```java
//*********************************************************************
//
// File:    SpaceStation.java
// Package: ---
// Unit:    Class SpaceStation
//
//*********************************************************************

/**
 * Class models a space station floating around in 3D space. This class contains
 * the math needed to calculate distances to other stations and the power needed
 * to transmit to them.
 *
 * @author Jimi Ford (jhf3617)
 * @version 4-2-2015
 *
 */
public class SpaceStation {

    /**
     * maximum distance a space station can transmit
     */
    public static final double MAX_DISTANCE = 40.0E6;

    /**
     * the station's x-coordinate
     */
    public final double x;

    /**
     * the station's y-coordinate
     */
    public final double y;

    /**
     * the station's z-coordinate
     */
    public final double z;

    /**
     * the station's unique identifier
     */
    public final int id;

    /**
     * Construct a new space station. It is assumed that all the parameters are
     * less than or equal to MAX_DIM.
     * @param x x-coordinate in 3D space
     * @param y y-coordinate in 3D space
     * @param z z-coordinate in 3D space
     */
    public SpaceStation(int id, double x, double y, double z) {
        this.id = id;
        this.x = x;
        this.y = y;
        this.z = z;
    }

```

```java
    /**
     * compute the straight line distance to another space station
     * @param other the other space station to compute the distance to
     * @return the Euclidean distance to this space station
     */
    public double distance(SpaceStation other) {
        return Math.sqrt(powerNeeded(other));
    }

    /**
     * compute the power needed to transmit to another space station
     * @param other the other space station to calculate the power needed
     * @return the power needed to transmit to the other space station
     */
    public double powerNeeded(SpaceStation other) {
        return  ((other.x - x)*(other.x - x)) +
                ((other.y - y)*(other.y - y)) +
                ((other.z - z)*(other.z - z));
    }
}

```