

## Cricket.java

```
1
2 public class Cricket extends Vertex {
3
4 // private boolean[] chirp = new boolean[3];
5 private boolean[] chirp = new boolean[2];
6 private boolean willChirp;
7 private int currentTick = 0;
8 private final CricketObserver observer;
9
10 public Cricket(int n, CricketObserver o) {
11     super(n);
12     this.observer = o;
13 }
14
15 public void forceChirp() {
16     willChirp = chirp[0] = true;
17 }
18
19 public void emitChirp() {
20     if(willChirp) {
21         willChirp = false;
22         int n = super.degree();
23         for(int i = 0; i < n; i++) {
24             edges.get(i).other(this).hearChirp();
25         }
26         observer.reportChirp(currentTick, super.n);
27     }
28 }
29
30 private void hearChirp() {
31     chirp[1] = true;
32 }
33
34 public void timeTick(int tick) {
35     currentTick = tick;
36     willChirp = chirp[0];
37     chirp[0] = chirp[1];
38 //     chirp[1] = chirp[2];
39 //     chirp[2] = false;
40     chirp[1] = false;
41 }
```

## Cricket.java

```
42
43     public boolean directFlight(Cricket other) {
44         boolean retval = false;
45         if(equals(other)) return true;
46         int e = super.degree();
47         Cricket o;
48         for(int i = 0; i < e && !retval; i++) {
49             o = super.edges.get(i).other(this);
50             retval = o.equals(other);
51         }
52         return retval;
53     }
54
55     public boolean equals(Object o) {
56         if( !(o instanceof Cricket)) {
57             return false;
58         }
59         if(o == this) {
60             return true;
61         }
62         Cricket casted = (Cricket) o;
63
64         return casted.n == this.n;
65     }
66 }
67
```