



```
package diktya1;

import java.io.*;
import ithakimodem.*;

/*η εφαρμογή μας αποτελείται από μια κλάση ,τη VirtualModem , που περιέχει
τη main συνάρτηση και τις συναρτήσεις:
readAndPrintTheMsg,requestEchoPackets,requestImages,requestGPS και ARQ
*/

public class VirtualModem {

    static int k;//εδω store temporarily το byte που διαβαζω απο την Ιθακη
    static Modem modem;
    static long t0,t1,t2; //εδω σώζω τις χρονικές στιγμές που χρειάζονται
    //παρακάτω για τη χρονομέτρηση κάποιων γεγονότων

    public static void main(String[] args) {

        modem=new Modem();//δημιουργώ το αντικείμενο modem και το σεταρω
        modem.setSpeed(80000);
        modem.setTimeout(2000);
        modem.open("ithaki");//καλω το μοντεμ της Ιθακης
        readAndPrintTheMsg();//διαβαζω και τυπωνω το μηνυμα καλωσορισματος
        //της Ιθακης
        requestEchoPackets();//για διαρκεια 4 λεπτων λαμβανω απο την Ιθακη
        //echo packets
        requestImages(); //ζητω 2 εικονες απο το video coder..μια error free
        //και μια με errors
        requestGPS(); //gps part της εργασιας
        ARQ(); //arq part της εργασιας
        modem.close();//τελος εργασιας...κλεινω το μοντεμ και τυπωνω μηνυμα
        //τελους
        System.out.println("end");

    }    //κλεινει η main
```

```

static void readAndPrintTheMsg(){

    InputStream inModem=modem.getInputStream();

    try{
        //διαβάζω ένα χαρακτήρα απο το μήνυμα καλωσορισματος και το
        //τυπωνω στην οθονη
        k=modem.read();
        if (k==-1){
            System.out.println("the communication with the server has
stopped");
            modem.close();
            System.exit(2);
        }
        System.out.print((char)k);
        //περιμένω 100 ms για να προλάβουν και τα αλλα bytes του welcome
        //message να φθάσουν
        t0=System.currentTimeMillis();
        while (System.currentTimeMillis() - t0 < 100) {}
        //όσο έχω bytes να διαβάσω,τα διαβαζω και τα τυπωνω στην οθονη
        while (inModem.available() != 0 ){
            k=modem.read();
            if (k==-1){
                System.out.println("the communication with the server has
stopped");
                modem.close();
                System.exit(2);
            }
            System.out.print((char)k);
        }

    } catch (Exception x){
        System.out.println(x.getMessage());
        modem.close();
        System.exit(2);
    }

}

} //κλεινει η read and print the welcome message

```

```

static void requestEchoPackets(){

    long dt; //εδω σωζω τη χρονικη αποκριση του συστηματος για καθε
    // πακετο
    try{
        File f=new File("C:\\Users\\john\\Desktop\\echoPackets.txt");
        /* καταγράφω στο αρχειο αυτο το dt για καθε πακετο. θα χρησιμοποιησω τα
        αποτελεσματα του file αυτου αργότερα για να φτιαξω το γραφημα G1 */

        PrintWriter out=new PrintWriter(new FileWriter(f));

        t0=System.currentTimeMillis();

        while ( System.currentTimeMillis() - t0 <=  4*60*1000){ //για
            //4 mins συνεχόμενα ζητω απο την Ιθακη echo packets

            char previous='g';

```

```

        t1=System.currentTimeMillis();//κρατω στην t1 τη χρονικη
        //στιγμη που εφυγε το request απο μενα
        modem.write("EXXX\r".getBytes()); //οπου XXXX ο εκαστοτε
        //echo request code που μου δινει η Ιθακη

        for(;;){

            k=modem.read();
            if (k==-1){
                System.out.println("the communication with the server
has stopped");

                out.close();
                modem.close();
                System.exit(2);
            }
            System.out.print((char)k);
            char now=(char)k;
            if (now=='P' && previous=='O'){
                //το πακετο καταληγει στην ακολουθια PSTOP
                //επομενως οταν ο χαρακτηρας που διαβασω ειναι ο P και ο
                //προηγουμενος το O
                //σημαινει οτι εφθασα στο τελος του πακετου
                t2=System.currentTimeMillis();
                dt=t2-t1;
                System.out.println();
                System.out.println(dt);
                out.println(dt);
                break;
            }
            previous=now;

        }

        }//εδω κλεινει το while loop των 4 mins

        out.close();

        }catch(Exception x){
            System.out.println(x.getMessage());
            modem.close();
            System.exit(2);
        }

    }//κλεινει η request Echo Packets

```

```

    static void requestImages(){

        int previous=0;
        try{
            //αρχικα ζητω μια εικονα error free
            File f1=new
File("C:\\Users\\john\\Desktop\\imageErrorFree.jpg");
            FileOutputStream out1=new FileOutputStream(f1);

            modem.write("MXXX\r".getBytes());

```

```

        for(;;){
            k=modem.read();
            if (k==-1){
                System.out.println("the communication with the server
has stopped");
                modem.close();
                System.exit(2);
            }
            if ((k==0xD9)&&(previous==0xFF)){
                //ενα jpeg αρχείο καταλήγει στην ακολουθία των 2
                //bytes 0xD9 και 0xFF
                out1.write(k);
                break;
            }
            out1.write(k);
            previous=k;
        }

        out1.close();

        //ζητώ μια εικόνα with errors
        File f2=new
File("C:\\Users\\john\\Desktop\\imageWithError.jpg");
        FileOutputStream out2=new FileOutputStream(f2);

        modem.write("GXXXX\r".getBytes());

        previous=0;
        for(;;){
            k=modem.read();
            if (k==-1){
                System.out.println("the communication with the server
has stopped");
                modem.close();
                System.exit(2);
            }
            if ((k==0xD9)&&(previous==0xFF)){
                out2.write(k);
                break;
            }

            out2.write(k);
            previous=k;
        }

        out2.close();

        }catch(Exception x){
            System.out.println(x.getMessage());
            System.exit(2);
        }
    }

    }//κλείνει η requestImages

```

```

static void requestGPS(){

    String request="PXXXX".concat("R=1PPPP31\r");

    //στην εικονα απο GoogleMaps θα απεικονισουμε 4 ιχνη
    //τα οποια θα απεχουν το ενα απο το αλλο 10 sec
    //γι αυτο θα παρω 31 ιχνη συνολικα
    //στην πρωτη συνοδο ξεκιναμε απο το ιχνος 100 (PPPP=0100)
    //και στην δευτερη με PPPP=0200

    modem.write(request.getBytes());

    try{
        char ch;
        File f1=new
File("C:\\Users\\john\\Desktop\\gps.jpg");
        FileOutputStream out1=new FileOutputStream(f1);
        String coordinates="";//εδω θα εχω την ακολουθια των
//συντεταγμενων απο τα ιχνη που θελω να απεικονισω σε εικονα απο GoogleMaps
        String line;//εδω σωζω μια γραμμη απο το response
        long previousSeconds=-1;

        while (true){
            //θα εκτελειται μια φορα για καθε γραμμη του GPS response
            //απο αυτο loop θα βγω οταν διαβασω το STOP ITHAKI GPS TRACKING
            //απο εδω θελω να παρω τα coordinates για να τα απεικονισω στο jpg
            line="";
            char previousChar='a';

            while(true){
                //διαβαζω μια γραμμη απο το response και τη σωζω στο line
                k=modem.read();
                if (k==-1){
                    System.out.println("the communication
with the server has stopped");
                    modem.close();
                    System.exit(2);
                }
                ch=(char)k;
                if (ch=='\n' && previousChar=='\r'){
                    line=line.concat("\n");
                    break;
                }
                line=line.concat(new Character(ch).toString());
                previousChar=ch;
            }

            //κανω parsing τη line
            if (line.equals("START ITHAKI GPS TRACKING\r\n")){
                System.out.print(line);
            }
            else if (line.equals("STOP ITHAKI GPS
TRACKING\r\n")){
                System.out.print(line);
                break;
            }
            else{
                System.out.print(line);
            }
        }
        //παιρνω coordinates απο το ιχνος αμα το ιχνος απεχει απο το last trace
    }
}

```

```

//που πηρα 10 seconds

String[] GpsTrace=line.split(",");
//κανω parse την ωρα ληψης
String hours=GpsTrace[1].substring(0,2);
String mins=GpsTrace[1].substring(2,4);
String secs=GpsTrace[1].substring(4,6);
long
seconds=Long.parseLong(hours)*3600+Long.parseLong(mins)*60+Long.parseLong(secs);

if(previousSeconds== -1){
    //ειμαι στο πρωτο ιχνοσ
    //παιρνω τις συντεταγμενες του
    previousSeconds=seconds;
    //παιρνω τις συντεταγμενες ..πρωτα το

//γ.μηκος
String gMikosMoires=GpsTrace[4].substring(1,3);
String gMikosLepta=GpsTrace[4].substring(3,5);
String gMLeptakia=GpsTrace[4].substring(6,10);
double gMsec= Double.parseDouble(gMLeptakia) *
Math.pow(10,-4) * 60;
long x=Math.round(gMsec);
String gMikosSeconds=Long.toString(x);
//τις τοποθετω στο string coordinates
coordinates=coordinates.concat(gMikosMoires).concat(gMikosLepta).concat(gMikosSeconds);

//παιρνω και το γ.πλατος
String gPlatosMoires=GpsTrace[2].substring(0,2);
String gPlatosLepta=GpsTrace[2].substring(2,4);
String gPLeptakia=GpsTrace[2].substring(5,9);
double gPsec= Double.parseDouble(gPLeptakia) *
Math.pow(10,-4) * 60;
x=Math.round(gPsec);
String gPlatosSeconds=Long.toString(x);

coordinates=coordinates.concat(gPlatosMoires).concat(gPlatosLepta).concat(gPlatosSeconds);
}
else{
    if ((seconds-previousSeconds)==10){
        previousSeconds=seconds;
        //παιρνω τις συντεταγμενες
String gMikosMoires=GpsTrace[4].substring(1,3);
String gMikosLepta=GpsTrace[4].substring(3,5);
String gMLeptakia=GpsTrace[4].substring(6,10);
double gMsec= Double.parseDouble(gMLeptakia) *
Math.pow(10,-4) * 60;
long x=Math.round(gMsec);
String gMikosSeconds=Long.toString(x);
//τις τοποθετω στο string coordinates
coordinates=coordinates.concat(gMikosMoires).concat(gMikosLepta).concat(gMikosSeconds);

//παιρνω και το γ.πλατος
String gPlatosMoires=GpsTrace[2].substring(0,2);
String gPlatosLepta=GpsTrace[2].substring(2,4);
String gPLeptakia=GpsTrace[2].substring(5,9);

```

```

        double gPsec= Double.parseDouble(gPLeptakia) *
Math.pow(10,-4) * 60;
        x=Math.round(gPsec);
        String gPlatosSeconds=Long.toString(x);

coordinates=coordinates.concat(gPlatosMoiros).concat(gPlatosLepta).concat(g
PlatosSeconds);

    }
    }
}

} //κλεινει το while loop
//στα coordinates εχω ενα string με τα 4 traces που
//θελω να απεικονιστουν
String s1=coordinates.substring(0,12);
String s2=coordinates.substring(12,24);
String s3=coordinates.substring(24,36);
String s4=coordinates.substring(36,48);

request="PXXXX" + "T=" + s1 + "T=" + s2 + "T=" + s3
+ "T=" + s4 + "\r" ;
modem.write(request.getBytes());

int previous=0;
for(;;){
    k=modem.read();
    if (k==-1){
        System.out.println("the communication with the
server has stopped");
        modem.close();
        System.exit(2);
    }
    if ((k==0xD9)&&(previous==0xFF)){
        out1.write(k);
        break;
    }

    out1.write(k);
    previous=k;
}

out1.close();

}catch(Exception x){
    System.out.println("PROBLEM!");
    System.out.println(x.getMessage());
    System.exit(2);
}

}

} //κλεινει η συναρτηση requestGPS

```

```

static void ARQ(){
    //ο τροπος χρησης των 3 παρακάτω μεταβλητων επεξηγείται
    //στο report.pdf
    int totalEpanekpompes=0;
    int pktTotal=0;
    int epanekpompes=0;
    //εδω σωζω τη χρονικη αποκριση του συστηματος για καθε πακετο
    //που λαμβάνω σωστά μέσω του μηχανισμού ARQ
    long dt;

    int i,x;
    char ch;
    char charX[]=new char[16];
    char fcs[]=new char[3];

    try{
        //στο αρχείο ARQ_i.txt σώζω το dt για κάθε πακέτο
        File f=new File("C:\\Users\\john\\Desktop\\ARQ_i.txt");
        PrintWriter out=new PrintWriter(new FileWriter(f));
        //η χρήση του αρχείου epanekpompes_i.txt εξηγείται στο
        //report.pdf
        File f1=new File("C:\\Users\\john\\Desktop\\epanekpompes_i.txt");
        PrintWriter out1=new PrintWriter(new FileWriter(f1));

        t0=System.currentTimeMillis();

        while ( System.currentTimeMillis() - t0 <= 4*60*1000){

            //κρατω στην t1 τη χρονικη στιγμή που εφυγε
            //το request απο μενα
            t1=System.currentTimeMillis();
            modem.write("QXXX\r".getBytes());
            //οι επανεκπομπές που χρειάστηκαν
            //μέχρις ότου λαβω σωστα το πακετο
            epanekpompes=0;

            while (true){
                //μενω στο loop αυτο
                //μεχρι να λαβω σωστα το πακετο
                //διαβαζω μεχρι το χαρακτηρα '<'
                do{
                    k=modem.read();
                    if (k== -1){
                        System.out.println("the
communication with the server has stopped");
                        modem.close();
                        System.exit(2);
                    }
                    ch=(char)k;
                    System.out.print(ch);
                }while(ch!='<');

                //θα διαβασω τους επομενους 16 χαρακτες XXXX
                //και θα τους αποθηκευσω στον πινακα charX[16]
                for ( i=0;i<16;i++){
                    ch=(char)modem.read();
                    if (k== -1){
                        System.out.println("the communication
with the server has stopped");

```



```

        modem.close();
        System.exit(2);
    }
    System.out.print(ch);
    charX[i]=ch;
}

//στο x αποθηκεύω το αποτέλεσμα της εφαρμογής
//του xor διαδοχικά στους 16 χαρακτήρες
x=charX[0]^charX[1];
for (i=2;i<16;i++){
    x= charX[i] ^ x;
}

//διαβάζω τους επομενους 2 χαρακτήρες
//που είναι το > και το space
for(i=1;i<3;i++){
    k=modem.read();
    if (k==-1){
        System.out.println("the
communication with the server has stopped");
        modem.close();
        System.exit(2);
    }
    ch=(char)k;
    System.out.print(ch);
}
//διαβάζω το FCS
//και το σωζω στο πίνακα fcs[3]
for(i=0;i<3;i++){
    k=modem.read();
    if (k==-1){
        System.out.println("the communication
with the server has stopped");
        modem.close();
        System.exit(2);
    }
    ch=(char)k;
    System.out.print(ch);
    fcs[i]=ch;
}

int fCs=Integer.parseInt(new String(fcs));

//διαβάζω και το υπολοιπο κομματι του πακετου
//δηλαδή τους χαρακτήρες " PSTOP"
for(i=0;i<6;i++){
    k=modem.read();
    if (k==-1){
        System.out.println("the communication
with the server has stopped");
        modem.close();
        System.exit(2);
    }
    ch=(char)k;
    System.out.print(ch);
}

```

```

        //τσεκάρω αν ο αριθμος x ισουται με το FCS
        //στο σημειο αυτο υλοποιω το error detection
        if (x == fCs){
            pktTotal++;
            System.out.println(" sosta to elaba");
            break;
        }
        else{
            pktTotal++;
            System.out.println(" lathos to elaba");
            epanekpompes++;
            totalepanekpompes++;
            //στελνω NACK και ζητω επανεκπομπη του
            //ιδιου πακέτου
            modem.write("R9459\r".getBytes());
        }
    }
    t2=System.currentTimeMillis();
    dt=t2-t1;
    System.out.println(dt);
    out.println( dt );
    out1.println(epanekpompes);
}
out1.println("totalepanekpompes: " + totalepanekpompes);
out1.println("total pkts rx: " + pktTotal);
out.close();
out1.close();

}catch(Exception x1){
    System.out.println(x1.getMessage());
    return;
}

}

} //κλεινει η class

```