

# Agency.java

```

1 //2016.Final
2
3 import java.util.ArrayList;
4
5
6
7
8
9 public class Agency {
10     private List<Player> malePlayers;
11     private List<Player> femalePlayers;
12     private List<Team> teams;
13     public Agency() {
14         malePlayers = new ArrayList<Player>();
15         femalePlayers = new ArrayList<Player>();
16         teams = new ArrayList<Team>();
17     }
18
19     public void addPlayer(Player p, boolean gender) {
20         if (gender) {
21             malePlayers.add(p);
22         } else {
23             femalePlayers.add(p);
24         }
25     }
26
27     public String toString() {
28         String s = "Male Players:\n";
29         for (Player p: malePlayers) {
30             s += p.toString() + "\n";
31         }
32         s += "Female Players:\n";
33         for (Player p: femalePlayers) {
34             s += p.toString() + "\n";
35         }
36         s+="Teams:\n";
37         for(Team t: teams) {
38             s+= t.toString() + "\n";
39         }
40         return s;
41     }
42
43     public void createTeams() {
44         Collections.sort(malePlayers, new Comparator<Player>() {
45             public int compare(Player o1, Player o2) {
46                 return Integer.compare(o2.getProficiency(), o1.getProficiency());
47             }
48         });
49         Collections.sort(femalePlayers, new Comparator<Player>() {
50             public int compare(Player o1, Player o2) {
51                 return Integer.compare(o2.getProficiency(), o1.getProficiency());
52             }
53         });
54         for (int i = 0; i < malePlayers.size(); i++) {
55             teams.add(new Team(malePlayers.get(i), femalePlayers.get(i)));
56         }
57     }
58
59     public Iterator<Player> iterator() {
60         List<Player> elite = new ArrayList<Player>();
61         for(Player p: femalePlayers) {
62             if(p.getProficiency() == 10) {
63                 elite.add(p);
64             }
65         }
66         for(Player p: malePlayers) {

```

```

67     if(p.getProficiency() == 10) {
68         elite.add(p);
69     }
70 }
71 return new Iterator<Player>() {
72     List<Player> e = elite;
73     public boolean hasNext() {
74         return !e.isEmpty();
75     }
76
77     public Player next() {
78         return e.remove(0);
79     }
80 }
81 }
82
83 public double fitness() {
84     double sum = 0;
85     for(Team t:teams) {
86         sum += t.getSatisfaction();
87     }
88     return sum;
89 }
90
91 public void improver() {
92     for(int i = 0; i < teams.size()-1; i++) {
93         double fitness = fitness();
94         for (int j = i + 1; j < teams.size(); j++) {
95             Player temp = teams.get(i).male;
96             teams.get(i).male = teams.get(j).male;
97             teams.get(j).male = temp;
98             double newFit = fitness();
99             if(newFit <= fitness) {
100                 temp = teams.get(i).male;
101                 teams.get(i).male = teams.get(j).male;
102                 teams.get(j).male = temp;
103             } else {
104                 fitness = newFit;
105             }
106         }
107     }
108 }
109 }
110 }
111 }
112

```