

```
1 import java.lang.Math;
2 import javax.swing.JOptionPane;
3
4 public class UnikeTall{
5     // Deklarasjon av array for lagring av
    tilfeldige heltall
6     private int[] array;
7
8
9     /* Konstruktør som mottar arrayets lengde som
    parameter og
10     oppretter arrayet. */
11     public UnikeTall(int storrelse){
12         array = new int[storrelse];
13         fyllArray();
14         GUI();
15     }
16
17
18     /* Metode som skal undersøke om et gitt tall
    finnes i arrayet fra før.
19     Tallet det letes etter skal mottas som
    parameter.
20     Metoden skal returnere true hvis tallet finnes
    i arrayet.
21     Hvis ikke skal metoden returnere false. */
22     private boolean tallSjekk(int tall){
23         for (int elem: array){
24             if (tall == elem) return true;
25         }
26         return false;
27     }
28
29
30     /* Metode som skal fylle arrayet med tilfeldige
    tall mellom 100 og 999,
31     begge grenser inkludert. Metoden skal gjøre
    dette på en måte
32     som sikrer at alle tallene er forskjellige.
    Metoden skal ikke ha
33     parametre og heller ikke returnere noen verdi.
```

```

34     Tips bruk (int)(Math.random()*(900)+100); for
    å generere tallene mellom 100 og 999
35     */
36     private int tilfeldigTall(int min, int max){
37         return (int)(Math.random()*((max - min) + 1
    )) + min;
38     }
39
40     private void fyllArray(){
41         for (int i = 0; i < array.length; i++){
42
43             while (true){
44                 int tall = tilfeldigTall(100, 999);
45                 if (!tallSjekk(tall)){
46                     array[i] = tall;
47                     break;
48                 }
49             }
50         }
51     }
52
53     // Metode som finner og returnerer det minste
    tallet i arrayet.
54     public int minsteTall(){
55         int minste = array[0];
56         for (int elem: array){
57             if (elem < minste){
58                 minste = elem;
59             }
60         }
61         return minste;
62     }
63
64     // Metode som finner og returnerer det største
    tallet i arrayet.
65     public int storsteTall(){
66         int storste = array[0];
67         for (int elem: array){
68             if (elem > storste){
69                 storste = elem;
70             }

```

```

71         }
72         return storste;
73     }
74
75     /* Metode som beregner og returnerer den
gjennomsnittlige verdien
76     (double-verdi) av tallene i arrayet. */
77     public double gjennomsnitt(){
78         double sum = 0;
79         for (double elem: array){
80             sum += elem;
81         }
82
83         return sum/array.length;
84     }
85
86     /* Metode som viser tallene i arrayet i en
meldngsboks.
87     I tillegg skal det, i meldingsboks, skrives
ut
88     opplysninger om hvilket tall som er minst,
hvilket som er størst, og
89     hva som er gjennomsnittsverdien, jfr bildet
over. Gjennomsnittsverdien
90     skal skrives ut med en desimal. Bruk gjerne
String.format("%.2f,tall) til dette.
91     Legg inn mellomrom mellom tallene og skriv ut
et passende antall tall
92     per linje. */
93     private void GUI(){
94         JOptionPane.showMessageDialog(null,
toString() +
95             "\n\n Minste tall er " +
minsteTall() +
96             "\n Storste tall er " +
storsteTall() +
97             "\n Gjennomsnittsverdien er " +
String.format("%.2f",gjennomsnitt()), null, 0);
98     }
99
100    public String toString(){

```

```
101         String a = "";
102         int teller = 0;
103         for (int elem: array){
104             if (teller%8 == 0) a += "\n";
105             a += elem + " ";
106             teller++;
107         }
108         return a;
109     }
110 }
```