

Filbehandling.java

```
1 /
   *****
   *****
2  * @author Øystein
3  * Filebehandling.class
4  *
5  * Klassen lagrer og leser et gruppeobjekt med studenter til fil
   med
6  * navn "gruppe.obj" ved å bruke Java sin serializable funksjon.
7  *****
   *****/
8  import java.io.BufferedReader;
17
18 public class Filbehandling
19 {
20
21     /*Filnavnet til gruppefila som blir lagret på harddisken
22     (statisk ettersom det ikke skal være flere tilfeller av
   denne.
23     final slik at den er fast og ikke endres).
24     Dette gjør det lett å endre filnavnet, uten å måtte gå gjennom
   hele koden.*/
25     public static final String FILNAVN = "gruppe.obj";
26
27
28     /*Metode for å laste gruppeobjektet, returnerer et Gruppe-
   objekt.
29     OBS! Hvis feil ved lagring, anta at filen er ødelagt eller ikke
30     eksisterer. Lag en tomt nytt gruppeobjekt! (Dette er ikke noe
31     produksjonsverdig løsning, men i forhold til kriteriene i dette
32     prosjektet så holder det).*/
33     public static Gruppe lastGruppe()
34     {
35         File f = new File(Filbehandling.FILNAVN);
36         Gruppe gruppe = null;
37
38         if (f.exists())
39         {
40             /*Filbehandling kjøres i en try-klamme, slik at feil
   kan
41             bli behandlet i stedet for å krøsjie programmet.*/
42             try
43             {
44                 /*Opprette en strøm fra filen, en buffer for stabil
   lesing,
```

Filbehandling.java

```
45         og en strøm for å ta imot objektet som blir lest.*/
46         FileInputStream fis = new
FileInputStream(Filbehandling.FILNAVN);
47         BufferedInputStream bis = new
BufferedInputStream(fis);
48         ObjectInputStream ois = new ObjectInputStream(bis);

49
50         //Leser objektet fra filen
51         Object obj = ois.readObject();
52
53         //Kontrollerer om objektet er av typen "Gruppe"
54         if (obj instanceof Gruppe)
55         {
56             gruppe = (Gruppe)obj;
57         }
58
59         //Lukker filen.
60         ois.close();
61         bis.close();
62         fis.close();
63     }
64     catch (ClassNotFoundException cnf)
65     {
66         /*Datafilen inneholder ikke klassen gruppe.
67         Start programmet med en ny, tom studentgruppe.*/
68         System.out.println(
69             "Fant ikke klassen ved lasting \n " +
70             cnf);
71         gruppe = new Gruppe();
72     }
73     catch (InvalidClassException ice)
74     {
75         /*Progammnet er ikke lenger kompatibelt med lagret
76         data.
77         Start programmet med en ny, tom studentgruppe.*/
78         System.out.println(
79             "Endringer siden sist! Ikke kompatibel med
80             gammel brukerdata.\n"
81             + "Lager ny brukerdata.");
82         gruppe = new Gruppe();
83     }
84     catch (IOException ioe)
85     {
86         /*Generelt problem med innlastning.
```

Filbehandling.java

```
85         Start programmet med en ny, tom studentgruppe.*/
86         System.out.println("Feil ved lasting \n" + ioe);
87         gruppe = new Gruppe();
88     }
89 }
90 else
91 {
92     /*Det eksisterer ikke en fil med gruppa.
93     Start programmet med en tom gruppe.*/
94     gruppe = new Gruppe();
95 }
96
97
98 //SKRIVER UT EN LISTE OVER ALLE LAGREDE STUDENTER HVER GANG
99 FILEN ≈PNES.
100 //Mest nyttig under utvikling, bare for å se at alt
101 stemmer :)
102 for (int i = 0; i < gruppe.hentStudenterAsVector().size();
103 i++){
104     System.out.println( gruppe.hentStudenterAsVector().elementAt(i));
105 }
106
107 return gruppe;
108 }
109
110 //Metode for å lagre et Gruppe-objekt, tar i mot et
111 gruppeobjekt.
112 public static void lagreGruppe(Gruppe grp)
113 {
114     /*Filbehandling kjøres i en try-klamme, slik at feil kan
115 bli
116 behandlet i stedet for å kræsje programmet.*/
117 try
118 {
119     /*Opprette en strøm til filen, en buffer for stabil
120 skrivning,
121 og en strøm for å sende objektet til filen.*/
122     FileOutputStream fos = new
123     FileOutputStream(Filbehandling.FILNAVN);
124     BufferedOutputStream bos = new
125     BufferedOutputStream(fos);
126     ObjectOutputStream oos = new ObjectOutputStream(bos);
```

Filbehandling.java

```
121         oos.writeObject(grp);
122
123         //Tømming av alle strømmen (for å hindre delevis
lagring og korrupte filer.
124         oos.flush();
125         oos.close();
126         bos.flush();
127         bos.close();
128         fos.flush();
129         fos.close();
130     }
131     catch (IOException ioe)
132     {
133         /*Gir brukeren beskjed om at lagringen ikke var
vellykket,
134         i tillegg vises java sin feilmelding.*/
135         System.out.println("Feil ved lagring! " + ioe);
136     }
137 }
138 }
```