

```

                                GenAirData.java
// GenAirData.java - GENERATE AIR QUALITY DATA
//
// MAINTENANCE HISTORY
// DATE          PROGRAMMER AND DETAILS
// 25-08-07      JS          Original
//
//-----

// IMPORTS

import java.io.*;
import java.util.*;
import java.text.*;

//-----

// MAIN CLASS

public class GenAirData {

    //-----

    // DEFINITIONS

    static private final String FROM_LONG = "10120000";
        // From longitude
    static private final String FROM_LAT = "00250000";
        // From latitude
    static private final String TO_LONG = "10150000";
        // From longitude
    static private final String TO_LAT = "00320000";
        // From latitude
    static private final int SENSOR_CNT = 498;
        // Number of sensors

    //-----

    // CONVERT DEGREES, MINUTES AND SECONDS TO TENTHS OF SECONDS

    static private int
    dmsToTenths (
        String          dms)          // Degrees, minutes and seconds
    {
        int            tenths;        // Tenths of seconds

        tenths = Character.digit (dms.charAt(0), 10);
        tenths = tenths * 10 + Character.digit (dms.charAt(1), 10);
        tenths = tenths * 10 + Character.digit (dms.charAt(2), 10);
        tenths = tenths * 6 + Character.digit (dms.charAt(3), 10);
        tenths = tenths * 10 + Character.digit (dms.charAt(4), 10);
        tenths = tenths * 6 + Character.digit (dms.charAt(5), 10);
        tenths = tenths * 10 + Character.digit (dms.charAt(6), 10);
        tenths = tenths * 10 + Character.digit (dms.charAt(7), 10);
        return tenths;
    }

    //-----

    // CONVERT TENTHS OF SECONDS INTO DEGREES, MINUTES AND SECONDS

    static private String
    tenthsToDms (
        int            tenths)        // Tenths of seconds
    {

```

```

                                GenAirData.java
StringBuffer    sb;                // String buffer for formatting output

sb = new StringBuffer();
sb.append (tenths / (100*60*60*10));
sb.append ((tenths / (10*60*60*10)) % 10);
sb.append ((tenths / (60*60*10)) % 10);
sb.append ((tenths / (10*60*10)) % 6);
sb.append ((tenths / (60*10)) % 10);
sb.append ((tenths / (10*10)) % 6);
sb.append ((tenths / 10) % 10);
sb.append (tenths % 10);
return sb.toString();
}

//-----

// MAIN LINE

static public void
main (
    String[]    argv)            // Argument values
{
    int          fromLongTenths; // From longitude in tenths of degrees
    int          fromLatTenths;  // From latitude in tenths of degrees
    int          toLongTenths;   // To longitude in tenths of degrees
    int          toLatTenths;    // To latitude in tenths of degrees
    Random       rand;          // Random number generator
    int          i;             // General purpose index
    int          thisLongTenths; // This longitude in tenths of degrees
    int          thisLatTenths;  // This latitude in tenths of degrees
    int          thisApi;        // This air pollution index
    DecimalFormat decFmt;       // Decimal formatter

    fromLongTenths = dmsToTenths (FROM_LONG);
    fromLatTenths = dmsToTenths (FROM_LAT);
    toLongTenths = dmsToTenths (TO_LONG);
    toLatTenths = dmsToTenths (TO_LAT);
    rand = new Random (20360);
    decFmt = new DecimalFormat ("000");

    for (i = 0; i < SENSOR_CNT; i++) {
        thisLongTenths = fromLongTenths +
            rand.nextInt (toLongTenths - fromLongTenths + 1);
        thisLatTenths = fromLatTenths +
            rand.nextInt (toLatTenths - fromLatTenths + 1);
        thisApi = (int) ((Math.exp(rand.nextDouble()*4) - 1) * 12.0);
        System.out.println (
            'E' +
            tenthsToDms (thisLongTenths) +
            'N' +
            tenthsToDms (thisLatTenths) +
            decFmt.format (thisApi)
        );
    }
}
}

```