.parseInt(tastatur.nextLine());
  }

}

public void skrivUtAlleFagMedTilhorendeStudenter(){
  for(Fag fag : fagliste.values()){
    System.out.println("Fag: " + fag.toString());
    HashMap<String, Student> listeAvStudenterSomTarFag = fag.listeAvStudenter();
    for(Student studentSomTarFag : listeAvStudenterSomTarFag.values()){
      System.out.println(studentSomTarFag.toString());
    }
  }
}

public void skrivTilFil(String filnavn) throws Exception{
  PrintWriter skriver = new PrintWriter(filnavn);
  for(Fag fag : fagliste.values()){
    skriver.println("*" + fag.toString());
    HashMap<String, Student> listeAvStudenterSomTarFag = fag.listeAvStudenter();
    for(Student studentSomTarFag : listeAvStudenterSomTarFag.values()){
      skriver.println(studentSomTarFag.toString());
    }
  }
  skriver.close();
}

public void leggTilNyStudent(){
  System.out.println("Hva heter studenten?");
  String navn = tastatur.nextLine();
  studentliste.put(navn, new Student(navn));
```

```
}

public void leggTilNyttFag(){
  System.out.println("Hva heter faget?");
  String fagkode = tastatur.nextLine();
  fagliste.put(fagkode, new Fag(fagkode));
}

public void skrivUtAlleStudenterSomTarFag(){
  System.out.println("Hvilket fag vil du se alle studenter til?");
  Fag fag = fagliste.get(tastatur.nextLine());
  fag.skrivUtMineStudenter();
}

public void skrivUtAlleFagSomStudentTar(){
  System.out.println("Hvilken student vil du se alle fagene til?");
  Student student = studentliste.get(tastatur.nextLine());
  student.skrivUtMineFag();
}

public void meldOppStudentTilEmne(){
  System.out.println("Hvilken student vil du melde opp?");
  Student student = studentliste.get(tastatur.nextLine());

  System.out.println("Hvilket fag vil du melde opp til?");
  Fag fag = fagliste.get(tastatur.nextLine());

  fag.leggTilStudent(student);
  student.leggTilFag(fag);
}

public void meldAvStudentTilEmne(){
  System.out.println("Hvilken student vil du melde av?");
  Student student = studentliste.get(tastatur.nextLine());

  System.out.println("Hvilket fag vil du melde av til?");
  Fag fag = fagliste.get(tastatur.nextLine());

  fag.fjernStudent(student);
  student.fjernFagFraStudent(fag);

}

public void slettStudent(){
  System.out.println("Hvilken student vil du fjerne?");
  Student student = studentliste.get(tastatur.nextLine());
```

```
    student.meldMegAvAlleFag();
    studentliste.remove(student);
  }

  public void slettFag(){
    System.out.println("Hvilket fag vil du fjerne?");
    Fag fag = fagliste.get(tastatur.nextLine());

    fag.fjernMegFraStudenter();
    fagliste.remove(fag);
  }

  public void finnMestPopulaereFag(){
    Fag fagMedMestFlestStudenterHittil = null;

    for(Fag fag : fagliste.values()){
      if(fagMedMestFlestStudenterHittil == null){
        fagMedMestFlestStudenterHittil = fag;
      } else {
        if(fagMedMestFlestStudenterHittil.antStudenter() < fag.antStudenter()){
          fagMedMestFlestStudenterHittil = fag;
        }
      }
    }

    System.out.println("Mest populaere fag er: " + fagMedMestFlestStudenterHittil.toString()
  }

  public void finnMestArbeidsommeStudent(){
    Student studentMedMestFlestFagHittil = null;

    for(Student student : studentliste.values()){
      if(studentMedMestFlestFagHittil == null){
        studentMedMestFlestFagHittil = student;
      } else {
        if(studentMedMestFlestFagHittil.antFag() < student.antFag()){
          studentMedMestFlestFagHittil = student;
        }
      }
    }
    System.out.println("Mest arbeidsomme student er: " + studentMedMestFlestFagHittil.toStri
  }
}
```