

Δίκτυα Υπολογιστών 1
AEM 9124
Γιώργος Εφραίμ Παππάς

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.lang.System;
import java.util.*;
import ithakimodem.Modem;
import java.util.Timer;
import java.lang.Math;

/* * * Δίκτυα Υπολογιστών I * * Experimental Virtual Lab * * Java virtual modem
communications seed code * */

public class virtualModem {

    public static void main(String[] param) throws IOException {    (new virtualModem()).demo();
}

    public void demo() throws IOException {
//ΑΡΧΙΚΗ ΔΙΑΔΙΚΑΣΙΑ ΤΟΥ MODEM
        int k;

        Modem modem;

        modem=new Modem();

        modem.setSpeed(80000);

        modem.setTimeout(1000);

        modem.open("ithaki");
//BYTECODES
        byte[] echob="E8084\r".getBytes();
        byte[] imagebNoE="M0765\r".getBytes();//παιρνει τα bytes για τον κωδικο εικονας χωρις
errors
        byte[] imagebWE="G6585\r".getBytes();//παιρνει τα bytes για τον κωδικο εικονας με
errors
        byte[] gpscode="P6809R=1008099\r".getBytes();
        byte[] ackcode="Q2618\r".getBytes();
        byte[] nackcode="R7561\r".getBytes();
        String gpsIMG="P6809T=";
        int timeRunning=300000; //χρονος που θα καλειται η echopackets και η arg

        PrintWriter echoResponseTimesTXT = new PrintWriter ("responseTimes.txt", "UTF-8"); //αρχιακι
για να κρατησω ολα τα response times της συναρτησης echo_packets

//DIAL ΓΙΑ ΝΑ ΠΑΡΩ ΤΟ ΑΡΧΙΚΟ ΜΗΝΥΜΑ
        first_dial(modem,echob);

//ΚΟΜΜΑΤΙ ΠΟΥ ΑΝΑΦΕΡΕΤΑΙ ΣΤΑ ECHO PACKETS ΚΑΙ ΣΤΑ RESPONSE TIMES
        long timeForEchoPackets=System.currentTimeMillis();
```

Δίκτυα Υπολογιστών 1
AEM 9124
Γιώργος Εφραίμ Παππάς

```
long end=timeForEchoPackets+timeRunning;

while(System.currentTimeMillis() < end) {//εδω τρεχει η echo_packets οσο χρονο της πουμε με
βαση την παραπανω μεταβλητη και συγκεντρωνουμε εκο πακετς και response times
    echo_packets(modem,echob,echoResponseTimesTXT);
}
echoResponseTimesTXT.close();
//EIKONA

FileOutputStream imageOut=new FileOutputStream("E1.jpg");

get_image(modem,imagebNoE,imageOut);

imageOut.close();
//EIKONA ME ΣΦΑΛΜΑ

FileOutputStream imageOutE=new FileOutputStream("E2.jpg");

get_image(modem,imagebWE,imageOutE);

imageOutE.close();
//GPS

for (;;) {
    try {
        k=modem.read();
        if (k==-1) break;
        System.out.print((char)k);
    } catch (Exception x)
    {
        break;}
}
long TT[new long[6];//παιρνω τα T στοιχεια
TT=gpsGA(modem,gpscode);

//Για την εικονα του gps με τα στιγματα
byte[]
gpsIMb=(gpsIMG+TT[5]+"T="+TT[1]+"T="+TT[2]+"T="+TT[3]+"T="+TT[4]+"T="+TT[0]+"\\r").ge
tBytes();//παιρνει τα bytes για τον κωδικο εικονας με errors

FileOutputStream imageOutGPS=new FileOutputStream("M1.jpg");

get_image(modem,gpsIMb,imageOutGPS);

imageOutGPS.close();

//ARQ
int BER[new int[2];
BER[0]=0;
BER[1]=0;
int middleMap[new int [2];
```

Δίκτυα Υπολογιστών 1
AEM 9124
Γιώργος Εφραίμ Παππάς

```
    long timeForARQ=System.currentTimeMillis();
    long endARQ=timeForARQ+timeRunning;
    PrintWriter ResponseTimesTXT = new PrintWriter ("arqRT.txt","UTF-8");
    PrintWriter ResendTimes = new PrintWriter ("nackTimes.txt","UTF-8");
    while(System.currentTimeMillis() < endARQ)
    {
        middleMap=ARQ(modem, ackcode,nackcode,ResponseTimesTXT,ResendTimes);//η
        συνάρτηση επιστρέφει τον πίνακα για κάθε session που περιέχει τα πακέτα που σταλθηκαν σε
        κάθε session καθώς και τα λάθη

        BER[0]+=middleMap[0];//προσθετούμε στον τελικό πίνακα για τα BER τις τιμές που
        πήραμε από την κάθε κλήση της ARQ
        BER[1]+=middleMap[1];

    }
    ResponseTimesTXT.close();
    ResendTimes.close();
    System.out.println("\n");
    System.out.println("\n");
    System.out.println("\n");
    System.out.println("ERRORS "+BER[1]+"-- PAKETA "+BER[0]);
    modem.close();
}
```

```
public void echo_packets(Modem mod, byte[] echo_code,PrintWriter txt) {
    mod.write(echo_code);
    long endTime=0;
    long responseTime=0;
    int readInt;
    int packetsCounter=0;

    long startTime=System.currentTimeMillis();//Αποθηκεύουμε ποτε ξεκίνησε ακριβώς η λήψη
του πακέτου

    for(packetsCounter=0;packetsCounter<35;packetsCounter++) {
        try {
            readInt=mod.read();

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

            if(readInt==-1) {
                break;}

        }

        catch (Exception x) {
            break;
        }

    }

    endTime=System.currentTimeMillis();

    responseTime=endTime-startTime;
}
```

Δίκτυα Υπολογιστών 1
AEM 9124
Γιώργος Εφραίμ Παππάς

```
txt.println(responseTime);

}
public void first_dial(Modem mod,byte[] echo_code) {
    for(int i=0;i<205;i++) {
        System.out.print((char)mod.read());
    }
}

public void get_image(Modem mod,byte[] im_code,FileOutputStream imageOut) throws
IOException {
    int kpre=0;//θα αποθηκευω το προηγουμενο read για τον ελεγχο τελους της εικονας
    int k=0;

    mod.write(im_code);

    for(;;) { //for loop μεχρι να φτασουμε στα bytes αρχης της εικονας (ffd8)
        kpre=k;
        k=mod.read();
        if(kpre==255 && k==216)break;
        if(k==-1)break;
    }

    kpre=0;
    k=0;

    imageOut.write(255);
    imageOut.write(216);

    for (;;) {

        try {

            k=mod.read();

            if(kpre==255)if(k==217) {

                imageOut.write(k);

                break;
            }
            kpre=k;

            imageOut.write(k);

            if (k==-1) break;
        }

        catch (Exception x) {
```

Δίκτυα Υπολογιστών 1
ΑΕΜ 9124
Γιώργος Εφραίμ Παππάς

```
        break; }

    }

}

public long[] gpsGA(Modem mod,byte[] gps_code) {//επιστρέφει πίνακα 2 διαστάσεων με 99
γραμμές για τα 99 traces και 81 στήλες ώστε σε κάθε στήλη να έχω από έναν ASCII για το
εκάστοτε trace
    mod.write(gps_code);
    int endOfpacket=0;
    int k=0;
    int packets[][]=new int [99][81];//αποθηκεύει σε int όλους τους ascii των 99 ιχνων
    int counter1=-1;//μεταβλητή που θα τρέξει στον πίνακα packets στις γραμμές και θα
αλλάζει με βάση το αν πήραμε το $ που δηλώνει την έναρξη του ιχνους
    int counter2=0;
    int coordsToInt[][]=new int [99][17];//αποθηκεύει μοίρες λεπτά από long/lat σε μια δομή
του τύπου 40376985022374813
    int seconds[][]=new int[99][2];//αποθηκεύει όλα τα δευτερολεπτά των 99 ιχνων
    int allCinSec[][]=new int[99][2];//αποθηκεύει από τα 99 ιχνη τα λεπτά και τα
δευτερολεπτά σε δευτερολεπτά ώστε να γίνει η σύγκριση για τα 4+ δευτερολεπτά διαφορά που
ζητάει η εκφώνηση στα τελικά ιχνη
    long T[]=new long[6];//αποθηκεύει τους 6 κωδικούς T που θα πάρω στο τέλος
    for(int l=0;l<27;l++) {//για να εκτυπωθεί το αρχικό μήνυμα πριν τα ιχνη
        try {

            k=mod.read();

            if (k==-1) break;

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

        }

        catch (Exception x)

        {

            break;    }

    }

    for (;;) {
        try {

            k=mod.read();

            if(k==42)endOfpacket++;// αν το k δίνει το αστεράκι στο τέλος του κάθε ιχνους η
endOfpacket αυξάνεται κατά 1 ώστε όταν φτάσω στα 99 να σταματήσω να γράφω στον packets

            if(endOfpacket==99)break;

            if(k==36)//36 Ascii = $
            {
                counter1++;//πάμε στην επόμενη γραμμή του packets μιας και είδαμε το σύμβολο αρχής
του ιχνους.
                counter2=0;
            }
        }
    }
}
```

Δίκτυα Υπολογιστών 1
AEM 9124
Γιώργος Εφραίμ Παππάς

```
    }
    packets[counter1][counter2]=k;
    counter2++;

    if (k==-1) break;

    System.out.print((char)k);
}
catch (Exception x)
{
    break;    }
}

System.out.println();
System.out.println();
System.out.println();
for(int i=0;i<99;i++) {// σε αυτη τη φορ αποθηκευουμε σε int τις τιμες των
συντεταγμενων
    int nullcheck=0; //χρησιμοποιειται για να μην παρεμβαλεται κενος χαρακτηρας αναμεσα στο
    Long και lat

    for(int j=18;j<27;j++) {//long

        if(packets[i][j]==46)
        {nullcheck=1;
        continue;

        }

        coordsToInt[i][j-18-nullcheck] = packets[i][j]-48;

    }
    nullcheck=0;
    for(int j=30;j<40;j++) {//lat

        if(packets[i][j]==46)
        {nullcheck=1;
        continue;

        }

        coordsToInt[i][j-22-nullcheck] = packets[i][j]-48;// το -48 γινεται γιατι ειχα τις
συντεταγμενες σε ascii και θελω να τις κανω σε int
    }

}

for(int i=0;i<99;i++) {//τωρα στον seconds εχω τα δευτερολεπτα ακριβως με τη σειρα
που τα θελει ο πινακας T στη μορφη πχ 36 75
    seconds[i][0]=((int)((coordsToInt[i][13]*1000+coordsToInt[i][14]*100+coordsToInt[i]
[15]*10+coordsToInt[i][16])*0.006));
    seconds[i][1]=((int)((coordsToInt[i][4]*1000+coordsToInt[i][5]*100+coordsToInt[i]
[6]*10+coordsToInt[i][7])*0.006));
```

Δίκτυα Υπολογιστών 1
ΑΕΜ 9124
Γιώργος Εφραίμ Παππάς

```
}

//Θελουμε στις τιμες των T τουλαχιστον 4 δευτερολεπτα διαφορα (στα 6 στοιχεια που θα
παρω εγω δηλαδη)
for(int i=0;i<99;i++) {
    allCinSec[i][0]=((coordsToInt[i][2]*10+coordsToInt[i][3])*60)+seconds[i][1];//πχ
πρωτη γραμμη 37,6985--> 37*60+(int)0,006*6985=2261 δευτερολεπτα
    allCinSec[i][1]=((coordsToInt[i][11]*10+coordsToInt[i][12])*60)+seconds[i][0];
}

int mapForTime[][]=new int[99][1];//περιεχει τους χρονους των 99 πακετων σε sec ωστε
να βρουμε παρακατω αυτα που διαφερουν κατα 4 sec
for(int i=0;i<99;i++) {//ωρες σε sec -----λεπτα σε sec----δευτερολεπτα-----
    mapForTime[i][0]=((packets[i][7]-48)*10+(packets[i][8]-48))*3600+((packets[i][9]-
48)*10+(packets[i][10]-48))*60+((packets[i][11]-48)*10+(packets[i][12]-48));
}

int coordsToIntPOS[]=new int[6];// κραται τις θεσεις των ιχνων που μας κανουν
coordsToIntPOS[0]=0;
int counterPOS=0;
int ps=1;

for(int i=0;i<99;i++) {

    if(Math.abs(mapForTime[counterPOS][0]-mapForTime[i][0])>=12) {
        counterPOS=i;
        coordsToIntPOS[ps]=i;
        ps++;
    }

    if(ps==6)break;
}

for(int i=0;i<6;i++) {//γεμιζουμε τον T με τους απαραιτητους κωδικους για την εικονα
    T[i]=(long) (coordsToInt[coordsToIntPOS[i]][9]*Math.pow(10,11)
        +coordsToInt[coordsToIntPOS[i]][10]*Math.pow(10,10)
        +coordsToInt[coordsToIntPOS[i]][11]*Math.pow(10,9)
        +coordsToInt[coordsToIntPOS[i]][12]*Math.pow(10,8)
        +seconds[coordsToIntPOS[i]][0]*Math.pow(10,6)
        +coordsToInt[coordsToIntPOS[i]][0]*Math.pow(10,5)
        +coordsToInt[coordsToIntPOS[i]][1]*Math.pow(10,4)
        +coordsToInt[coordsToIntPOS[i]][2]*Math.pow(10,3)
        +coordsToInt[coordsToIntPOS[i]][3]*Math.pow(10,2)
        +seconds[coordsToIntPOS[i]][1]);
}
//ΝΑ ΘΥΜΗΘΩ ΟΤΙ ΑΝ ΠΑΡΩ ΧΡΟΝΙΚΗ ΔΙΑΦΟΡΑ ΜΙΚΡΗ ΠΧ 4 ΛΟΓΩ ΣΤΡΟΓΓΥΛΟΠΟΙΗΣΕΩΝ ΘΑ
ΕΧΩ ΙΔΙΕΣ ΣΥΝΤΕΤΑΓΜΕΝΕΣ (ΜΟΙΡΕΣ/ΛΕΠΤΑ/ΔΕΥΤΕΡΟΛΕΠΤΑ)
return T;
}
```

Δίκτυα Υπολογιστών 1
AEM 9124
Γιώργος Εφραίμ Παππάς

```
public int XOR(int map[]) {
    int value=map[0]^map[1];

    for(int i=0;i<14;i++)
        value=value^map[i+2];

return value;
}

public int[] ARQ(Modem mod, byte[] ack,byte[] nack,PrintWriter txt,PrintWriter txt2) {
    int ackNackCounter[]= new int [2];//μετράει στην πρώτη θέση όλα τα πακέτα και στη
δεύτερη τις επανεκπομπές (δηλαδή τα ερρορς)

    ackNackCounter[0]=0;
    ackNackCounter[1]=0;

    mod.write(ack);
    ackNackCounter[0]++;

    long endTime=0;
    long responseTime=0;
    int readInt=0;
    int packetArray[]=new int [58];
    int sequence[]=new int [16];
    long startTime=System.currentTimeMillis();//Αποθηκευουμε ποτε ξεκίνησε ακριβώς η ληψη
του πακέτου
    int xorV;
    int FCS;

    for(;;) {
        try {
            for(int l=0;l<58;l++) {
                readInt=mod.read();

                packetArray[l]=readInt;
            }
            for(int i=0;i<16;i++) {

                sequence[i]=packetArray[i+31];

            }
            System.out.println("\n");
            xorV=XOR(sequence);
            FCS=100*(packetArray[49]-48)+10*(packetArray[50]-48)+
(packetArray[51]-48);

            if(xorV==FCS) {
                for(int i=0;i<58;i++) {
                    System.out.print((char)packetArray[i]);}
                break;
            }
        }
    }
}
```


Δίκτυα Υπολογιστών 1
AEM 9124
Γιώργος Εφραίμ Παππάς

```
    }  
    else { System.out.println("ERRORRRRR");  
           mod.write(nack);  
           ackNackCounter[1]++;  
           ackNackCounter[0]++;}  
  
    if(readInt==-1) {  
        break;}  
  
    }  
  
    catch (Exception x) {  
        break;  
    }  
  
    }endTime=System.currentTimeMillis();  
  
    responseTime=endTime-startTime;  
    txt2.println(ackNackCounter[1]);  
    txt.println(responseTime);  
    return ackNackCounter;  
}  
  
}
```