userApplication.java

```java
package source;
//Ergasia Diktya Ypologistwn I , April 2021
//userApplication Java code on PDF version
//9671 Stavros Vasileios Bouliopoulos

import java.util.ArrayList;
import java.io.FileOutputStream;
import ithakimodem.Modem;
import ithakimodem.*;
import java.util.Scanner;
import java.io.File;
import java.io.BufferedWriter;
import java.io.FileWriter;

//erwthmata 1-2 prokatarktika


public class userApplication {

    public Modem giveModem() {
        int n;
        Modem modem;
        modem = new Modem();
        modem.setSpeed(80000); //max speed tou MODEM
        modem.setTimeout(20000);
        //modem.open("ithaki");
        modem.write("ATD2310ITHAKI\r".getBytes());
        for (;;) {
            try {
                n= modem.read();
                if (n== -1) {
                    //System.out.println("Connection lost!");
                    break;
                }
                System.out.print((char)n);   //type cast n according to ASCII
            } catch (Exception x) {
                break;
            }
        }
        //modem.close();
        return modem;
    }

    public void giveEchoPackets() { //erwthma 3 gia echo
        String password;
        Scanner echoRequestCode = new Scanner(System.in);
        System.out.println("Insert the 4 digits echo request code. ");
        password= echoRequestCode.nextLine();
        Modem modem;
        modem= giveModem();
        int oldP= 0,newP= 0, packets= 0;
        double start_t= 0,end_t= 0,total_t= 0,avg_t= 0;
        double start_system= 0,end_system= 0;
        ArrayList<String> general= new ArrayList<String>();
        start_system= System.nanoTime();
        //tests gia 4 lepta se ms
        while(end_system< 1000*4*60) {
            packets++;
            modem.write(("E"+password+"\r").getBytes());
            start_t= System.nanoTime();
            for(;;) {
                try {
                    oldP= newP;
                    newP= modem.read();
                    System.out.print((char)newP);
                    if(((char)newP== 'P') && ((char)oldP== 'O')){ //stOP
                        end_t= (System.nanoTime()-start_t)/1000000;
                        break;
                    }
                } catch (Exception e) {
                    System.out.println("Echo packet missing. ");
                }
            }
            total_t+= end_t;
            general.add(String.valueOf(end_t));
            end_system= (System.nanoTime()-start_system)/1000000;
            System.out.println(" ");
        }
        modem.close();
        avg_t= total_t/packets;
        general.add("Total time of receiving process is "+String.valueOf(total_t)+"milliseconds");
        System.out.println("Total time of receiving process is "+String.valueOf(total_t)+"milliseconds");
        general.add("Total linking time with ithaki server is "+String.valueOf(end_system)+"milliseconds");
        System.out.println("Total linking time with ithaki server is "+String.valueOf(end_system)+"milliseconds");
        general.add("Total packets received are "+String.valueOf(packets));
        System.out.println("Total packets received are "+String.valueOf(packets));
        general.add("Average time for packet to be received is "+String.valueOf(avg_t)+"milliseconds");
        System.out.println("Average time for packet to be received is "+String.valueOf(avg_t)+"milliseconds");
        BufferedWriter texter = null;
        try {
            File echoPackets= new File("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα I\\echop"+password+".txt");
            texter= new BufferedWriter(new FileWriter(("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα I\\echop"+password+".txt"),true));
            if(!echoPackets.exists()) {
                echoPackets.createNewFile();
            }
            int i;
            for(i= 0; i<general.size(); i++) {
                texter.write(String.valueOf(general.get(i)));
                texter.newLine();
            }
            texter.newLine();
        } catch(Exception e){
            System.out.println("File for echo packets not written. ");
        } finally {
            try {
                if(texter != null)
                    texter.close();
            } catch(Exception e){
                System.out.println("BufferedWriter not turned off. "+e);
            }
        }
        modem.close();
        echoRequestCode.close();
        System.out.println("Function giveEchoPackets executed.");
    }

    public void giveImage(int mode) { //erwthma 4 gia image
        String password;
        Scanner imageRequestCode= new Scanner(System.in);
        System.out.println("Insert the 4 digits image request code. ");
        password= imageRequestCode.nextLine();
        Modem modem;
        modem= giveModem();
        String form;
        if(mode== 0) {
            form= "M"; //no errors
        }
        else {
            form= "G"; //with errors
        }
        modem.write((form+password+"\r").getBytes());
        int old_val= 0,new_val= 0;
        ArrayList<Integer> image= new ArrayList<Integer>(); //akeraious ascii
        for(;;) {
            try {
                old_val= new_val;
                new_val= modem.read();
                System.out.println(new_val);
                image.add(new_val);
                if((new_val== 217)&&(old_val== 255)) //thelw delimiter gia 0xFF=255 kai 0xD9=217
                        break;
            }catch(Exception e){
                    System.out.println("Image not retrieved. ");
            }
        }
        String source;
        if(mode== 0) {
            source= ("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα I\\imageNER"+password+".jpeg");
        }
        else {
            source= ("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα I\\imageER"+password+".jpeg");
        }
        try {
            FileOutputStream image_doc= new FileOutputStream(source);
            int i;
            for(i= 0; i< image.size(); i++) {
                image_doc.write(image.get(i));
            }
            image_doc.close();
        }catch(Exception e) {
            System.out.println("File for image not written. ");
        }
        modem.close();
        System.out.println("Function giveImage executed.");
    }

    public void giveGPS() { //erwthma 5 gia gps
        String password;
        Scanner gpsRequestCode= new Scanner(System.in);
        System.out.println("Insert the 4 digits GPS request code. ");
        password= gpsRequestCode.nextLine();
        int old_val= 0,new_val= 0;
        float[] length= new float[6];//mhkos
        float[] lengthD= new float[6];//gwnia moirwn mhkous
        float[] width= new float[6];//platos
        float[] widthD= new float[6];//gwnia moirwn platous
        int c= 0; //counter gia ton katw
        int[] gps_pu= new int[80]; //gps per unit array
        int[][] wid_len= new int[2][10]; //proswrina gia 0platos kai 1mhkos
        //$GPGGA,045208.000,4037.6331,N,02257.5633,E,1,07,1.5,57.8,M,36.1,M,,0000*6D
        //        wra        D platos     D mhkos
        //thelw na ta balw ola se ena T string pinaka AABBCCDDEEZZ ara pinakaki
        ArrayList<String> Tall= new ArrayList<String>(); //kratw oles tis parametrous T
        //String T= new String();
        String[] Tmini= new String[6];
        Modem modem;
        modem= giveModem();
        modem.write(("P"+password+"R=1000199"+"\r").getBytes()); //R=XPPPPLL
        for(;;) {
            try {
                old_val= new_val;
                new_val= modem.read();
                gps_pu[c]= new_val;
                System.out.print((char)gps_pu[c]);
                c++;
            if(((char)new_val== 'G')&&((char)old_val== 'N')){ //meta to trackiNG
                continue;
            }
            if(((char)new_val== 'P')&&((char)old_val== 'O')){ //stamataw sto telos stOP
                break;
            }
            if((new_val== 10)&&(old_val== 13)) { //ksekinaw otan <CR> kai <LF>
                int i;
                for(i= 0; i<10; i++) {
                    wid_len[0][i]= gps_pu[i+18]; //platos erxetai prwta meta thn wra.000,...
                    wid_len[1][i]= gps_pu[i+30]; //mhkos meta to platos kai to N,
                }
                //apo ascii ta kanw akeraious kai meta float . 0=ascii 48
                //antlhsh
                //platos
                widthD[0]= (float) (wid_len[0][0]-48)*10;
                widthD[1]= (float) (wid_len[0][1]-48);
                width[0]= (float) (wid_len[0][2]-48)/10;
                width[1]= (float) (wid_len[0][3]-48)/100;
                width[2]= (float) (wid_len[0][5]-48)/1000; //phdaw .
                width[3]= (float) (wid_len[0][6]-48)/10000;
                width[4]= (float) (wid_len[0][7]-48)/100000;
                width[5]= (float) (wid_len[0][8]-48)/1000000;
                //mhkos
                lengthD[0]= (float) (wid_len[1][1]-48)*10; //
                lengthD[1]= (float) (wid_len[1][2]-48);
                length[0]= (float) (wid_len[1][3]-48)/10;
                length[1]= (float) (wid_len[1][4]-48)/100;
                length[2]= (float) (wid_len[1][6]-48)/1000; //phdaw .
                length[3]= (float) (wid_len[1][7]-48)/10000;
                length[4]= (float) (wid_len[1][8]-48)/100000;
                length[5]= (float) (wid_len[1][9]-48)/1000000;
                //kataxwrhsh kai apokwdikopoihsh tou dekadikou systhmatos apo pshfia se
```

```java
arithmous
                float width_min= 0, width_sec= 0, width_d= 0, len_min= 0, len_sec= 0, len_d= 0;
                int j;
                for(j= 0; j<2; j++) {
                    width_d+= widthD[j];
                    len_d+= lengthD[j];
                }
                for(j= 0; j<6; j++) {
                    width_min+= width[j];
                    len_min+= length[j];
                }
                System.out.print(" Width degrees: "+width_d);
                System.out.print(" Length degrees: "+len_d);
                len_min= len_min*100;
                len_sec= (len_min % 1)*60;
                width_min= width_min*100;
                width_sec= (width_min % 1)*60;
                System.out.print(" Width minutes: "+width_min);
                System.out.print(" Width seconds: "+width_sec);
                System.out.print(" Length minutes: "+len_min);
                System.out.print(" Length seconds: "+len_sec);
                //telos parametros T ,prwta mhkos kai meta platos
                Tmini[0]= String.valueOf(len_d);
                Tmini[1]= String.valueOf(len_min);
                Tmini[2]= String.valueOf(len_sec);
                Tmini[3]= String.valueOf(width_d);
                Tmini[4]= String.valueOf(width_min);
                Tmini[5]= String.valueOf(width_sec);
                // ana 2 pshfia morfh sto teliko T AA ktlp.
                for(j= 0; j<6; j++) {
                    Tmini[j]= Tmini[j].substring(0,2);
                }
                String T= new String();
                T= (Tmini[0]+Tmini[1]+Tmini[2]+Tmini[3]+Tmini[4]+Tmini[5]);
                c= 0;
                Tall.add(T);

            }

            }catch(Exception e) {
                System.out.println("GPS data not retrieved. ");
            }
        }
        modem.close();
        ArrayList<Integer> imageGPS= new ArrayList<Integer>();
        System.out.println("\nInsert again the 4 digits image GPS request code. ");
        password= gpsRequestCode.nextLine();
        modem = giveModem();
        modem.write(("P"+password+"T="+Tall.get(5)+"T="+Tall.get(40)+"T="+Tall.get(80)+"T="+Tall.get(90)+"\r").getBytes()); //3 shmeia apo 0-99
        for(;;) {
            try {
                old_val= new_val;
                new_val= modem.read();
                System.out.println(new_val);
                imageGPS.add(new_val);
                if((new_val== 217)&&(old_val== 255)) { //thelw delimiter gia 0xFF=255 kai 0xD9=217
                    break;
                }
            }catch(Exception e) {
                System.out.println("GPS pin image not retrieved. ");
            }
        }
        String source= ("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα Ι\\imageGPS.jpeg");
        try {
            FileOutputStream image_doc= new FileOutputStream(source);
            int i;
            for(i= 0; i< imageGPS.size(); i++) {
                image_doc.write(imageGPS.get(i));
            }
            image_doc.close();
        }catch(Exception e) {
            System.out.println("File for GPS image not written. ");
        }
        modem.close();
        gpsRequestCode.close();
        System.out.println("Function giveGPS executed.");
    }

    public void giveARQ() { //erwthmata 6,7,8 QXXXX gia ack , RXXXX gia nack
        //PSTART DD-MM-YYYY HH-MM-SS PC <XXXXXXXXXXXXXXXX> FCS PSTOP
        String passwordQ;
        Scanner ackResultCode= new Scanner(System.in);
        System.out.println("Insert the 4 digits ACK result code. ");
        passwordQ= ackResultCode.nextLine();
        String passwordR;
        Scanner nackResultCode= new Scanner(System.in);
        System.out.println("Insert the 4 digits NACK result code. ");
        passwordR= nackResultCode.nextLine();
        int old_val= 0, new_val= 0, packets= 0, error_packets= 0;
        double start_t= 0,end_t= 0,total_t= 0,avg_t= 0;
        double start_system= 0,end_system= 0;
        ArrayList<String> general= new ArrayList<String>();
        Modem modem;
        modem= giveModem();
        int XOR= 0,FCS= 0;
        int m= 0, retrans= 0; //m epanalhpseis gia na stal8ei to paketo, retrans an stal8hke swsta amesws h xreiazetai retrans
        int[] m_times= new int[10]; //m=0 , m=1 ...
        start_system= System.nanoTime();
        //tests gia 4 lepta se ms
        while(end_system< 1000*4*60) {
            modem.write(("Q"+passwordQ+"\r").getBytes());
            packets++;
            start_t= System.nanoTime();
            for(;;) {
                try {
                    old_val= new_val;
                    new_val= modem.read();
                    System.out.print((char)new_val);
                    if(old_val== '<') { //pairnw ta X...X kai metraw XOR
                        while(new_val!= '>') {
                            XOR= XOR^ (char)new_val;
                            new_val= modem.read();
                            System.out.print((char)new_val);
                        }
                        System.out.print("XOR="+XOR);
                    }
                    if ((new_val== ' ')&&(old_val== '>')) { //metraw FCS
                        new_val= modem.read();
                        FCS= Character.getNumericValue((char)new_val)* 100;
                        new_val= modem.read();
                        FCS= FCS+ Character.getNumericValue((char)new_val)* 10;
                        new_val= modem.read();
                        FCS= FCS+ Character.getNumericValue((char)new_val);
                        System.out.print("FCS="+FCS);
                    }
                    if ((new_val== 'P')&&(old_val== 'O')) { //stOP
                        System.out.println(" ");
                        break;
                    }
                }catch (Exception e) {
                    System.out.println("ACK mode packet not retrieved. ");
                }
            }
            m= 0;
            retrans= 0;
            while (XOR!= FCS) {
                if (retrans== 0) {
                    error_packets++;
                }
                retrans= 1; //freno gia error_packets++
                XOR= 0;
                FCS= 0;
                modem.write(("R"+passwordR+"\r").getBytes());
                m++;
                for (;;) {
                    try {
                        old_val= new_val;
                        new_val= modem.read();
                        System.out.print((char)new_val);
                        if (old_val== '<') { //pairnw ta X...X kai metraw XOR
                            while (new_val!= '>') {
                                XOR= XOR^ (char)new_val;
                                new_val= modem.read();
                                System.out.print((char)new_val);
                            }
                            System.out.print("XOR="+XOR);
                        }
                        if ((new_val== ' ')&&(old_val== '>')) { //metraw FCS
                            new_val= modem.read();
                            FCS= Character.getNumericValue((char)new_val)* 100;
                            new_val= modem.read();
                            FCS= FCS+ Character.getNumericValue((char)new_val)* 10;
                            new_val= modem.read();
                            FCS= FCS+ Character.getNumericValue((char)new_val);
                            System.out.print("FCS="+FCS);
                        }
                        if ((new_val== 'P')&&(old_val== 'O')) { //stOP
                            System.out.println(" ");
                            break;
                        }
                    }catch (Exception e) {
                        System.out.println("NACK mode packet not retrieved. ");
                    }
                }
            }
```

```
            m_times[m] += 1;
            end_t= (System.nanoTime()- start_t)/ 1000000;
            total_t+= end_t;
            general.add(String.valueOf(end_t));
            end_system=(System.nanoTime()- start_system)/ 1000000;
        }
        //L=16chars=16x8=128
        //PER = (1-BER)^L
        //PER = ACK / (ACK + NACK)
        double per= 0, ber= 0;
        int nack= 0;
        int i;
        for(i= 1; i<10; i++) {
            nack+=m_times[i]*i;
        }
        per=(double) packets/(packets+nack); //isws thelei error_packets
        ber=1-(Math.pow(per,1.0/128.0));
        //System.out.println("\nPackets received: "+packets);
        //System.out.println("Total linking time with ithaki server: "+total_t);

        avg_t= total_t/packets;
        general.add("Total time of receiving process is "+String.valueOf(total_t)+"milliseconds");
        System.out.println("Total time of receiving process is "+String.valueOf(total_t)+"milliseconds");
        general.add("Total linking time with ithaki server is "+String.valueOf(end_system)+"milliseconds");
        System.out.println("Total linking time with ithaki server is "+String.valueOf(end_system)+"milliseconds");
        general.add("Total packets calls received are "+String.valueOf(packets+nack));
        System.out.println("Total packets calls received are "+String.valueOf(packets+nack));
        general.add("Average time for packet to be received is "+String.valueOf(avg_t)+"milliseconds");
        System.out.println("Average time for packet to be received is "+String.valueOf(avg_t)+"milliseconds");
        general.add("Total NACK calls are "+nack);
        System.out.println("Total NACK calls are "+nack);
        general.add("Bit error rate is "+ber);
        System.out.println("Bit error rate is "+ber);
        for(i= 0; i<10; i++) {
            System.out.println(m_times[i]+" packets needed "+i+" requests.");
            general.add(m_times[i]+" packets needed "+i+" requests.");
        }

        BufferedWriter texter= null;
        try {
            File ARQstats= new File("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα I\\ARQ.txt");
            texter= new BufferedWriter(new FileWriter(("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα I\\ARQ.txt"),true));
            if(!ARQstats.exists()) {
                ARQstats.createNewFile();
            }
            int j;
            for(j= 0; j<general.size(); j++) {
                texter.write(String.valueOf(general.get(j)));
                texter.newLine();
            }
            texter.newLine();
        } catch(Exception e) {
            System.out.println("File for ARQ statistics not written. ");
        } finally {
            try {
                if(texter != null)
                    texter.close();
            } catch(Exception e){
                System.out.println("BufferedWriter not turned off. "+e);
            }
        }
        modem.close();
        ackResultCode.close();
        nackResultCode.close();
        System.out.println("Function giveARQ executed. ");
    }

    public static void main(String[] args) {
        Scanner order= new Scanner(System.in);
        System.out.print("\nWelcome to userApplication for virtual modem ithaki server project. Enter one of the following numbers:\n 1 for echo packets\n 2 for clear image\n 3 for error image\n 4 for GPS pins\n 5 for ARQ\n 6 for exit.");
        for(;;) {

            try {
                int key= order.nextInt();
                switch(key) {
                case 1:
                    new userApplication().giveEchoPackets();
                    continue;
                case 2:
                    new userApplication().giveImage(0);
                    continue;
                case 3:
                    new userApplication().giveImage(1);
                    continue;
                case 4:
                    new userApplication().giveGPS();
                    continue;
                case 5:
                    new userApplication().giveARQ();
                    continue;
                case 6:
                    System.out.println("Bye bye user.Hope you enjoyed the virtual ride :) ");

                    break;
                }
                if(key== 6) {
                    break;
                }
            } catch(Exception e) {
                System.out.println("Invalid key given. ");
                break;
            }
        }
    }
}

```