

## Main.java

```
1  /*****
2  * @author AnneLine
3  * Denne klassen skal først og fremst starte opp en meny
4  * slik at brukeren får noen valg å gjøre. Valgene brukeren
5  * får er å legge til student, generere grupper, endre student
6  * eller legge til oppgaver, eller å avslutte programmet.
7  * Alt etter hva brukeren velger å gjøre i menyen så vil det
8  * bli kalt opp en metode som så starter, enten innad i klassen
9  * eller i andre klasser.
10 *
11 * Det første som skjer i denne klassen er at vi initialiserer
12 * en variabel som kalles gruppe, og setter den til null. Dette
13 * gjør vi fordi vi vil at denne variabelen skal være
14 * tilgjengeig i hele klassen. Etter dette kaller vi opp klassen
15 * Gruppe og setter verdier fra den inn i variabelen gruppe.
16 * Når dette er gjort så vises menyen til brukeren på skjermen.
17 *
18 * Når visMeny starter får brukeren fire valg og skal så trykke
19 * det tallet som tilsvarer det brukeren ønsker å gjøre. Her får
20 * brukeren en feilmelding om den skriver et annet tall som ikke
21 * finnes eller om den skriver bokstaver.
22 * Etter at brukeren har valgt sitt tall blir det gjort en sjekk
23 * som sender brukeren dit den ønsker, altså kaller opp ulike
24 * metoder innad i klassen eller i andre klasser.
25 *
26 * I metoden leggTilStudent blir opprettet en konstruktør
27 * student som viser til klassen Student. I tillegg blir det
28 * opprettet to variabler; en boolean og en int. Før det settes
29 * i gang en do/while-løkke som kjører så lenge brukeren velger
30 * å trykke ja på spørsmål om den vil lese inn flere studenter.
31 * Inne i denne do/while-løkken får brokredet beskjed om å legge
32 * til informasjon om studenter. Informasjon som skal legges
33 * inn er fornavn, etternavn, kjønn, studiestart og fagområde.
34 * Fornavn, etternavn, kjønn og studiestart er input der
35 * brukeren kan bruke fritekst. Dette blir sjekket for noen
36 * feil, men til forbedring av programmet så kunne man ha
37 * sjekket om det ble lagt inn stor bokstav i navnene. På
38 * fagområde inputen skal brukeren velge mellom tre ulike fag og
39 * skrive inn tallet som stemmer overens med det faget studenten
40 * har. Dette gjorde vi fordi vi så at det ble for omfattende
41 * sjekker om brukeren skulle få lov til å skrive inn fritekst.
42 * Til senere forbedring kan det bli sett på hvordan man skal
43 * gjøre disse sjekkene slik at man kan bruke informasjonen fra
44 * brukeren her til å dele inn i grupper senere. Etter at vi har
45 * fått all informasjonen fra brukeren skal vi plassere dette
```

## Main.java

```
46 * inn i klassen Student. Vi sender dermed en og en
47 * informasjonsbolk inn i klassen Student og til de ulike
48 * set.metodene. Når dette er gjort kaller vi på klassen Gruppe
49 * og sender inn informasjonsbolkene om studenten inn der til
50 * leggTilStudent metoden. Helt til slutt kaller vi på klassen
51 * Filbehandling og metoden lagreGruppe og sender inn gruppe-
52 * objektene til denne klassen slik at de blir lagret.
53 *
54 * I metoden lagDiverseGrupper blir det laget en do/while-løkke
55 * som kjører så lenge brukeren velger å lage nye grupper, hvis
56 * ikke så vises start menyen på nytt. Valgene brukeren får i
57 * lagDiverseGrupper er 1. Sortere etter kjønn, 2. Sortere etter
58 * fag, 3. Sortere etter studiestart, 4. Sorter alfabetisk, 5.
59 * Lage tilfeldig sammensetning, 6. Lage liste over de som kan
60 * gå opp til eksamen og 0. avslutte programmet. Alt etter hva
61 * brukeren velger å gjøre, så blir den sendt videre til til
62 * klassen GenererGruppe klassen og forskjellige metoder innad i
63 * den klassen.
64 *****/
65 // importerer ulike klasser
66 import java.util.FormatterClosedException;
67
68
69
70 public class Main
71 {
72     /*Dette er en konstruktør som initialiserer
73     * variabelen gruppe og setter denne til null
74     * i første omgang, for at denne skal være
75     * tilgjengelig i hele klassen.*/
76     private Gruppe gruppe = null;
77
78     // sender inn verdier fra klassen Gruppe
79     public Main(Gruppe grp)
80     {
81         /*Setter verdiene fra klassen Gruppe til
82         * den lokale gruppevariablen slik at verdien
83         * i gruppe blir fylt opp med objekter.*/
84         gruppe = grp;
85
86         //Viser menyen (starter programmet)
87         visMeny();
88     }
89
90     public void visMeny()
91     {
92         // gir brukeren ulike valg som kan taes
```

## Main.java

```
93     String valg = JOptionPane.showInputDialog(
94         null,
95         "Skriv inn tallet p\u00E5 hva du vil gj
\u00F8re:\n"
96         + "1: Legg til student.\n"
97         + "2: Generer gruppe.\n"
98         + "3: Endre student eller legg til
oppgaver.\n"
99         + "0: Avslutt.");
100     /* sjekker om brukeren trykker p\u00E5 cancel og gir
101     mulighet til \u00E5 starte p\u00E5 nytt*/
102     if (valg == null)
103     {
104         System.exit(0);
105     }
106     if (valg.equals("1"))// hvis brukeren skriver 1 sl\u00E5r denne
til
107     {
108         // starter opp metoden aapneFil og leggTilStudent
109         leggTilStudent();
110     }
111     else if (valg.equals("2"))// hvis brukeren skriver 2 sl\u00E5r
dette til
112     {
113         // starter metoden lagDiverseGrupper-GenererGruppe fra
Ida
114         lagDiverseGrupper();
115     }
116     else if (valg.equals("3"))
117     {
118         // kaller opp vindus klassen GUIMain til \u00D8ystein
119         new GUIMain(gruppe, this);
120     }
121     else if (valg.equals("0"))// hvis brukeren skriver 0 s\u00E5
sl\u00E5r dette til
122     {
123         System.exit(0); // Avslutt
124     }
125     else/* sjekker om brukeren skriver inn noe feil eller
trykker
126         ok/kryss og gir brukeren mulighet til \u00E5 starte p\u00E5
nytt*/
127     {
128         int ja = JOptionPane.showOptionDialog(null, "Du
skrev ikke riktig,"
```

Main.java

```
129         + " pr\u00F8ve igjen?",
130         "Feil",
131         JOptionPane.YES_NO_OPTION,
132         JOptionPane.QUESTION_MESSAGE,
133         null,
134         null,
135         null);
136     if (ja == JOptionPane.YES_OPTION)
137     {
138         // start på nytt om bruker trykker ja
139         visMeny();
140     }
141     else
142     {
143         // avslutt om bruker trykker nei eller cancel/
144 kryss
145         System.exit(0);
146     }
147 }// slutt på else/if tester
148 }// slutt på visMeny-metode
149
150 public void leggTilStudent()
151 {
152     // oppretter en konstruktør til klassen Student
153     Student student = null;
154     // gjør mann tilgjengelig gjennom hele metoden
155     boolean mann = false;
156     // gjør svar tilgjengelig gjennom hele metoden
157     int svar;
158
159     // starter en løkke der brukeren kan skrive inn input
160     do
161     {
162         // kaller opp Studentklassen
163         student = new Student();
164         // fornavn fra brukeren
165         String elevFornavn = JOptionPane.showInputDialog(null,
166             "Skriv inn fornavnet p\u00E5 eleven:",
167             "Fornavn",
168             JOptionPane.QUESTION_MESSAGE);
169         /* sjekker om brukeren trykker cancel eller kryss
170         og avslutter programmet*/
171         if (elevFornavn == null || elevFornavn == "")
172         {
173             System.exit(0);
174         }
175     }
176 }
```

## Main.java

```
173     }
174     // hvis brukeren lar input stå tomt får sjans på nytt
175     if (elevFornavn.isEmpty())
176     {
177         do
178         {
179             elevFornavn = JOptionPane.showInputDialog(null,
180                 "M\u00E5 ha et fornavn, skriv inn p
\u00E5 nytt:");
181             // avslutt om det blir trykket cancel/kryss
182             if (elevFornavn == null || elevFornavn == "")
183             {
184                 System.exit(0);
185             }
186         }while (elevFornavn.isEmpty()); // fortsett så lenge
input er tom
187     }
188     // input fra brukeren
189     String elevEtternavn =
JOptionPane.showInputDialog(null,
190         "Skriv inn etternavn p\u00E5 eleven:",
191         "Etternavn",
192         JOptionPane.QUESTION_MESSAGE);
193     // sjekker om brukeren trykker cancel eller kryss
avslutt program
194     if (elevEtternavn == null || elevEtternavn == "")
195     {
196         System.exit(0);
197     }
198     // hvis brukeren lar input stå tomt får sjans på nytt
199     if (elevEtternavn.isEmpty())
200     {
201         do
202         {
203             elevEtternavn =
JOptionPane.showInputDialog(null,
204                 "M\u00E5 ha et etternavn, skriv inn p
\u00E5 nytt:");
205             // sjekker om bruker trykker cancel/kryss og
avslutter
206             if (elevEtternavn == null || elevEtternavn ==
"")
207             {
208                 System.exit(0);
209             }
```

## Main.java

```
210         }while (elevFornavn.isEmpty());// fortsett så lenge
input er tom
211     }
212     // input fra brukeren
213     String elevKjonn = JOptionPane.showInputDialog(null,
214         "Skriv inn hvilket kj\u00F8nn, mann eller
dame?",
215         "Kj\u00F8nn",
216         JOptionPane.QUESTION_MESSAGE);
217     // sjekker om bruker trykker cancel/kryss og avslutter
218     if (elevKjonn == null || elevKjonn == "")
219     {
220         System.exit(0);
221     }
222     // hvis brukeren lar input stå tom, gi en ny sjans
223     if (elevKjonn.isEmpty())
224     {
225         do
226         {
227             elevKjonn = JOptionPane.showInputDialog(null,
228                 "M\u00E5 ha et kj\u00F8nn, skriv inn p
\u00E5 nytt:");
229             // hvis bruker trykker cancel/kryss avslutt
230             if (elevKjonn == null || elevKjonn == "")
231             {
232                 System.exit(0);
233             }
234             }while (elevKjonn.isEmpty());// fortsett så lenge
input er tom
235     }
236     // sjekker om brukeren har skrevet inn mann
237     if (elevKjonn.toLowerCase().equals("mann"))
238         mann = true;// setter da mann til true
239     // sjekker om brukeren har skrevet inn noe annet enn
mann
240     else
241         mann = false;// setter da mann til false
242     // input fra brukeren
243     String studieStart = JOptionPane.showInputDialog(null,
244         "Hvilket \u00E5r startet studenten?",
245         "Studiestart",
246         JOptionPane.QUESTION_MESSAGE);
247     // hvis brukeren trykker cancel skriv ut beskjed og ny
input
248     if (studieStart == null || studieStart == "")
```

## Main.java

```
249     {
250         System.exit(0);
251     }
252     // hvis brukeren lar input stå tom start på nytt
253     if (studieStart.isEmpty())
254     {
255         do
256         {
257             studieStart = JOptionPane.showInputDialog(null,
258                 "M\u00E5 ha et studie\u00E5r, skriv inn
259                 p\u00E5 nytt:");
260             if (studieStart == null || studieStart == "")
261             {
262                 System.exit(0);
263             }
264             }while (studieStart.isEmpty()); // fortsett så lenge
265             input er tom
266             }
267             int studStart = Integer.parseInt(studieStart); // parse
268             til integer
269             // hvis input er mindre enn 2010 eller større enn 2014
270             if (studStart < 2010 || studStart > 2014)
271             {
272                 do
273                 {
274                     studieStart = JOptionPane.showInputDialog(null,
275                         "Studenter som g\u00E5r p\u00E5 dette "
276                         + "studiet har startet mellom 2010 og
277                         2014, "
278                         + "skriv inn \u00E5rstall p\u00E5
279                         nytt.");
280                     // hvis bruker trykker cancel/kryss avslutt
281                     if (studieStart == null || studieStart == "")
282                     {
283                         System.exit(0);
284                     }
285                     studStart = Integer.parseInt(studieStart);
286                     }while (studStart < 2010 || studStart > 2014); /*
287                     fortsett
288                     så lenge input er mindre enn 2010 eller større enn
289                     2014*/
290                 }
291                 // input fra brukeren
292                 String fagomrade = JOptionPane.showInputDialog(
293                     "Velg nr. for faget til studenten:\n"
```

## Main.java

```
287         + "1. Norsk\n"
288         + "2. Engelsk\n"
289         + "3. Matematikk");
290 // hvis bruker trykker cancel/kryss avslutt
291 if (fagomrade == null || fagomrade == "")
292 {
293     System.exit(0);
294 }
295 // hvis input står tom, gi en ny sjans
296 if (fagomrade.isEmpty())
297 {
298     do
299     {
300         fagomrade = JOptionPane.showInputDialog(
301             "M\u00E5 ha et fag, pr\u00F8v p\u00E5
nytt.\n"
302             + "1. Norsk\n"
303             + "2. Engelsk\n"
304             + "3. Matematikk");
305         if (fagomrade == null || fagomrade == "")// hvis
kryss/cancel avslutt
306         {
307             System.exit(0);
308         }
309     }while (fagomrade.isEmpty());// fortsett så lenge
input er tom
310 }
311 int fag = Integer.parseInt(fagomrade);// parse til
integer
312 // hvis bruker har tatt et tall utenfor intervallet,
start på nytt
313 if (fag < 1 || fag > 3)
314 {
315     do
316     {
317         JOptionPane.showMessageDialog(null,
318             "Feil nr, pr\u00F8v igjen.");
319         fagomrade = JOptionPane.showInputDialog(
320             "Velg nr. for faget til studenten:\n"
321             + "1. Norsk\n"
322             + "2. Engelsk\n"
323             + "3. Matematikk");
324         if (fagomrade == null || fagomrade == "")// hvis
kryss/cancel, avslutt
325         {
```



## Main.java

```
326         System.exit(0);
327     }
328     fag = Integer.parseInt(fagomrade);
329     }while (fag < 1 || fag > 3);/* fortsett så
330     lenge tallet er utenfor intervallet*/
331 }
332 // slutt på input fra brukeren
333 // start å sende input fra brukeren til klassen Student
for senere arbeid
334     try
335     {
336         student.setFornavn(elevFornavn);//
337         student.setEtternavn(elevEtternavn);
338         student.setKjonn(mann);
339         student.setStudiestart(studieStart);
340         if (fagomrade.equals("1"))
341         {
342             // her bestemmes det hvilket fag studenten har
343             student.setFag("Norsk");
344         }
345         else if (fagomrade.equals("2"))
346         {
347             // her bestemmes det hvilket fag studenten har
348             student.setFag("Engelsk");
349         }
350         else if (fagomrade.equals("3"))
351         {
352             // her bestemmes det hvilket fag studenten har
353             student.setFag("Matematikk");
354         }
355         // sender studentene til klassen Gruppe for å gjøre
til objekter
356         gruppe.leggTilStudent(student);
357         /* etter hver gang leggTilStudent har kjørt så
358         sendes de inn i filbehandlings-klassen og lagres i
objekter der*/
359         Filbehandling.lagreGruppe(gruppe);
360     }
361     catch (FormatterClosedException fce)// hvis try ikke
slår til - feilmelding
362     {
363         JOptionPane.showMessageDialog(
364             null,
365             "",
366             "Feil ved skriving til fil",
```

# Main.java

```

367             JOptionPane.PLAIN_MESSAGE);
368         return;
369     }
370     catch (NoSuchElementException ee)// hvis try ikke slår
    til - feilmelding
371     {
372         JOptionPane.showMessageDialog(
373             null,
374             "",
375             "Feil input, pr\u00F8v igjen",
376             JOptionPane.PLAIN_MESSAGE);// feilmelding
377     }
378     // brukeren får valg om å starte innlesing av flere
    elever på nytt
379     svar = JOptionPane.showOptionDialog(null,
380         "Vil du lese inn flere elever?",
381         "Klasseliste",
382         JOptionPane.YES_NO_OPTION,
383         JOptionPane.QUESTION_MESSAGE,
384         null,
385         null,
386         null);
387     if (svar == JOptionPane.NO_OPTION)
388     {
389         visMeny();
390     }
391     while (svar == JOptionPane.YES_OPTION);
392     System.exit(0);
393 }// slutt på metoden leggTilStudent
394
395 public void lagDiverseGrupper()
396 {
397     //lokal variabel
398     int igjen;
399     // start en løkke for å finne ut hvilke grupper brukeren
    vil generere
400     do
401     {
402         // hvis et valg-tre som sender brukeren videre på
    riktig gren
403         String valgtre = JOptionPane.showInputDialog(null,
404             "Skriv inn tallet p\u00E5 hva du vil gj
    \u00F8re:\n"
405             + "1: Sorter etter kj\u00F8nn.\n"
406             + "2: Sorter etter fag.\n"

```

## Main.java

```
407         + "3: Sorter etter studiestart.\n"
408         + "4: Sorter alfabetisk.\n"
409         + "5: Lag tilfeldig sammensetting.\n"
410         + "6: Lag liste over de som er godkjent
    til eksamen.\n"
411         + "0: Avslutt.");
412     // kaller opp og konstruerer en konstruktør for klassen
    GenererGruppe
413     GenererGruppe kjor = new GenererGruppe(gruppe);
414     // sortere etter kjønn
415     if (valgtre == null){ //Brukeren lukker dialogen.
416         visMeny(); //GÅ tilbake til hovedmeny
417     }
418
419     if (valgtre.equals("1"))
420     {
421         // start metoden sortereKjonn i klassen
    GenererGruppe (Ida)
422         kjor.sortereKjonn();
423     }
424     // sortere etter fag
425     else if (valgtre.equals("2"))
426     {
427         // start metoden sortereFag i klassen GenererGruppe
    (Ida)
428         kjor.sortereFag();
429     }
430     // sortere etter studiestartår
431     else if (valgtre.equals("3"))
432     {
433         // start metoden sortereStudiestart i klassen
    GenererGruppe (Ida)
434         kjor.sortereStudiestart();
435     }
436     // sorterer alfabetisk
437     else if (valgtre.equals("4"))
438     {
439         /* start metoden sortereAlfabetisk i klassen
    GenererGruppe (Ida),
440         sender inn studenter fra klassen Gruppe*/
441     kjor.sortereAlfabetisk(gruppe.hentStudenterAsArray());
442     }
443     // sett sammen tilfeldig
444     else if (valgtre.equals("5"))
```

## Main.java

```
445         {
446             /* start metoden sortereTilfeldig i klassen
GenererGruppe (Ida),
447             sender inn studenter fra klassen Gruppe*/
448             kjor.sortereTilfeldig(gruppe.hentStudenterAsArray());
449         }
450         // sorterer etter de som kan gå til eksamen
451         else if (valgtre.equals("6"))
452         {
453             // start metoden sortereGodkjent i klassen
GenererGruppe (Ida)
454             kjor.sortereGodkjent();
455         }
456         // avslutter
457         else if (valgtre.equals("0"))
458         {
459             System.exit(0);
460         }
461         // hvis det er skrevet feil tall eller en bokstav så
får bruker ny sjans
462         else
463         {
464             igjen = JOptionPane.showOptionDialog(null,
465             "Du skrev inn et valg som ikke finnes, pr
\u00F8ve p\u00E5 nytt?",
466             "Lister",
467             JOptionPane.YES_NO_OPTION,
468             JOptionPane.QUESTION_MESSAGE,
469             null,
470             null,
471             null);
472             // hvis brukeren velger ja, start gruppevalg igjen
473             if (igjen == JOptionPane.YES_OPTION)
474                 lagDiverseGrupper();
475             // hvis brukeren gjør noe annet start hele
programmet på nytt
476             else
477             {
478                 visMeny();
479             }
480         }
481         // brukeren får valg om å lage nye lister
482         igjen = JOptionPane.showOptionDialog(null,
483         "Vil du generere nye lister?",
```

## Main.java

```
484         "Lister",
485         JOptionPane.YES_NO_OPTION,
486         JOptionPane.QUESTION_MESSAGE,
487         null,
488         null,
489         null);
490         // hvis brukeren trykker nei start hele programmet på
    nytt
491         if (igjen == JOptionPane.NO_OPTION)
492         {
493             visMeny();
494         }
495     }while(igjen == JOptionPane.YES_OPTION);/* fortsett løkke
    med å
496     generere grupper så lenge igjen er ja*/
497     System.exit(0);
498 }// slutt på metoden lagDiverseGrupper
499 }// slutt på klassen Main
500
```