

```
1 class Bongo {
2     public static void main(String[] args){
3         int [] [] [] bongo = new int [2] [3] [4];
4
5         // SVAR: 24
6     }
7 }
8
```

```
1  class Lovlige {
2      public static void main(String[] args) {
3          {
4              // 1) SVAR: NEI
5              int x = 4.0 / (int) 2.0;
6          }
7          {
8              // 2) SVAR: JA
9              boolean t = (1 > 2);
10         }
11         {
12             // 3) SVAR: NEI
13             double[] f = new double[2.0];
14         }
15         {
16             // 4) SVAR: JA
17             int i=0, j=i, k, l, m=j+2*i;
18         }
19         {
20             // 5) SVAR: NEI
21             int [] heltall = 3;
22         }
23         {
24             // 6) SVAR: NEI
25             int[3] tre = new int {1, 2, 3};
26         }
27         {
28             // 7) SVAR: JA
29             String int2 = "" + true;
30         }
31         {
32             // 8) SVAR: NEI
33             while (i++ < 1000);
34         }
35         {
36             // 9) SVAR: NEI
37             boolean [] a, b == true;
38         }
39         {
40             // 10) SVAR: JA
41             int[][] tusen = {{(int) 1.25},{7}, {(int)3.75}};
42         }
43     }
44 }
45
46
47
48
49
50
51
52
53
54
```

```
1  class AntallGanger {
2      public static void main(String[] args) {
3          {
4              // 1) SVAR: 3 ganger
5              for (int i=2; i<8; i+=2) {
6                  System.out.println("INF1000");
7              }
8          }
9          System.out.println("****");
10         {
11             // 2) SVAR: 0 ganger
12             for (int j=1; j < 10 ; j = j+1 ) {
13                 for (int i = j -1; ++i < j++; j = i+1)
14                     System.out.println("INF1000");
15             }
16         }
17         System.out.println("****");
18         {
19             // 3) SVAR: 8 ganger
20             for (int i=0; i < 2; i++){
21                 for (int j=1; j < 3; j++){
22                     for(int k=2; k < 4; k++){
23                         System.out.println("INF1000");
24                     }
25                 }
26             }
27         System.out.println("****");
28         {
29             // 4) SVAR: 3 ganger
30             int k = 5;
31             while (k > 5 - k--) {
32                 System.out.println("INF1000 ");
33             }
34         }
35         System.out.println("****");
36     }
37 }
38 }
39 }
40 }
```

```
1  class VerdienTilt {
2      public static void main(String[] args){
3          int i=1,j=2, f, t=0;
4          f = (int) ((i*1.0/j)*2.0);
5          while (f < 5) {
6              t = f;
7              f = j;
8              j = t+1;
9          }
10         System.out.println("t=" + t);
11
12         // SVAR: t = 4
13     }
14 }
15
```

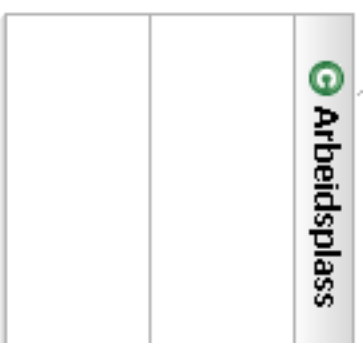
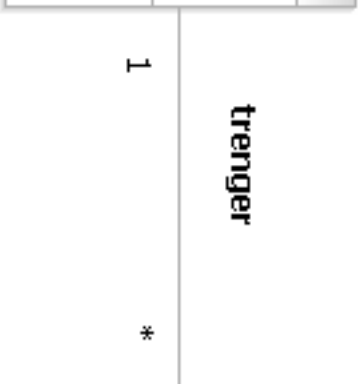
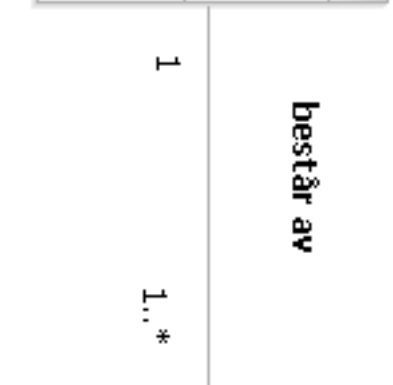
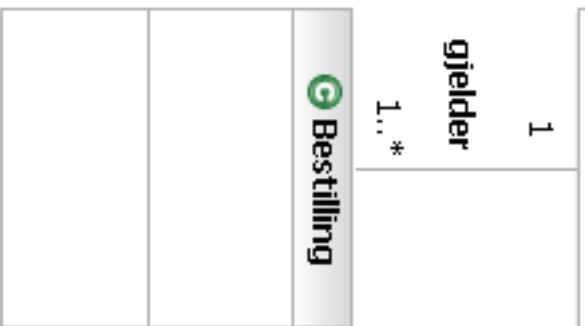
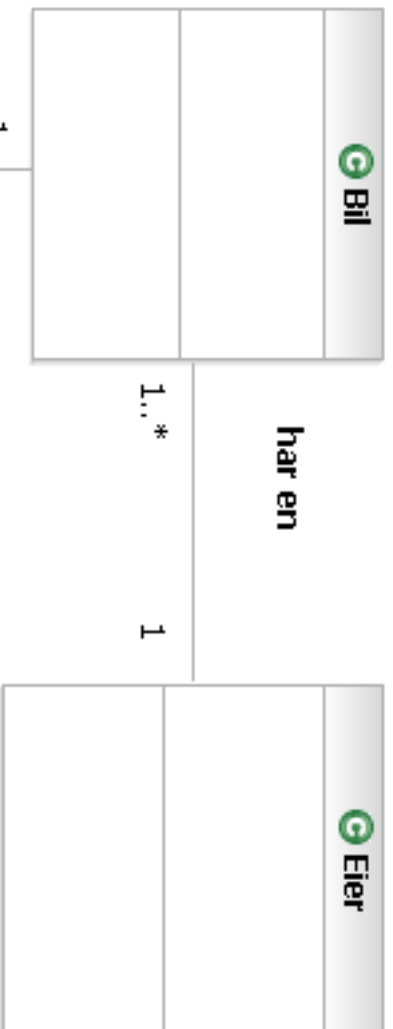
```
1  class ProsentJenter {
2      double prosentJenter(int antallJenter, int antallGutter){
3          return ((double)antallJenter)/(antallJenter + antallGutter);
4      }
5      public static void main(String[] args){
6          ProsentJenter pj = new ProsentJenter();
7          int jenter1 = 50;
8          int gutter1 = 50;
9          System.out.println("" + jenter1 + "j, " + gutter1 + "g => " + pj.
          prosentJenter(jenter1, gutter1) + "%");
10
11         int jenter2 = 3;
12         int gutter2 = 1;
13         System.out.println("" + jenter2 + "j, " + gutter2 + "g => " + pj.
          prosentJenter(jenter2, gutter2) + "%");
14     }
15 }
16
```

```
1  class ArealRektangel {
2      int[] arealRektangel(int[] lengder, int[] bredder){
3          if(lengder.length != bredder.length){
4              return new int[0];
5          }
6          int[] res = new int[lengder.length];
7          for(int i=0; i<lengder.length; i++){
8              if(lengder[i] >=0 && bredder[i] >=0){
9                  res[i] = lengder[i] * bredder[i];
10             } else {
11                 res[i] = 0;
12             }
13         }
14         return res;
15     }
16     public static void main(String[] args){
17         ArealRektangel ar = new ArealRektangel();
18         int res[] = null;
19         res = ar.arealRektangel(new int[]{1, 2, 3, -1}, new int[]{-2, 4,
20             6, 8});
21         //res = ar.arealRektangel({1, 2, 3, -1}, {-2, 4, 6, 8});
22         System.out.print("[");
23         for(int i=0; i<res.length; i++){
24             System.out.print("" + res[i] + ", ");
25         }
26         System.out.print("]");
27         System.out.println();
28     }
29 }
30
```

```
1  class Flyttall{
2      public static String rest(double teller, double nevner){
3          double dele = teller/nevner;
4          String tekst = "" + dele;
5          int desimalpunkt = tekst.indexOf(".");
6          return tekst.substring(desimalpunkt, tekst.length());
7      }
8      public static void main(String[] args){
9          // "1/2"
10         System.out.println(rest(1,2));
11         // "3/4"
12         System.out.println(rest(3,4));
13
14         // SVAR:   .5
15         //         .75
16     }
17 }
18
```

```
1  class Faktorer {
2      String faktorer (int tall){
3          String res = "";
4          for (int i = 2; i <= tall / i; i++) {
5              while (tall % i == 0) {
6                  res += i + " ";
7                  tall = tall / i;
8              }
9          }
10         if (tall > 1)
11             res+= tall;
12         return res;
13     }
14     public static void main(String[] args) {
15         Faktorer fak = new Faktorer();
16         int test1 = 12;
17         int test2 = 111;
18         int svar = 20;
19         System.out.println(fak.faktorer(test1));
20         System.out.println(fak.faktorer(test2));
21         System.out.println("****");
22         System.out.println(fak.faktorer(svar));
23
24         // SVAR: 2 2 5
25     }
26 }
27
```





```
1  import easyIO.*;
2  import java.util.*;
3
4  // Klassen Reisesystem
5  class Reisesystem {
6      public static void main(String [] args) {
7          Brukermodul b = new Brukermodul("Ruter.txt", "Kunder.txt",
8              "Bestilling.txt");
9          b.bestillReise();
10     }
11 }
12
13 // Klassen Brukermodul
14 class Brukermodul {
15     HashMap<String, Rute> ruter = new HashMap<String, Rute>();
16     HashMap<String, Kunde> kunder = new HashMap<String, Kunde>();
17     String kunderFil = null;
18     String bestillingFil = null;
19     String[] DAGER = {"mandag", "tirsdag", "onsdag", "torsdag", "fredag",
20         "lørdag", "søndag"};
21
22     // Konstruktøren til Brukermodul
23     Brukermodul (String ruter, String kunder, String bestilling) {
24         this.bestillingFil = bestilling;
25         this.kunderFil = kunder;
26         In rFil = new In(ruter);
27         while(rFil.hasNext()){
28             Rute rute = new Rute(
29                 rFil.inWord(),
30                 rFil.inWord(),
31                 rFil.inWord(),
32                 rFil.inWord(),
33                 rFil.inWord(),
34                 rFil.inInt());
35             this.ruter.put(rute.nr, rute);
36         }
37         rFil.close();
38         In kFil = new In(kunder);
39         while(kFil.hasNext()){
40             Kunde kunde = new Kunde(
41                 kFil.inWord(),
42                 kFil.inInt(),
43                 kFil.inLine());
44             this.kunder.put(kunde.epost, kunde);
45         }
46         kFil.close();
47     }
48     void bestillReise(){
49         Out skjerm = new Out();
50         skjerm.out("Kundens epostadresse: ");
51         In tast = new In();
52         String epost = tast.inLine();
53
54         Kunde kunde = this.kunder.get(epost);
55         if(kunde == null){
56             skjerm.outln("Ny kunde!");
57             skjerm.out("Kundens navn: ");
58             String navn = tast.inLine();
59             kunde = new Kunde(epost, 0, navn);
60             this.kunder.put(epost, kunde);
61         }
62         skjerm.out("Reise fra: ");
63         String fra = tast.inLine();
64         skjerm.out("Reise til: ");
65         String til = tast.inLine();
66         skjerm.out("Utreisedag: ");
67         String utreisedag = tast.inLine();
```

```

67     skjerm.out("Dager: ");
68     int dager = tast.inInt();
69
70     Rute utreise = null;
71     Iterator<Rute> it = this.ruter.values().iterator();
72     while(it.hasNext()){
73         Rute reise = it.next();
74         if(reise.fraBy.equals(fra) && reise.tilBy.equals(til) &&
75            reise.dag.equals(utreisedag)){
76             utreise = reise;
77             break;
78         }
79     }
80     String hjemreisedag = finndag(utreisedag, dager);
81     Rute hjemreise = null;
82     it = this.ruter.values().iterator();
83     while(it.hasNext()){
84         Rute reise = it.next();
85         if(reise.fraBy.equals(til) && reise.tilBy.equals(fra) &&
86            reise.dag.equals(hjemreisedag)){
87             hjemreise = reise;
88             break;
89         }
90     }
91     if(utreise == null || hjemreise==null){
92         skjerm.outln("Ingen reise funnet!");
93         return;
94     }
95     int pris = utreise.pris + hjemreise.pris;
96
97     Out best = new Out(this.bestillingFil, true);
98     best.outln(kunde.navn + ";" +
99             kunde.epost + ";" +
100            utreise.nr + ";" +
101            utreise.fraBy + ";" +
102            utreise.tilBy + ";" +
103            hjemreise.nr + ";" +
104            utreisedag + ";" +
105            dager + ";" +
106            pris);
107     best.close();
108
109     skjerm.outln("Reise fra " + utreise.fraBy + " til " + utreise.
110            tilBy + " bestilt.");
111
112     int poeng = pris/3;
113     kunde.poeng += poeng;
114
115     Out nyKFil = new Out(this.kunderFil);
116     Iterator<Kunde> itKunder = this.kunder.values().iterator();
117     while(itKunder.hasNext()){
118         Kunde k = itKunder.next();
119         nyKFil.outln(k.epost + " " + k.poeng + " " +
120            k.navn);
121     }
122     nyKFil.close();
123
124     skjerm.outln("Poengsum for " + kunde.navn + " er " + kunde.poeng
125            + ".");
126 }
127 String finndag(String utreisedag, int dager){
128     int utD = 0;
129     for(int i=0; i<DAGER.length; i++){
130         if(DAGER[i].equals(utreisedag)){
131             utD = i;
132             break;

```

```
129         }
130     }
131     int hjD = (utD + dager) % 7;
132     return DAGER[hjD];
133 }
134 }
135
136 class Rute {
137     String nr;
138     String fraBy;
139     String tilBy;
140     String dag;
141     String kl;
142     int pris;
143     Rute(String nr, String fraBy, String tilBy, String dag, String kl,
144         int pris) {
145         this.nr = nr;
146         this.fraBy = fraBy;
147         this.tilBy = tilBy;
148         this.dag = dag;
149         this.kl = kl;
150         this.pris = pris;
151     }
152 }
153 class Kunde {
154     String epost;
155     int poeng;
156     String navn;
157     Kunde(String epost, int poeng, String navn) {
158         this.epost = epost;
159         this.poeng = poeng;
160         this.navn = navn;
161     }
162 }
```

```
1  class Stabel {
2      Element toppen;
3      Element toppen(){return this.toppen;}
4      Element løftAv(){
5          Element tmp = this.toppen;
6          this.toppen = this.toppen.neste;
7          return tmp;
8      }
9      void leggPå(Element ny){
10         ny.neste = this.toppen;
11         this.toppen = ny;
12     }
13     public static void main(String[] args) {
14         Stabel s = new Stabel();
15         Element x = new Element("A");
16         s.leggPå(x);
17         x = new Element("B");
18         s.leggPå(x);
19         System.out.print(s.løftAv().verdi);
20         x = new Element("C");
21         s.leggPå(x);
22         x = new Element("D");
23         s.leggPå(x);
24         System.out.print(s.toppen().verdi);
25         System.out.print(s.løftAv().verdi);
26         System.out.print(s.løftAv().verdi);
27         System.out.print(s.løftAv().verdi);
28
29         // SVAR: BDDCA
30
31         System.out.println();
32     }
33 }
34 class Element {
35     String verdi;
36     Element neste;
37     Element(String verdi){
38         this.verdi = verdi;
39     }
40 }
41
```

```
1  class Sortering {
2      int finnMinste (int[] a, boolean[] brukt) {
3          // Skrive dette:
4          int min = Integer.MAX_VALUE;
5          int index = 0;
6          for(int i=0; i<a.length; i++){
7              if(a[i] < min && ! brukt[i]){
8                  min = a[i];
9                  index = i;
10             }
11         }
12         return index;
13     }
14     int[] sorter (int[] a){
15         int[] b = new int[a.length];
16         boolean[] brukt = new boolean[a.length];
17
18         // Skrive dette:
19         for(int i=0; i<a.length; i++){
20             int ind = finnMinste(a, brukt);
21             b[i] = a[ind];
22             brukt[ind] = true;
23         }
24         return b;
25     }
26     public static void main(String[] args) {
27         int[] test = new int [] {3,5,7,12,54,12,1,1,9,34,54,23,1,68,92,3,5,
28             78,0,1};
29         for(int i=0;i<test.length;i++){
30             System.out.print(test[i] + " ");
31         }
32         System.out.println();
33         Sortering s = new Sortering();
34         int[] res = s.sorter(test);
35         for(int i=0;i<res.length;i++){
36             System.out.print(res[i] + " ");
37         }
38         System.out.println();
39     }
40 }
```