

UndirectedEdge.java

```
1 //*****
2 //
3 // File:    UndirectedEdge.java
4 // Package: ---
5 // Unit:    Class UndirectedEdge
6 //
7 //*****
8
9 /**
10  * Class UndirectedEdge represents an edge in a graph that connects two
11  * vertices. It's important to note that the edge does not have a direction nor
12  * weight.
13  *
14  * @author Jimi Ford
15  * @version 2-15-2015
16  */
17 public class UndirectedEdge {
18
19     // private data members
20     private Cricket a, b;
21
22     // future projects may rely on a unique identifier for an edge
23     private final int id;
24
25     /**
26      * Construct an undirected edge
27      * @param id a unique identifier to distinguish between other edges
28      * @param a one vertex in the graph
29      * @param b another vertex in the graph not equal to <I>a</I>
30      */
31     public UndirectedEdge(int id, Cricket a, Cricket b) {
32         this.id = id;
33         // enforce that a.n is always less than b.n
34         if(a.n < b.n) {
35             this.a = a;
36             this.b = b;
37         } else if(b.n < a.n) {
38             this.a = b;
39             this.b = a;
40         } else {
41             throw new IllegalArgumentException("Cannot have self loop");
42         }
43         this.a.addEdge(this);
44         this.b.addEdge(this);
45     }
46
47     /**
48      * Get the <I>other</I> vertex given a certain vertex connected to
49      * this edge
50      *
51      * @param current the current vertex
52      * @return the other vertex connected to this edge
53      */
54     public Cricket other(Cricket current) {
55         if(current == null) return null;
56         return current.n == a.n ? b : a;
57     }
58 }
```