CLASS "userAplication"

```
import java.io.FileNotFoundException;
import java.util.Scanner;
import java.io.IOException;

public class userApplication {

    public static void main(String[] param) throws IOException , FileNotFoundException {

        System.out.print("OPTIONS\n\n1.Echo Packet\n2.Image\n"
                    + "3.Error Image\n4.GPS\n5.ARQ \n\nEnter Option: ");
        String code;
        Scanner scn = new Scanner(System.in);
        int option = scn.nextInt();
        scn.close();

//--------------------------------------ECHO_PACKET----------------------------------
        if (option == 1) {
            code = "EXXXX\r";
            new echoPacket().run(code);
        }
//-------------------------------------IMAGE-----------------------------------
        if (option == 2) {
            code = "MXXXXFIX\r";
            new image().run(code);
        }
//-------------------------------------ERROR_IMAGE----------------------------------
        if (option == 3) {
            code = "GXXXXFIX\r";
            new errorImage().run(code);
        }
//-----------------------------------------GPS-------------------------------------
        if (option == 4) {
            code = "PXXXXR=1003090\r";
            new gps().run(code);
        }
//-----------------------------------------ARQ-------------------------------------
        if (option == 5) {
            code = "QXXXX\r";
            new arq().run(code);
        }
    }
}
```

Class "echoPacket"

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.util.ArrayList;
import ithakimodem.Modem;

public class echoPacket {

    public void run(String Code)  throws IOException, FileNotFoundException {

        Modem modem = new Modem();
        modem.setSpeed(1000);
        modem.setTimeout(8000);
        modem.open("ithaki");

        int k;
        long strTime = 0;
        String response = "";
        ArrayList<Long> rTime = new ArrayList<Long>();
        long startTime = System.currentTimeMillis();

        while (System.currentTimeMillis() < startTime + 300000) {
            modem.write(Code.getBytes());
            for (;;) {
                try {
                    k = modem.read();
                    response += (char)k;
                    if (k == -1) {
                        System.out.println("Connection Closed");
                        break;
                    }
                    if (response.indexOf("\r\n\n\n") != -1) {           // WELCOME_MSG_RECEIVED /
START_OF_PACKETS
                        strTime = System.currentTimeMillis();
                        response = "";
                    }
                    if (response.indexOf("PSTOP") != -1) {
                        rTime.add(System.currentTimeMillis() - strTime);
                        System.out.print((char)k);
                        System.out.println("|| Packet Arrived");
                        response = "";
                        strTime = System.currentTimeMillis();
                        break;
                    }
                    System.out.print((char)k);
                }catch (Exception x) {break;}
            }
        }
        modem.close();
//-------------------------------------------------------------------------------ADDING_TO_FILE
        FileOutputStream echoPacket = new FileOutputStream("Echo Packets.text");
        PrintStream prt = new PrintStream(echoPacket);
        prt.println("Echo Packets");
        for (int i=0; i<rTime.size(); i++)
            prt.println("Packet " + (i+1) + " Response Time: " + rTime.get(i) + "ms");
        prt.close();
        System.out.println("\n\nFile Created");
    }
}
```

Class "image"

```
import java.awt.image.BufferedImage;
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import javax.imageio.ImageIO;
import ithakimodem.Modem;

public class image {

    public void run(String Code)  throws IOException, FileNotFoundException{

        Modem modem = new Modem();
        modem.setSpeed(8000);
        modem.setTimeout(8000);
        modem.open("ithaki");

        int k;
        String response = "";
        ArrayList<Byte> image = new ArrayList<Byte>();

        modem.write((Code).getBytes());

        for (;;) {
            try {
                k = modem.read();
                response += (char)k;
                if (k == -1) {
                    System.out.println("Connection Closed");
                    break;
                }
                if (response.indexOf("\r\n\n\n") != -1) {               // WELCOME_MSG_RECEIVED /
LOADING_BYTES_OF_IMAGE
                    System.out.println("Loading Bytes of Image...");
                    while (response.indexOf("255217") == -1) {            // EndDelimiterOfImage (255, 217 = ffd8, ffd9)
                        k = modem.read();
                        response += k;
                        image.add((byte)k);
                    }
                    break;
                }
                System.out.print((char)k);
            }catch (Exception x) {break;}
        }
        modem.close();
//-------------------------------------------------------------------------------IMAGE_CONVERSION
        byte[] img = new byte[image.size()];
        for (int i=0; i<image.size(); i++)
            img[i] = image.get(i);

        BufferedImage bufferedImage = ImageIO.read(new ByteArrayInputStream(img));
        ImageIO.write(bufferedImage,"jpg",new File("C:\\Users\\Marios\\Java\\DIKTIA\\Images\\Image_E1.jpg"));
        System.out.println("\nImage Created");
    }
}
```

Class "errorImage"

```
import java.awt.image.BufferedImage;
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import javax.imageio.ImageIO;
import ithakimodem.Modem;

public class errorImage {

    public void run(String Code)  throws IOException, FileNotFoundException{

        Modem modem = new Modem();
        modem.setSpeed(8000);
        modem.setTimeout(8000);
        modem.open("ithaki");

        int k;
        String response = "";
        ArrayList<Byte> eImage = new ArrayList<Byte>();

        modem.write((Code).getBytes());

        for (;;) {
            try {
                k = modem.read();
                response += (char)k;
                if (k == -1) {
                    System.out.println("Connection Closed");
                    break;
                }
                if (response.indexOf("\r\n\n\n") !=- 1) {               // WELCOME_MSG_RECEIVED /
LOADING_BYTES_OF_IMAGE
                    System.out.println("Loading Bytes of Image...");
                    while (response.indexOf("255217") == -1) {            // EndDelimiterOfImage (255, 217 = ffd8, ffd9)
                        k = modem.read();
                        response += k;
                        eImage.add((byte)k);
                    }
                    break;
                }
                System.out.print((char)k);
            }catch (Exception x) {break;}
        }
        modem.close();

//-------------------------------------------------------------------------------------IMAGE_CONVERSION
        byte[] eImg = new byte[eImage.size()];
        for (int i=0; i<eImage.size(); i++)
            eImg[i] = eImage.get(i);

        BufferedImage bufferedImage = ImageIO.read(new ByteArrayInputStream(eImg));
        ImageIO.write(bufferedImage,"jpg",new File("C:\\Users\\Marios\\Java\\DIKTIA\\Images\\Error Image_E2.jpg"));
        System.out.println("\nImage Created");
    }
}
```

Class "gps"

```
import java.awt.image.BufferedImage;
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.util.ArrayList;
import javax.imageio.ImageIO;
import ithakimodem.Modem;

public class gps {
    public void run(String Code)  throws IOException, FileNotFoundException{

        Modem modem = new Modem();
        modem.setSpeed(8000);
        modem.setTimeout(8000);
        modem.open("ithaki");

        int k,   j = 0;
        boolean start = false;
        String response = "";
        String[] gpsPacket = new String[90],    time = new String[5];
        String[] latitude = new String[5], latiT = new String[5];
        String[] longitude = new String[5],   longiT = new String[5];
        String[] info = new String[20],         newCode = new String[5];
        ArrayList<Byte> gpsImage = new ArrayList<Byte>();

        modem.write((Code).getBytes());
        for (;;) {
            try {
                k = modem.read();
                response += (char)k;
                if (k == -1) {
                    System.out.println("Connection Closed");
                    break;
                }
                if (response.indexOf("START ITHAKI GPS TRACKING\r\n") != -1) { // RECEIVING_PACKETS
                    for (int i=0; i<gpsPacket.length; i++) {
                        response = "";
                        while (k != 13) {                                   // EndOfGpsPacket (13 = \r)
                            k = modem.read();
                            System.out.print((char)k);
                            response += (char)k;
                        }
                        gpsPacket[i] = response;
                        k = modem.read();
                    }
                    start = true;
                }
                if (start) {                                                // CREATING_NEWCODE
                    for (int i=0; i<gpsPacket.length; i+=11) {
                        info = gpsPacket[i].split("[.,]");
                        if (j<5) {
                            time[j] = info[1];
                            latitude[j] = info[3];
                            latiT[j] = Integer.toString((int)Math.round((Double.parseDouble(info[4])*0.006)));
                            longitude[j] = Integer.toString(Integer.parseInt(info[6]));
                            longiT[j] = Integer.toString((int)Math.round((Double.parseDouble(info[7])*0.006)));
                            newCode[j] = (longitude[j] + longiT[j] + latitude[j] + latiT[j]);
                            j++;
```

```
                    }
                }
                Code =
("PXXXXT="+newCode[0]+"T="+newCode[1]+"T="+newCode[2]+"T="+newCode[3]+"T="+newCode[4]+"\r");
                modem.write(Code.getBytes());
                start = false;
            }

            if (response.indexOf("STOP ITHAKI GPS TRACKING\r\n") != -1) {   // LOADING_BYTES_OF_IMAGE
                System.out.println("\n\nLoading Bytes of Image...");
                while (response.indexOf("255217") == -1) {                  // EndDelimiterOfImage (255, 217 = ffd8,
ffd9)
                    k = modem.read();
                    response += k;
                    gpsImage.add((byte)k);
                }
                break;
            }
            System.out.print((char)k);
        }catch (Exception x) {break;}
    }
    modem.close();
//-------------------------------------------------------------------------------ADDING_TO_FILE
            FileOutputStream gpsPackets = new FileOutputStream("Gps Packets.text");
            PrintStream prt = new PrintStream(gpsPackets);
            for (int i=0; i<time.length; i++)
                prt.println("Trace " + (i+1) + " time: " + time[i]);
            prt.close();
            System.out.println("\nFile Created");
//-------------------------------------------------------------------------------IMAGE_CONVERSION
    byte[] gpsImg = new byte[gpsImage.size()];
    for (int i=0; i<gpsImage.size(); i++)
        gpsImg[i] = gpsImage.get(i);

    BufferedImage bufferedImage = ImageIO.read(new ByteArrayInputStream(gpsImg));
    ImageIO.write(bufferedImage,"jpg",new File("C:\\Users\\Marios\\Java\\DIKTIA\\Images\\Gps Image_M1.jpg"));
    System.out.println("\nImage Created");
  }
}
```

Class "arq"

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import ithakimodem.Modem;

public class arq {

    public void run(String Code)  throws IOException, FileNotFoundException{

        Modem modem = new Modem();
        modem.setSpeed(1000);
        modem.setTimeout(8000);
        modem.open("ithaki");

        int k,   rNack = 0;
        double ack = 0, nack = 0;
        long strTime = 0,    stpTime = 0;
        String response = "";
        ArrayList<Long> rTime = new ArrayList<Long>();
        ArrayList<Double> pDis = new ArrayList<Double>();
        ArrayList<Byte> digits = new ArrayList<Byte>();
        List<Integer> list = new ArrayList<Integer>();

        long startTime = System.currentTimeMillis();

        while (System.currentTimeMillis() < startTime + 300000) {

            String fcs = "";
            modem.write(Code.getBytes());

            for (;;) {
                try {
                    k = modem.read();
                    response += (char)k;
                    if (k == -1) {
                        System.out.println("Connection Closed");
                        break;
                    }
//-------------------------------------------------------------------------WELCOME_MSG_RECEIVED / START_OF_PACKETS
                    if (response.indexOf("\r\n\n\n") != -1) {
                        strTime = System.currentTimeMillis();
                        response = "";
                    }
//-----------------------------------------------------------------------------------RECEIVING_<XXXX..XX> / FCS
                    if (k == 60) {                          // Decimal of Character "<" (=60)
                        while(k != 62){                         // Decimal of Character ">" (=62)
                            System.out.print((char)k);
                            k = modem.read();
                            digits.add((byte)k);
                        }
                        System.out.print((char)k);
                        k = modem.read();
                        for(int i=0; i<3; i++) {            // RECEIVING_FCS
                            System.out.print((char)k);
                            k = modem.read();
                            fcs += (char)k;
                        }
```

```
                }
                if (response.indexOf("PSTOP") != -1){
                    stpTime = System.currentTimeMillis();
                    System.out.println((char)k);
                    response = "";
                    break;
                }
                System.out.print((char)k);
            }catch (Exception x) {break;}
        }
        int num = 0;                                // CALCULATING XOR
        for (int i=0; i<digits.size()-1; i++)
            num ^= digits.get(i);
        digits.clear();
        if (num == Integer.parseInt(fcs)) {                 // CHECKING_PACKET
            rTime.add(stpTime-strTime);
            System.out.println("Positive Acknowledgement || " + fcs + " = " + num);
            ack++;
            rNack = 0;
            Code = "QXXXX\r";
            strTime = System.currentTimeMillis();
        }
        else {
            rNack++;
            list.add(rNack);
            nack++;
            System.out.println("Negative Acknowledgement || " + fcs + " != " + num);
            Code = "RXXXX\r";
        }
    }
    modem.close();
//-------------------------------------------------------------------------------CALCULATING/ADDING_TO_FILE
    double pA = ack / (ack + nack),      pN = nack / (ack + nack);
    double ber = (float) (1 - Math.pow(pA, 1.0/128.0));              // 128 = 16(bytes) * 8(bits)
    for(int n=1; n<=Collections.max(list); n++)                     // Max_Repeated_Nack_Same_Packet
        pDis.add((1 - pN) * Math.pow(pN, n-1));

    FileOutputStream arqPacket = new FileOutputStream("Arq Packets.text");
    PrintStream prt = new PrintStream(arqPacket);
    prt.println("Packets");
    for (int i=0; i<rTime.size(); i++)
        prt.println("Packet " + (i+1) + " Response Time: " + rTime.get(i) + "ms");
    prt.println("\nACK Probability =  " + pA + "\tAck = " + (int)ack);
    prt.println("NACK Probalitity =  " + pN  + " \tNack = " + (int)nack);
    prt.println("Bit Error Rate =  " + ber);
    prt.println("\nProbability Distribution (Xmax = " + + Collections.max(list) + ")");
    for (int i=0; i<pDis.size(); i++)
        prt.println("P(X= " + (i+1) + ") = " + pDis.get(i));
    prt.close();

    System.out.println("\nFile Created");
  }
}
```