

# Komparatory.java

```
01: import java.util.*;
02:
03: public class Komparatory {
04:     public static void main(String[] args) {
05:         new Komparatory();
06:     }
07:
08:     Komparatory() {
09:         List<Osoba> lista = new ArrayList<Osoba>();
10:         lista.add(new Osoba("Kasia",1980,165));
11:         lista.add(new Osoba("Basia",1986,171));
12:         lista.add(new Osoba("Kasia",1980,168));
13:         lista.add(new Osoba("Hania",1980,171));
14:         lista.add(new Osoba("Magda",1980,171));
15:         lista.add(new Osoba("Kasia",1980,169));
16:         lista.add(new Osoba("Basia",1979,171));
17:         lista.add(new Osoba("Gosia",1975,171));
18:
19:         Collections.sort(lista);
20:         piszListe(lista, "Naturalny: imie, wiek, wzrost");
21:         Collections.sort(lista, new Comp1());
22:         piszListe(lista, "Comp1:      wzrost, wiek, imie");
23:         Collections.sort(lista, new Comp2());
24:         piszListe(lista, "Comp2:      wiek, imie, wzrost");
25:     }
26:
27:     static void piszListe(List lista, String naglowek) {
28:         System.out.println(naglowek);
29:         for (int i = 0; i < lista.size(); i++)
30:             System.out.println(lista.get(i));
31:     }
32: }
33:
34: class Osoba implements Comparable<Osoba> {
35:     String imie;
36:     int rok_ur;
```

```

37:     int    wzrost;
38:
39:     Osoba(String i, int r, int w) {
40:         imie   = i;
41:         rok_ur = r;
42:         wzrost = w;
43:     }
44:
45:     /**
46:      * Porzadek naturalny:
47:      * wedlug imienia, potem wieku, potem wzrostu
48:      */
49:     public int compareTo(Osoba o) {
50:         int k = imie.compareTo(o.imie);
51:         if ( k != 0 ) return k;
52:         k = o.rok_ur - rok_ur;
53:         if ( k != 0 ) return k;
54:         return wzrost - o.wzrost;
55:     }
56:
57:     public String toString() {
58:         return imie + ", " + (2003-rok_ur) +
59:             " lat, " + wzrost + " cm. wzrostu";
60:     }
61: }
62:
63: /**
64:  * Komparator 1:
65:  * wedlug wzrostu, potem wieku, potem imienia
66:  */
67: class Comp1 implements Comparator<Osoba> {
68:     public int compare(Osoba o1, Osoba o2) {
69:         int k = o1.wzrost - o2.wzrost;
70:         if ( k != 0 ) return k;
71:         k = o2.rok_ur - o1.rok_ur;
72:         if ( k != 0 ) return k;
73:         return o1.imie.compareTo(o2.imie);
74:     }
75: }
76:
77: /**
78:  * Komparator 2:
79:  * wedlug wieku, potem imienia, potem wzrostu
80:  */
81: class Comp2 implements Comparator<Osoba> {
82:     public int compare(Osoba o1, Osoba o2) {

```

```
83:     int k = o2.rok_ur - o1.rok_ur;
84:     if ( k != 0 ) return k;
85:     k = o1.imie.compareTo(o2.imie);
86:     if ( k != 0 ) return k;
87:     return o1.wzrost-o2.wzrost;
88: }
89: }
```