

Source

```
//  
//Official Project in Computer Networks I 2020  
//All credits and copyrights of this project go to :Kessopoulos Ioannis Aem:9271  
//  
import ithakimodem.Modem;  
  
import java.io.*;  
import java.util.ArrayList;  
  
public class Diktya1 {  
  
    public static int k;  
    //k will be the global variable to read modem  
    public static void main(String[] param) throws IOException {  
  
        ArrayList<Long> response_time = new ArrayList<Long>();  
  
        Modem echo_modem;  
        //creates a new variable for the echo modem
```

```
        echo_modem = new Modem(8000);  
//initializing the modem  
  
        echo_modem.setSpeed(8000);  
//setting the speed of the modem  
  
        echo_modem.setTimeout(2000);  
//setting case of timeout  
  
        String dial = "ATD2310ITHAKI\r";  
  
        echo_modem.write(dial.getBytes());  
  
        for (;;) {  
//reading welcoming notes from modem  
  
            try {  
  
                k = echo_modem.read();  
//get results from server  
  
                if (k == -1) break;  
//if connection timed out  
  
                System.out.print((char) k);  
            }  
            catch (Exception x) {  
  
                break;
```

```
    }  
}  
  
//welcome notes stopped  
  
//  
System.out.println("HI");  
long timeStarted = System.currentTimeMillis();  
//from this time the modem starts sending echo_packets  
  
long timePassed;  
//variable usefull to see how much time passed from the time the first packet was  
recieved  
  
String code_echo = "E6062";  
//echo_package code(a random one)  
  
do {  
  
    System.out.println("SENDING PASSOWRD");  
  
    code_echo = "E6062\r";  
    //setting the code  
  
    long time_of_sending_started = System.currentTimeMillis();  
    //get the time that modem initialized  
  
    echo_modem.write(code_echo.getBytes());  
    //sending request in order to start receiving packets  
  
    for (int i = 0; i < 35; i++) {
```

```
try {  
  
    k = echo_modem.read();  
    // reanding package from server  
  
    if (k == -1) break;  
  
    System.out.print((char) k);  
    // print message  
}  
catch (Exception x) {  
  
    break;  
}  
}  
  
long time_of_recieving_ended = System.currentTimeMillis();  
//get time after recieving echo_packet  
  
long temp_response = time_of_recieving_ended - time_of_sending_started;  
//calculating response time according to sending and receiving times  
  
response_time.add(temp_response);  
//add this response time to arraylist  
  
timePassed = System.currentTimeMillis() - timeStarted;  
// update time that has passed from the moment we initialized modem
```

```
}  
    while (timePassed < 250000);  
// do all the above for 250 seconds  
  
    PrintWriter writer = new PrintWriter("echo_session1.txt", "UTF-8");  
//initiate printwriter  
//Here we will print echopackets and their response times in a text file  
    for (long l : response_time) {  
  
        String temp = String.valueOf(l);  
//get item  
  
        writer.println(temp);  
//write it to echo_response.txt  
  
    }  
    writer.close();  
//save the text file  
//We will need it to built the response diagrams  
  
    echo_modem.close();  
//end of echo packets part  
//Sterting the image without error part  
  
    ArrayList<Integer> arrayList_image_noerror = new ArrayList<Integer>();  
//ArrayList for savingthe image we are recieving image to bytes  
  
    Modem modem_image = new Modem();  
//initiliazing modem
```

```
modem_image.setSpeed(80000);  
//setting the speed of the modem  
  
modem_image.open("ithaki");  
//setting the appropriate destination  
  
String image_code = "M0915\r";  
//setting the code (a random one)  
  
char[] arrayChar = image_code.toCharArray();  
//transforms the code it was given an array of chars  
  
int length = arrayChar.length;  
//get the length of that array  
  
//send a request to get the image in bytes  
modem_image.write(image_code.getBytes());  
  
int count=0;  
//initialize counter  
  
for (;;) {  
  
    try {  
  
        k = modem_image.read();  
//start reading from the server  
  
        if(count<208) {
```

```
        ++count;
// skipping welcome notes with this counter

    }
    if(count>205){

        arrayList_image_noerror.add(k);
//after welcoming notes we start saving the image in the arraylist

        System.out.print((char) k);
    }
    if (k == -1) {

        break;
    }

}
catch (Exception x) {
    break;
}

}
//stop reading image

int size = arrayList_image_noerror.size();
//get the size of the arraylist in which we saved the bytes of the iamge
```

```
byte[] array = new byte[size];  
//create a byte array with the same size as the arraylist  
  
for (int a = 0; a < size; a++) {  
  
    array[a] = arrayList_image_noerror.get(a).byteValue();  
    //getting the byte values  
  
}  
//time to save the image we received from the modem  
OutputStream out = null;  
  
try {  
    try {  
        out = new BufferedOutputStream(new  
FileOutputStream("image_noerror1.jpg"));  
        out.write(array);  
    } catch (IOException e) {  
  
    }  
} finally {  
    if (out != null) {  
        try {  
  
            out.close();  
        } catch (IOException ex) {  
  
        }  
    }  
}
```



```
}

//end of saving image_noerror.jpg

modem_image.close();
//Part of image without error: Complete!
//Starting the part of the image with error

ArrayList<Integer> arrayList_image_error = new ArrayList<Integer>();
//ArrayList for saving image into bytes

Modem modem_image_error = new Modem();
//initiliaze modem

modem_image_error.setSpeed(80000);
//setting the speed of the modem

modem_image_error.open("ithaki");
//setting the appropriate destination

String error_code = "G1076\r";
//Setting the code (includes the error, also a random one)

//send the request to start receiving bytes of the image
modem_image_error.write(error_code.getBytes());

int count_error=0;
//skiping welcome notes with this counter
```

```
for (;;) {  
  
    try {  
  
        k = modem_image_error.read();  
        //start reading from server  
  
        if(count_error<208) {  
  
            ++count_error;  
  
        }  
        if(count_error>205){  
  
            arrayList_image_error.add(k);  
            //start saving the image that contains error  
  
            System.out.print((char) k);  
  
        }  
        if (k == -1) {  
  
            break;  
        }  
  
    }  
    catch (Exception x) {  
        break;  
    }  
}
```

```
    }

}

    int size_error = arrayList_image_error.size();
//get the size of the arraylist in which we saved the bytes of the image

    byte[] array_error = new byte[size_error];
//create a byte array in order to save the mage with error

    for (int a = 0; a < size_error; a++) {

        array_error[a] = arrayList_image_error.get(a).byteValue();
//get byte values

    }

//starting to save the image that contains error

    OutputStream out_error = null;

    try {
        try {
            out_error = new BufferedOutputStream(new
FileOutputStream("image_witherror.jpg"));

            out_error.write(array_error);

        } catch (IOException e) {

        }

    } finally {
```

```
        if (out_error != null) {  
            try {  
  
                out_error.close();  
            } catch (IOException ex) {  
  
            }  
        }  
    }  
}  
  
modem_image_error.close();  
// closing modem that we used for the image with error  
//Part of the image with error: Complete!  
//Starting tha part of receiving gps signal  
  
Modem gps_modem = new Modem();  
//initializing modem for the gps part  
  
gps_modem.setSpeed(80000);  
//setting the speed of the modem  
  
gps_modem.open("ithaki");  
//setting the appropriate destination  
  
String code = "P0924";  
// Code to receive gps signal ( a random one)  
  
String gpsInput = code + "R=1000020\r";  
//sending a request
```

```
String gpsData = "";

gps_modem.write(gpsInput.getBytes());

int count_gps=0;

String gps_ = "";
for (;;) {
    try {
        k = gps_modem.read();

        if(count_gps<240) {
            ++count_gps;
        }
        if(count_gps>230){
            char c = (char) k ;
            gps_ = gps_+c;
            System.out.print((char) k);
        }
        if (k == -1) {

            break;
        }

    } catch (Exception x) {
        break;
    }
}
```

```
}
```

```
String[] gps_array = gpsArrayGenerator(gps_);
```

```
String[] parts_offline1 = gps_array[1].split(",");
```

```
String width_offline1 = parts_offline1[2];
```

```
//getting width of the first line
```

```
String[] parts_offline6 = gps_array[6].split(",");
```

```
String width_offline6 = parts_offline6[2];
```

```
//getting width of sixth line
```

```
String[] parts_offline11 = gps_array[11].split(",");
```

```
String width_offline11 = parts_offline11[2];
```

```
//getting width of eleventh line
```

```
String[] parts_offline16 = gps_array[16].split(",");
```

```
String width_offline16 = parts_offline16[2];
```

```
//getting width of sixteenth line
```

```
//string splitting
```

```
String gps_width1 = width_offline1;
```

```
//setting width1
```

```
String gps_width2 = width_offline6;  
//setting width2
```

```
String gps_width3 = width_offline11;  
// setting width3
```

```
String gps_width4 = width_offline16;  
// setting width4
```

```
String length_offline1 = parts_offline1[4];
```

```
String length_offline6 = parts_offline6[4];
```

```
String length_offline11 = parts_offline11[4];
```

```
String length_offline16 = parts_offline16[2];
```

```
String gps_length1 = length_offline1;  
//setting lenth1
```

```
String gps_length2 = length_offline6;  
// setting lenth2
```

```
String gps_length3 = length_offline11;  
// setting lenth3
```

```
String gps_length4 = length_offline16;  
// setting lenth4
```

```
//time to split width
```

```
String[] splitWidth1 = gps_width1.split("\\.");
```

```
String[] splitWidth2 = gps_width2.split("\\.");
```

```
String[] splitWidth3 = gps_width3.split("\\.");
```

```
String[] splitWidth4 = gps_width4.split("\\.");
```

```
String part1_ofwidth1 = splitWidth1[1];
```

```
String part1_ofwidth2 = splitWidth2[1];
```

```
String part1_ofwidth3 = splitWidth3[1];
```

```
String part1_ofwidth4 = splitWidth4[1];
```

```
//time to split length
```

```
String[] splitLength1 = gps_length1.split("\\.");
```

```
String[] splitLength2 = gps_length2.split("\\.");
```

```
String[] splitLength3 = gps_length3.split("\\.");
```

```
String[] splitLength4 = gps_length4.split("\\.");
```

```
String part1_oflength1 = splitLength1[1];
```



```
//set length1
```

```
String part1_oflength2 = splitLength2[1];
```

```
//set length2
```

```
String part1_oflength3 = splitLength3[1];
```

```
//set length3
```

```
String part1_oflength4 = splitLength4[1];
```

```
//set length4
```

```
//tranformation of decWidths
```

```
int transWidth1 = (int) (Integer.parseInt(part1_ofwidth1)*0.006);
```

```
int transWidth2 = (int) (Integer.parseInt(part1_ofwidth2)*0.006+1);
```

```
int transWidth3 = (int) (Integer.parseInt(part1_ofwidth3)*0.006-1);
```

```
int transWidth4 = (int) (Integer.parseInt(part1_ofwidth4)*0.006);
```

```
//tranformation of decWidths
```

```
int transLength1 = (int) (Integer.parseInt(part1_oflength1)*0.006);
```

```
int transLength2 = (int) (Integer.parseInt(part1_oflength2)*0.006);
```

```
int transLength3 = (int) (Integer.parseInt(part1_oflength3)*0.006);
```

```
int transLength4 = (int) (Integer.parseInt(part1_oflength4)*0.006+1);

part1_ofwidth1 = transWidth1 +"";

part1_ofwidth2 = transWidth2 +"";

part1_ofwidth3 = transWidth3 +"";

part1_ofwidth4 = transWidth4 +"";

part1_oflength1 = transLength1 +"";

part1_oflength2 = transLength2 +"";

part1_oflength3 = transLength3 +"";

part1_oflength4 = transLength4 +"";

ArrayList<Integer> myList = new ArrayList<Integer>();

//initialize myList ArrayList

String gpsValue = code +"T=2257"+part1_oflength1+"4037"+part1_ofwidth1
+"T=2257"+part1_oflength2+"4037"+part1_ofwidth2+"T=2257"+part1_oflength3+"4
037"+part1_ofwidth3+"T=2257"+part1_oflength4+"4037"+part1_ofwidth4+"\r";

try{

    for(int i=0; i<gpsValue.length();i++)
```

```
{

    gps_modem.write((int)gpsValue.toCharArray()[i]);
//start sending request for points

}

}
catch (Exception e) {

}

//reading values
for (;;) {

    try {

        k = gps_modem.read();
//starting to read modem again

        if (k==-1) break;

        System.out.print((char)k);

        myList.add(k);
//get the gps image from the server

    }
    catch (Exception x) {
        break;
    }
}
```

```
    }

}

byte [] imageMaker = new byte [myList.size()];
//create a byte array (we are going need it for the image)

for (int i = 0; i < imageMaker.length; i++) {

    imageMaker [i] = myList.get(i).byteValue();
//convert myList into bytes

}

//It's time to save the image
try {

    FileOutputStream fileOutputStream = new FileOutputStream("gps.jpg");

    try {

        fileOutputStream.write(imageMaker);

    }

    finally {

        fileOutputStream.close();

    }

}
```

```
    }  
    catch (Exception e) {  
  
    }  
  
    //Gps signal receiving: Complete!  
    //Initializing ARQ and BER part  
  
    gps_modem.close();  
    //Closing modem we used for the gps signal  
    //Gps signal receiving: Complete!  
    //Initializing ARQ and BER part  
  
    String dial_arq_ez;  
  
    int k, F = 0, C = 0, S = 0, n = 1;  
  
    ArrayList<Long> packagetime = new ArrayList<Long>();  
  
    ArrayList<Integer> packagetime_resent = new ArrayList<Integer>();  
  
    Modem ez_modem;  
  
    ez_modem = new Modem(8000);  
  
    ez_modem.setSpeed(8000);
```

```
ez_modem.setTimeout(2000);

dial_arq_ez = "ATD2310ITHAKI\r";

ez_modem.write(dial_arq_ez.getBytes());

for (;;) {

    try {

        k = ez_modem.read();

        if (k == -1) break;

        System.out.print((char) k);

    }

    catch (Exception x) {

        break;

    }

}

//skipping welcoming notes from server

int xor = 0;

long timePassed_arq_ez = 0;
```

```
String code_arq_ez;

code_arq_ez = "Q7758\r";

ez_modem.write(code_arq_ez.getBytes());

long getCurrentTIME = System.currentTimeMillis();

long getCTime = getCurrentTIME;

do {

    for (int j = 1; j <= 58; j++) {
        try {

            k = ez_modem.read();

            if (k == -1) {

                System.out.print("Bad");

                break;

            }

            System.out.print((char) k); if (j == 32)

            xor = k;
```

```
if (j > 32 && j < 48)
```

```
    xor = k ^ xor;
```

```
if (j == 50)
```

```
    F = k - 48;
```

```
if (j == 51)
```

```
    C = k - 48;
```

```
if (j == 52)
```

```
    S = k - 48;
```

```
}
```

```
catch (Exception x) {
```

```
    System.out.print("Exception");
```

```
    break;
```

```
}
```

```
}
```

```
long t2 = System.currentTimeMillis();
```

```
int FCS = F * 100 + C * 10 + S;
```



```
if (xor == FCS) {  
    packagetime.add(t2 - getCTime);  
    packagetime_resent.add(n);  
    n = 1;  
    getCTime = System.currentTimeMillis();  
    code_arq_ez = "Q7758\r"; ez_modem.write(code_arq_ez.getBytes());  
}  
else {  
    n++;  
    code_arq_ez = "R0987\r";  
    ez_modem.write(code_arq_ez.getBytes());  
}  
  
timePassed_arq_ez = System.currentTimeMillis() - getCTime;  
  
System.out.println(" ");  
}  
while (timePassed_arq_ez < 250000); System.out.println(packagetime);  
  
String listString = "";  
  
PrintWriter writer_arq = new PrintWriter("arq_response.txt", "UTF-8");  
//initialize PrintWriter  
  
for (long l : packagetime) {  
  
    listString = String.valueOf(l);  
  
    writer_arq.println(listString);  
}
```

```
    }

    writer_arq.close();

    PrintWriter writer_arq_rep = new PrintWriter("arq_rep.txt", "UTF-8");
//initialize Printwriter

    String newListString = "";

    for (long w : packagetime_resent) {

        newListString = String.valueOf(w);

        writer_arq_rep.println(newListString

    );
    }

    writer_arq_rep.close();

}
//End of ARQ and BER part

public static String[] gpsArrayGenerator(String enteredString){

    String packets[];
```

```
    packets = enteredString.split("\\$GPGGA");  
    System.out.print(packets.length);  
    for(int i=0; i<packets.length;i++){  
        if(packets[i].contains("0,0000*")){  
            packets[i] = "$GPGGA"+packets[i];  
        }  
    }  
    for(int i=0; i<packets.length;i++){  
        System.out.println(packets[i]);  
    }  
    return packets;  
}  
  
}  
//End of project
```