```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();
```

```
61              for(;;) {
62                  try {
63                      oldP= newP;
64                      newP= modem.read();
65                      System.out.print((char)newP);
66                      if((char)newP== 'P') && ((char)oldP== 'O')){ //stOP
67                          end_t= (System.nanoTime()-start_t)/1000000;
68                          break;
69                      }
70                  } catch (Exception e) {
71                      System.out.println("Echo packet missing. ");
72                  }
73              }
74              total_t+= end_t;
75              general.add(String.valueOf(end_t));
76              end_system= (System.nanoTime()-start_system)/1000000;
77              System.out.println(" ");
78          }
79          modem.close();
80          avg_t= total_t/packets;
81          general.add("Total time of receiving process is
    "+String.valueOf(total_t)+"milliseconds");
82          System.out.println("Total time of receiving process is
    "+String.valueOf(total_t)+"milliseconds");
83          general.add("Total linking time with ithaki server is
    "+String.valueOf(end_system)+"milliseconds");
84          System.out.println("Total linking time with ithaki server is
    "+String.valueOf(end_system)+"milliseconds");
85          general.add("Total packets received are "+String.valueOf(packets));
86          System.out.println("Total packets received are "+String.valueOf(packets));
87          general.add("Average time for packet to be received is
    "+String.valueOf(avg_t)+"milliseconds");
88          System.out.println("Average time for packet to be received is
    "+String.valueOf(avg_t)+"milliseconds");
89          BufferedWriter texter = null;
90          try {
91              File echoPackets= new File("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6
    ΕΞΑΜΗΝΟ\\Δίκτυα Ι\\echop"+password+".txt");
92              texter= new BufferedWriter(new FileWriter("D:\\Stavros\\StavrosDIAFORA\
    \Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα Ι\\echop"+password+".txt",true));
93              if(!echoPackets.exists()) {
94                  echoPackets.createNewFile();
95              }
96              int i;
97              for(i= 0; i<general.size(); i++) {
98                  texter.write(String.valueOf(general.get(i)));
99                  texter.newLine();
100             }
101             texter.newLine();
102         } catch(Exception e){
103             System.out.println("File for echo packets not written. ");
104         } finally {
105             try {
106                 if(texter != null)
107                     texter.close();
108             } catch(Exception e){
109                 System.out.println("BufferedWriter not turned off. "+e);
110             }
111         }
112         modem.close();
```

```
113         echoRequestCode.close();
114         System.out.println("Function giveEchoPackets executed.");
115     }
116 
117     public void giveImage(int mode) { //erwthma 4 gia image
118         String password;
119         Scanner imageRequestCode= new Scanner(System.in);
120         System.out.println("Insert the 4 digits image request code. ");
121         password= imageRequestCode.nextLine();
122         Modem modem;
123         modem= giveModem();
124         String form;
125         if(mode== 0) {
126             form= "M"; //no errors
127         }
128         else {
129             form= "G"; //with errors
130         }
131         modem.write((form+password+"\r").getBytes());
132         int old_val= 0,new_val= 0;
133         ArrayList<Integer> image= new ArrayList<Integer>(); //akeraious ascii
134         for(;;) {
135             try {
136                 old_val= new_val;
137                 new_val= modem.read();
138                 System.out.println(new_val);
139                 image.add(new_val);
140                 if((new_val== 217)&&(old_val== 255)) //thelw delimiter gia 0xFF=255 kai
    0xD9=217
141                     break;
142             }catch(Exception e){
143                     System.out.println("Image not retrieved. ");
144             }
145         }
146         String source;
147         if(mode== 0) {
148             source= ("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα Ι\
    \imageNER"+password+".jpeg");
149         }
150         else {
151             source= ("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα Ι\
    \imageER"+password+".jpeg");
152         }
153         try {
154             FileOutputStream image_doc= new FileOutputStream(source);
155             int i;
156             for(i= 0; i< image.size(); i++) {
157                 image_doc.write(image.get(i));
158             }
159             image_doc.close();
160          catch(Exception e) {
161             System.out.println("File for image not written. ");
162         }
163         modem.close();
164         System.out.println("Function giveImage executed.");
165     }
166 
167     public void giveGPS() { //erwthma 5 gia gps
168         String password;
169         Scanner gpsRequestCode= new Scanner(System.in);
```

```
170         System.out.println("Insert the 4 digits GPS request code. ");
171         password= gpsRequestCode.nextLine();
172         int old_val= 0,new_val= 0;
173         float[] length= new float[6];//mhkos
174         float[] lengthD= new float[6];//gwnia moirwn mhkous
175         float[] width= new float[6];//platos
176         float[] widthD= new float[6];//gwnia moirwn platous
177         int c= 0; //counter gia ton katw
178         int[] gps_pu= new int[80]; //gps per unit array
179         int[][] wid_len= new int[2][10] ; //proswrina gia 0platos kai 1mhkos
180         //$GPGGA,045208.000,4037.6331,N,02257.5633,E,1,07,1.5,57.8,M,36.1,M,,0000*6D
181         //       wra        D platos     D mhkos
182         //thelw na ta balw ola se ena T string pinaka AABBCCDDEEZZ ara pinakaki
183         ArrayList<String> Tall= new ArrayList<String>(); //kratw oles tis parametrous T
184         //String T= new String();
185         String[] Tmini= new String[6];
186         Modem modem;
187         modem= giveModem();
188         modem.write(("P"+password+"R=1000199"+"\r").getBytes()); //R=XPPPPLL
189         for(;;) {
190             try {
191                 old_val= new_val;
192                 new_val= modem.read();
193                 gps_pu[c]= new_val;
194                 System.out.print((char)gps_pu[c]);
195                 c++;
196             if(((char)new_val== 'G')&&((char)old_val== 'N')){ //meta to trackiNG
197                 continue;
198             }
199             if(((char)new_val== 'P')&&((char)old_val== 'O')){ //stamataw sto telos stOP
200                 break;
201             }
202             if((new_val== 10)&&(old_val== 13)) { //ksekinaw otan <CR> kai <LF>
203                 int i;
204                 for(i= 0; i<10; i++) {
205                     wid_len[0][i]= gps_pu[i+18]; //platos erxetai prwta meta thn
    wra.000,...
206                     wid_len[1][i]= gps_pu[i+30]; //mhkos meta to platos kai to N,
207                 }
208                 //apo ascii ta kanw akeraious kai meta float . 0=ascii 48
209                 //antlhsh
210                 //platos
211                 widthD[0]= (float) (wid_len[0][0]-48)*10;
212                 widthD[1]= (float) (wid_len[0][1]-48);
213                 width[0]= (float) (wid_len[0][2]-48)/10;
214                 width[1]= (float) (wid_len[0][3]-48)/100;
215                 width[2]= (float) (wid_len[0][5]-48)/1000; //phdaw .
216                 width[3]= (float) (wid_len[0][6]-48)/10000;
217                 width[4]= (float) (wid_len[0][7]-48)/100000;
218                 width[5]= (float) (wid_len[0][8]-48)/1000000;
219                 //mhkos
220                 lengthD[0]= (float) (wid_len[1][1]-48)*10; //
221                 lengthD[1]= (float) (wid_len[1][2]-48);
222                 length[0]= (float) (wid_len[1][3]-48)/10;
223                 length[1]= (float) (wid_len[1][4]-48)/100;
224                 length[2]= (float) (wid_len[1][6]-48)/1000; //phdaw .
225                 length[3]= (float) (wid_len[1][7]-48)/10000;
226                 length[4]= (float) (wid_len[1][8]-48)/100000;
227                 length[5]= (float) (wid_len[1][9]-48)/1000000;
228                 //kataxwrhsh kai apokwdikopoihsh tou dekadikou systhmatos apo pshfia se
```

```
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;
```

```
285                 }
286             }catch (Exception e) {
287                 System.out.println("GPS pin image not retrieved. ");
288             }
289         }
290         String source= ("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα Ι\
    \imageGPS.jpeg");
291         try {
292             FileOutputStream image_doc= new FileOutputStream(source);
293             int i;
294             for(i= 0; i< imageGPS.size(); i++) {
295                 image_doc.write(imageGPS.get(i));
296             }
297             image_doc.close();
298         }catch (Exception e) {
299             System.out.println("File for GPS image not written. ");
300         }
301         modem.close();
302         gpsRequestCode.close();
303         System.out.println("Function giveGPS executed.");
304     }
305 
306     public void giveARQ() { //erwthmata 6,7,8 QXXXX gia ack , RXXXX gia nack
307         //PSTART DD-MM-YYYY HH-MM-SS PC <XXXXXXXXXXXXXXXX> FCS PSTOP
308         String passwordQ;
309         Scanner ackResultCode= new Scanner(System.in);
310         System.out.println("Insert the 4 digits ACK result code. ");
311         passwordQ= ackResultCode.nextLine();
312         String passwordR;
313         Scanner nackResultCode= new Scanner(System.in);
314         System.out.println("Insert the 4 digits NACK result code. ");
315         passwordR= nackResultCode.nextLine();
316         int old_val= 0, new_val= 0, packets= 0, error_packets= 0;
317         double start_t= 0,end_t= 0,total_t= 0,avg_t= 0;
318         double start_system= 0,end_system= 0;
319         ArrayList<String> general= new ArrayList<String>();
320         Modem modem;
321         modem= giveModem();
322         int XOR= 0,FCS= 0;
323         int m= 0, retrans= 0; //m epanalhpseis gia na stal8ei to paketo, retrans an
    stal8hke swsta amesws h xreiazetai retrans
324         int[] m_times= new int[10]; //m=0 , m=1 ...
325         start_system= System.nanoTime();
326         //tests gia 4 lepta se ms
327         while(end_system< 1000*4*60) {
328             modem.write(("Q"+passwordQ+"\r").getBytes());
329             packets++;
330             start_t= System.nanoTime();
331             for(;;) {
332                 try {
333                     old_val= new_val;
334                     new_val= modem.read();
335                     System.out.print((char)new_val);
336                     if(old_val== '<') { //pairnw ta X...X kai metraw XOR
337                         while(new_val!= '>') {
338                             XOR= XOR^ (char)new_val;
339                             new_val= modem.read();
340                             System.out.print((char)new_val);
341                         }
342                         System.out.print("XOR="+XOR);
```

```
343                     }
344                     if((new_val== ' ')&&(old_val== '>')) { //metraw FCS
345                         new_val= modem.read();
346                         FCS= Character.getNumericValue((char)new_val)* 100;
347                         new_val= modem.read();
348                         FCS= FCS+ Character.getNumericValue((char)new_val)* 10;
349                         new_val= modem.read();
350                         FCS= FCS+ Character.getNumericValue((char)new_val);
351                         System.out.print("FCS="+FCS);
352                     }
353                     if((new_val== 'P')&&(old_val== 'O')) { //stOP
354                         System.out.println(" ");
355                         break;
356                     }
357                 }catch(Exception e) {
358                     System.out.println("ACK mode packet not retrieved. ");
359                 }
360             }
361             m= 0;
362             retrans= 0;
363             while(XOR!= FCS) {
364                 if(retrans== 0) {
365                     error_packets++;
366                 }
367                 retrans= 1; //freno gia error_packets++
368                 XOR= 0;
369                 FCS= 0;
370                 modem.write(("R"+passwordR+"\r").getBytes());
371                 m++;
372                 for(;;) {
373                     try {
374                         old_val= new_val;
375                         new_val= modem.read();
376                         System.out.print((char)new_val);
377                         if(old_val== '<') { //pairnw ta X...X kai metraw XOR
378                             while(new_val!= '>') {
379                                 XOR= XOR^ (char)new_val;
380                                 new_val= modem.read();
381                                 System.out.print((char)new_val);
382                             }
383                             System.out.print("XOR="+XOR);
384                         }
385                         if((new_val== ' ')&&(old_val== '>')) { //metraw FCS
386                             new_val= modem.read();
387                             FCS= Character.getNumericValue((char)new_val)* 100;
388                             new_val= modem.read();
389                             FCS= FCS+ Character.getNumericValue((char)new_val)* 10;
390                             new_val= modem.read();
391                             FCS= FCS+ Character.getNumericValue((char)new_val);
392                             System.out.print("FCS="+FCS);
393                         }
394                         if((new_val== 'P')&&(old_val== 'O')) { //stOP
395                             System.out.println(" ");
396                             break;
397                         }
398                     }catch(Exception e) {
399                         System.out.println("NACK mode packet not retrieved. ");
400                     }
401                 }
402             }
```

```
403             m_times[m]+= 1;
404             end_t= (System.nanoTime()- start_t)/ 1000000;
405             total_t+= end_t;
406             general.add(String.valueOf(end_t));
407             end_system=(System.nanoTime()- start_system)/ 1000000;
408         }
409         //L=16chars=16x8=128
410         //PER = (1-BER)^L
411         //PER = ACK / (ACK + NACK)
412         double per= 0, ber= 0;
413         int nack= 0;
414         int i;
415         for(i= 1; i<10; i++) {
416             nack+=m_times[i]*i;
417         }
418         per=(double) packets/(packets+nack); //isws thelei error_packets
419         ber=1-(Math.pow(per,1.0/128.0));
420         //System.out.println("\nPackets received: "+packets);
421         //System.out.println("Total linking time with ithaki server: "+total_t);
422 
423         avg_t= total_t/packets;
424         general.add("Total time of receiving process is
    "+String.valueOf(total_t)+"milliseconds");
425         System.out.println("Total time of receiving process is
    "+String.valueOf(total_t)+"milliseconds");
426         general.add("Total linking time with ithaki server is
    "+String.valueOf(end_system)+"milliseconds");
427         System.out.println("Total linking time with ithaki server is
    "+String.valueOf(end_system)+"milliseconds");
428         general.add("Total packets calls received are "+String.valueOf(packets+nack));
429         System.out.println("Total packets calls received are
    "+String.valueOf(packets+nack));
430         general.add("Average time for packet to be received is
    "+String.valueOf(avg_t)+"milliseconds");
431         System.out.println("Average time for packet to be received is
    "+String.valueOf(avg_t)+"milliseconds");
432         general.add("Total NACK calls are "+nack);
433         System.out.println("Total NACK calls are "+nack);
434         general.add("Bit error rate is "+ber);
435         System.out.println("Bit error rate is "+ber);
436         for(i= 0; i<10; i++) {
437             System.out.println(m_times[i]+" packets needed "+i+" requests.");
438             general.add(m_times[i]+" packets needed "+i+" requests.");
439         }
440         BufferedWriter texter= null;
441         try {
442             File ARQstats= new File("D:\\Stavros\\StavrosDIAFORA\\Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\
    \Δίκτυα Ι\\ARQ.txt");
443             texter= new BufferedWriter(new FileWriter(("D:\\Stavros\\StavrosDIAFORA\
    \Πανεπιστήμιο\\6 ΕΞΑΜΗΝΟ\\Δίκτυα Ι\\ARQ.txt"),true));
444             if(!ARQstats.exists()) {
445                 ARQstats.createNewFile();
446             }
447             int j;
448             for(j= 0; j<general.size(); j++) {
449                 texter.write(String.valueOf(general.get(j)));
450                 texter.newLine();
451             }
452             texter.newLine();
453         } catch(Exception e){
```

```
454             System.out.println("File for ARQ statistics not written. ");
455         } finally {
456             try {
457                 if(texter != null)
458                     texter.close();
459             } catch(Exception e){
460                 System.out.println("BufferedWriter not turned off. "+e);
461             }
462         }
463         modem.close();
464         ackResultCode.close();
465         nackResultCode.close();
466         System.out.println("Function giveARQ executed. ");
467     }
468 
469     public static void main String[] args) {
470         Scanner order= new Scanner(System.in);
471         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.");
472         for(;;) {
473 
474             try {
475                 int key= order.nextInt();
476                 switch(key) {
477                 case 1:
478                     new userApplication().giveEchoPackets();
479                     continue;
480                 case 2:
481                     new userApplication().giveImage(0);
482                     continue;
483                 case 3:
484                     new userApplication().giveImage(1);
485                     continue;
486                 case 4:
487                     new userApplication().giveGPS();
488                     continue;
489                 case 5:
490                     new userApplication().giveARQ();
491                     continue;
492                 case 6:
493                     System.out.println("Bye bye user.Hope you enjoyed the virtual ride :) ");
494                     break;
495                 }
496                 if(key== 6) {
497                     break;
498                 }
499             }catch(Exception e) {
500                 System.out.println("Invalid key given. ");
501                 break;
502             }
503         }
504     }
505 }
506 
```