

## Student.java

```
1 /
   *****
   *****
2  * @author Øystein
3  * Student.class
4  *
5  * Klassen representerer en student som en del av en gruppe i
6  * Studentadministrasjonsprogrammet.
7  *
8  * Klassen skal inneholde, sette og svare på
9  * 1.Studentens navn
10 * 2.Studentens studiestart
11 * 3.Studentens kjønn
12 * 4.Studentens oppgaver (dypere informasjon om oppgavene lagres
13 * i et eget oppgave-objekt).
14 * 5.Studentens status i forhold til om alle oppgaver er godkjente,
   og
15 * studenten er klar til eksamen.
16 * 6.Studentnummer (vi tar ikke denne i bruk, men legger den ved da
   vi
17 * kan se en fremtidig nytte av dette; skille studenter med likt
   navn og
18 * samkjøring mot andre skolesystemer).
19 * 7.Studentens fagområde
20
   *****
   *****/
21 import java.io.Serializable;
27
28 public class Student implements Serializable{
29
30     //Studentens fornavn
31     private String fornavn = "";
32     //Studentens etternavn
33     private String etternavn = "";
34     //Studentens fagområde (f.eks. Matematikk)
35     private String fagomrade = "";
36     /*Variabel for kjønn; true = mann,
37     false = kvinne (tilfeldig valgt, trenger ikke tolkes... :)/
38     private boolean mann = true;
39     //Studentens studiestart f.eks. 2014
40     private int studiestart = 2014;
41     //Studentnummer f.eks. s123456789
42     private String studentnummer = "";
43     //En vektor som inneholder alle studentens oppgaver.
```

## Student.java

```
44     private Vector<Oppgave> oppgaver = new Vector();
45
46
47     /*En tom konstruktør i tilfellet man trenger å
48     opprette et studentobjekt før en har data til å fylle det.*/
49     public Student(){
50
51     }
52
53     //Konstruktør for å opprette en ny student med alle verdier
    satt.
54     public Student(String forNavn,
55                     String etterNavn,
56                     int studieStart,
57                     boolean kjonnMann,
58                     String fagOmrade,
59                     Vector<Oppgave> oppgaver){
60         this.fornavn = forNavn;
61         this.etternavn = etterNavn;
62         this.studiestart = studieStart;
63         this.mann = kjonnMann;
64         this.fagomrade = fagOmrade;
65         this.oppgaver = oppgaver;
66     }
67
68
69     //Returnerer studentens fornavn
70     public String getFornavn(){
71         return fornavn;
72     }
73
74     //Setter eller endrer studentens fornavn
75     public void setFornavn(String forNavn){
76         this.fornavn = forNavn;
77     }
78
79     //Returnerer studentens etternavn
80     public String getEtternavn(){
81         return etternavn;
82     }
83
84     //Setter eller endrer studentens etternavn
85     public void setEtternavn(String etterNavn){
86         this.etternavn = etterNavn;
87     }
```

## Student.java

```
88
89
90  /*Returnerer studentens hele navn dersom metoden blir kalt.
91  Dette skjer f.eks. ved at man System.out.println
    studentobjektet,
92  eller ved at man legger objektet i en liste. Med andre ord,
93  denne metoden bestemmer hvordan objektet vises i en liste.*/
94  public String toString(){
95      return fornavn + " " + etternavn;
96  }
97
98  //Setter eller endrer studentens fagområde
99  public void setFag(String fag){
100      this.fagomrade = fag;
101  }
102
103  //Returnerer studentens fagområdet
104  public String getFag(){
105      return fagomrade;
106  }
107
108  //Setter eller endrer studentens studiestart
109  public void setStudiestart(String studieStart){
110      try{
111          this.studiestart = Integer.parseInt(studieStart);
112      }catch (Exception e){
113          System.out.println(
114              "Feil ved lesing av studiestart. "
115              + "Setter studiestart til gjeldene \u00E5r!" +
    e);
116          //Ved feil, sett årstall til innenvêrende år
117          GregorianCalendar gc = new GregorianCalendar();
118          this.studiestart = gc.get( Calendar.YEAR);
119      }
120  }
121
122  //Returnerer studentens studiestart
123  public int getStudiestart(){
124      return studiestart;
125  }
126
127  //Returnerer true hvis mann, false hvis dame.
128  public boolean isMann(){
129      return mann;
130  }
```

## Student.java

```
131
132  /*Returnerer true hvis dame, false hvis mann
133  (denne metoden er laget i likestillingens navn!)*//
134  public boolean isDame(){
135      return !mann;
136  }
137
138  //Returnerer kjønnet som en streng.
139  public String getKjonn(){
140      if (mann) return "Mann"; else return "Kvinne";
141  }
142
143  //Setter eller endrer studentens kjønn
144  public void setKjonn(boolean mann){
145      this.mann = mann;
146  }
147
148  //Setter eller endrer studentens fornavn
149  public void setStudentnummer(String studentnummer){
150      this.studentnummer = studentnummer;
151  }
152
153  //Returnerer studentnummer
154  public String getStudentnummer(){
155      return studentnummer;
156  }
157
158  //Legger til en oppgave (et oppgaveobjekt) i studentens
    oppgaveliste.
159  public void addOppgave(Oppgave opg){
160      oppgaver.addElement(opg);
161  }
162
163  //Returnerer alle studentens oppgaver som en array av
    oppgaveobjekter.
164  public Oppgave[] getOppgaver(){ //Returnerer alle oppgavene som
    en array.
165      Oppgave[] opg = new Oppgave[oppgaver.size()];
166      oppgaver.toArray(opg);
167      return opg;
168  }
169
170  //Returnerer alle studentens oppgaver som en Vector
171  public Vector<Oppgave> getOppgaverAsVector(){
172      return oppgaver;
```

## Student.java

```
173     }
174
175     /*Kontrollerer hver eneste oppgave studenten har, og
176     returnerer true hvis alle er godkjente (altså at studenten kan
177     ta eksamen)
178     og false hvis ikke.*/
179     public boolean isGodkjent(){
180         /*Gå gjennom alle oppgaver, dersom en ikke er godkjent:
181         returner
182         false eller returner true*/
183         for (int i = 0; i < oppgaver.size(); i++){
184             if (oppgaver.elementAt(i).isGodkjent())
185                 continue;
186             else
187                 return false;
188         }
189
190         //returner FALSE dersom studenten ennå ikke har gjort noen
191         oppgaver.
192         if (oppgaver.size() < 1)
193             return false;
194
195         return true;
196     }
197 }
```