

SpaceStation.java

```
1 //*****
2 //
3 // File:    SpaceStation.java
4 // Package: ---
5 // Unit:    Class SpaceStation
6 //
7 //*****
8
9 /**
10  * Class models a space station floating around in 3D space. This class contains
11  * the math needed to calculate distances to other stations and the power needed
12  * to transmit to them.
13  *
14  * @author Jimi Ford (jhf3617)
15  * @version 4-2-2015
16  *
17  */
18 public class SpaceStation {
19
20     /**
21      * maximum distance a space station can transmit
22      */
23     public static final double MAX_DISTANCE = 40.0E6;
24
25     /**
26      * the station's x-coordinate
27      */
28     public final double x;
29
30     /**
31      * the station's y-coordinate
32      */
33     public final double y;
34
35     /**
36      * the station's z-coordinate
37      */
38     public final double z;
39
40     /**
41      * the station's unique identifier
42      */
43     public final int id;
44
45     /**
46      * Construct a new space station. It is assumed that all the parameters are
47      * less than or equal to MAX_DIM.
48      * @param x x-coordinate in 3D space
49      * @param y y-coordinate in 3D space
50      * @param z z-coordinate in 3D space
51      */
52     public SpaceStation(int id, double x, double y, double z) {
53         this.id = id;
54         this.x = x;
55         this.y = y;
56         this.z = z;
57     }
58 }
```

SpaceStation.java

```
59  /**
60   * compute the straight line distance to another space station
61   * @param other the other space station to compute the distance to
62   * @return the Euclidean distance to this space station
63   */
64  public double distance(SpaceStation other) {
65      return Math.sqrt(powerNeeded(other));
66  }
67
68  /**
69   * compute the power needed to transmit to another space station
70   * @param other the other space station to calculate the power needed
71   * @return the power needed to transmit to the other space station
72   */
73  public double powerNeeded(SpaceStation other) {
74      return ((other.x - x)*(other.x - x)) +
75             ((other.y - y)*(other.y - y)) +
76             ((other.z - z)*(other.z - z));
77  }
78 }
79
```