```
import java.io.*;
import java.util.Scanner;
import java.util.Arrays;
import java.util.ArrayList;
import java.io.FileWriter;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.File;
import java.io.FileOutputStream;


public class userApplication {
 public static void main(String[] param) {
             new userApplication().menu();
 }

 public void menu(){
   Scanner input = new Scanner(System.in);
   for(;;){
     int select;
     System.out.print("\nSelect from the following:\n1. Echo Request ");
     System.out.print("\n2. Image error free\n3. Image with errors\n4. Gps\n5. ARQ\n6. Exit\n");
     try{
       select=input.nextInt();
       if (select==1){
         echo();
       }else if(select==2){
         image(0);
       }else if(select==3){
         image(1);
       }else if(select==4){
          gps();
       }else if(select==5){
          ARQ();
       }else if(select==6){
          System.out.print("Programm exit! \n");
         return;
       }else{
          System.out.print("You have selected none of the listed options, try again \n\n");
       }
     }catch(Exception x){
          System.out.print("You have selected none of the listed options, try again \n\n");
     }
   }
 }
public Modem createModem(){
  int data;
  Modem modem;
  modem=new Modem();
  modem.setSpeed(82000);
  modem.setTimeout(20000);
  System.out.print("Trying to connect with server... \n\n");
  modem.write("atd2310ithaki\r".getBytes());
  for(;;){
    try{
      data=modem.read();
      if(data==-1) break;
      System.out.print((char)data);
    }catch (Exception x) {
      break;
    }
  }
  return modem;
}
public void echo( ){

    System.out.print("Recieving echo packets\n");
    //CODE SCANNER
    ArrayList<String> filewrite = new ArrayList<String>();
    Scanner input = new Scanner(System.in);
    String code;
    int numberOfPackets=0;
    System.out.print("Enter the echo code from netlab site:EXXXX\n");
    code=input.nextLine();
    Modem modem;
    modem=createModem();
    //data input from server
    int data=0,prevData=0; //prevData is used in order to stop the recieve
    double startTime=0,endTime=0,totalTime=0,avg=0;
    double startloop=0,endloop=0;
    startloop = System.nanoTime();
    while(endloop<(5*60*1000)){
    modem.write(("E"+code+"\r").getBytes());
    numberOfPackets++;
    startTime = System.nanoTime();
    for (;;) {
      try {
        prevData=data;
        data=modem.read();
        System.out.print((char)data );
        if(prevData=='O' && data=='P'){
          endTime=(System.nanoTime()- startTime)/1000000;
          System.out.print("    Time to recieve this packet was "+endTime+"\n");
        break;
        }

      } catch (Exception x) {
        break;
```

```
     }
    }

    filewrite.add("Time to recieve this packet was "+endTime+"\n");

    totalTime+=endTime;
    endloop=(System.nanoTime()-startloop)/1000000;

  }



  avg=totalTime/numberOfPackets;
  filewrite.add("Number of packets : "+String.valueOf((double)numberOfPackets));
  filewrite.add("Average time : "+String.valueOf(avg));
  filewrite.add("Total time of communication : "+(totalTime/60)/1000+" minutes\n");
  filewrite.add("Total time of test : "+(endloop/60)/1000+" minutes\n");

  BufferedWriter bw = null;
  try{
    File file =new File(("Echo"+code+".txt"));
    bw = new BufferedWriter(new FileWriter(("Echo"+code+".txt"), true));
    if(!file.exists()){
      file.createNewFile();
    }
    for (int i=0; i <filewrite.size(); i++){

      bw.write(String.valueOf(filewrite.get(i)));
      bw.newLine();
    }
    bw.newLine();
  }catch(IOException ioe){
    ioe.printStackTrace();
  }
  finally{
    try{
      if(bw != null) bw.close();
    }catch(Exception ex){
      System.out.println("Error in closing the BufferedWriter" + ex);
    }
  }

  System.out.println("");
            System.out.print("Average packet recieve time:"+avg+" milliseconds for "+numberOfPackets+ " packets\n");
            System.out.print("\r\n");
modem.close();
}


public void image(int error){
```

```
Scanner input = new Scanner(System.in);
String code;
String letter;
if(error==0){
 letter="M";
 System.out.print("Recieving error free image\n");
}else{
 letter="G";
 System.out.print("Recieving image with errors\n");
}
System.out.print("Enter the image code from netlab site:"+letter+"XXXX\n");
code=input.nextLine();
Modem modem;
modem=createModem();
//data input from server
int data=0,prevData=0; //prevData is used in order to stop the recieve
ArrayList<Integer> file = new ArrayList<Integer>();
int[] endOfImage; //dec of hex 0xFF=255 and 0xD9=217
//create modem

double totalTime=0;
double startTime=System.currentTimeMillis();
String fileName;

modem.write((letter+code+"\r").getBytes());
for (;;) {
 try {
 prevData=data;
 data=modem.read();
 endOfImage=new int[]{prevData,data};
 file.add(data);
 if (Arrays.equals(endOfImage, new int[]{255,217})){
  totalTime=System.currentTimeMillis()-startTime;

 }
 if(data==-1){
  break;
 }
 } catch (Exception x) {
 break;
 }
}
System.out.print("Time to recieve the image: "+totalTime+" milliseconds\n");
if(error==0){
 letter="M";
 fileName = ("image"+code+".jpeg");
}else{
    letter="G";
    fileName = ("errorImage"+code+".jpeg");
  }
  try{
     FileOutputStream out = new FileOutputStream(fileName);

    for (int i=0; i <file.size(); i++){
      out.write(file.get(i));
    }
    out.close();
  }
  catch(IOException ex){
    System.out.println("error writing file");
  }
  System.out.println("");
modem.close();
}


public void gps(){

  int data=0,prevData=0; //prevData is used in order to stop the recieve
  double startTime=0,totalTime=0;
  int k;

  Scanner input = new Scanner(System.in);
  String code;
  int samples;
  System.out.print("Enter the GPS code from netlab site:PXXXX\n");
  code=input.nextLine();
  System.out.print("Enter the number of GPS samples (01 - 99) to take :\n");
  samples=input.nextInt();
  Modem modem;
  modem=createModem();
  String time="";
  String longitude="";
  String longitudeSecond="";
  String latitude="";
  String latitudeSecond="";
  String timeArray[]=new String[samples];
  String longitudeArray[]=new String[samples];
  String longitudeSecondArray[]=new String[samples];
  String latitudeArray[]=new String[samples];
  String latitudeSecondArray[]=new String[samples];



  String temp;
  int flag=0,counter=0,mhkosCounter=0,platosCounter=0;
  int timeFlag=0,platosFlag=0,mhkosFlag=0,j=0;
```

```
modem.write(("P"+code+"R=10001"+Integer.toString(samples)+"\r").getBytes());
System.out.print("Recieving GPS packet\n");
for (;;) {
 try {
   prevData=data;
   data=modem.read();
   System.out.print((char)data );
   if(prevData=='$'&&data=='G'){
     counter=0;
     mhkosCounter=0;
     platosCounter=0;
     timeFlag=0;
     platosFlag=0;
     mhkosFlag=0;
     longitude="";
     time="";
     longitudeSecond="";
     latitude="";
     latitudeSecond="";

   }
   if(data==','){
     counter++;
   }
   if((counter==1)&&(data!=',')&&(timeFlag==0)){
     if(data!='.'){
       time+=(char)data;
     }else{
       timeFlag=1;
     }

   }
   if(counter==2&&platosFlag==0){
     if(data!=','&&data!='.'){
       if(platosCounter<4){
         latitude+=(char)data;
       }else if(platosCounter>=4&&platosCounter<=7){
         latitudeSecond+=(char)data;
         if(platosCounter==7){
            platosFlag=1;
         }
       }
       platosCounter++;
     }
   }
   if(counter==4&&mhkosFlag==0){
     if(data!=','&&data!='.'){
       if(mhkosCounter<5&&mhkosCounter>0){
         longitude+=(char)data;
```

```
        }else if(mhkosCounter>=5&&mhkosCounter<=8){
          longitudeSecond+=(char)data;
          if(mhkosCounter==8){
            mhkosFlag=1;
          }
        }
        mhkosCounter++;
      }
    }
    if((platosFlag==1)&&(mhkosFlag==1)){
      longitudeArray[j]=longitude;
      longitudeSecondArray[j]=longitudeSecond;
      latitudeArray[j]=latitude;
      latitudeSecondArray[j]=latitudeSecond;
      timeArray[j]=time;
      timeFlag=0;
      platosFlag=0;
      mhkosFlag=0;
      j++;

    }
    if(prevData=='N' && data=='G' && flag==0){
      startTime = System.nanoTime();
      flag++;
    }
    if(prevData=='S' && data=='T' && flag==1){
      totalTime = System.nanoTime()-startTime;
      flag++;

    }
    if(data==-1){
      break;
    }
  } catch (Exception x) {
    break;
  }
}
totalTime=totalTime/1000000;
System.out.print("\nTime to recieve GPS packet : "+totalTime+" milliseconds\n\n");
temp= "P"+code;
int intTime;
float  tmp;

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

    intTime = Integer.parseInt(timeArray[i]);
    int hour = intTime/10000;
    int minutes = intTime/100;
    minutes = minutes % 100;
  int seconds = intTime % 100;
  intTime = seconds + (minutes*60) + (hour*3600);
  timeArray[i]=Integer.toString(intTime);



  //metatroph apo dekadikh morfh se moires lepta kai deytera lepta
  tmp= Float.parseFloat(longitudeSecondArray[i]); //=tmp2
  tmp=((tmp/10000)*60); //apomononw ta lepta

  longitudeSecondArray[i]=Integer.toString((int)tmp);
  longitudeArray[i]+=longitudeSecondArray[i];

  tmp = Float.parseFloat(latitudeSecondArray[i]);
  tmp=(tmp/10000)*60;

  latitudeSecondArray[i]=Integer.toString((int)tmp);
  latitudeArray[i]+=latitudeSecondArray[i];

}
int start=Integer.parseInt(timeArray[0]);
int z=0;
int timeDifference;
if(samples<=20){
  timeDifference=5;
}else{
  timeDifference=samples/5;
}
for(int i=0;i<samples;i++){
  if( (Integer.parseInt(timeArray[i]) - start)>=timeDifference &&z<5 ){
    temp+= "T="+longitudeArray[i]+latitudeArray[i];
    start=Integer.parseInt(timeArray[i]);
    z++;
  }
  if(z==5){
    break;
  }
}
temp+="\r\n";
System.out.println("Sent to ithaki: "+temp+"\n");
modem.write((temp).getBytes());

ArrayList<Integer> file = new ArrayList<Integer>();
for (;;){
  try {
    k=modem.read();
    file.add(k);
    if (k==-1) break;
  } catch (Exception x) {
    break;
```

```
      }
    }
    String fileName = (("GPS"+code+".jpeg"));
    try{
      FileOutputStream out = new FileOutputStream(fileName);

      for(int i = 0 ; i < file.size() ; i++){
        out.write(file.get(i));
      }
      out.close();
     }
     catch(IOException ex){
      System.out.println("error writing file");
     }
     System.out.println("");

modem.close();
}

public void ARQ(){

  Scanner input = new Scanner(System.in);
  int data=0,prevData=0,flag=0,samples=0;
  int numTry[]=new int[10];
  for(int i=0;i<10;i++){
    numTry[i]=0;
  }
  String ack, nack;
  int xor=0,fcs=0,fcscounter=0,z=0,packetErrors=0;
  int addFlag=0,exitFlag=0,stringlengh=59;
  double startTime=0,endTime=0,totalTime=0;
  double startloop=0,endloop=0;

  ArrayList<String> filewrite = new ArrayList<String>();
  System.out.print("Enter the Ack code from netlab site:QXXXX\n");
  ack=input.nextLine();
  System.out.print("Enter the Nack code from netlab site:RXXXX\n");
  nack=input.nextLine();
  Modem modem;
  modem=createModem();
  startloop=System.nanoTime();
  while (endloop<(5*60*1000)){//
    modem.write(("Q"+ack+"\r").getBytes());
    samples++;
    startTime=System.nanoTime();
    for(;;){
      try{
        prevData=data;
        data=modem.read();
      System.out.print((char)data);

      if(flag==1&&data!='>'){
        xor=xor^(char)data;
      }

      if ((flag==2)&&(fcscounter<=3)){
        if (fcscounter>=1){
          fcs=10*fcs;
          fcs=fcs+Character.getNumericValue((char)data);
        }
        fcscounter++;
      }

      if(data=='<'){
        flag=1;
      }
      if(data=='>'){
        flag=2;
      }
      if(prevData==' ' && data=='P'){

        exitFlag=1;
      }
      if(prevData=='O' && data=='P' && exitFlag==1){

        break;
      }

    }catch (Exception x) {
      break;
    }
  } //for
  z=0;
  data=prevData=0;
  if(fcs!=(int)xor){
    packetErrors++;
    addFlag=1;
  }
  while(fcs!=(int)xor){
    //System.out.print(" Xor = "+xor+" while fcs = "+fcs+"\n");

    xor=0;
    flag=0;
    exitFlag=0;
    modem.write(("R"+nack+"\r").getBytes());
    for(;;){
      try{
        prevData=data;
        data=modem.read();

        if(flag==1&&data!='>'){
          xor=xor^(char)data;
        }

        if(data=='<'){
          flag=1;
        }
        if(data=='>'){
          flag=2;
        }
        if(prevData==' ' && data=='P'){
          exitFlag=1;
        }
        if(prevData=='O' && data=='P' && exitFlag==1){

          break;
        }
        //System.out.print((char)data);
      }catch (Exception x) {
        break;
      }
    }
    z++;
}//while
if(addFlag==1){
  numTry[z]=numTry[z]+1;
  addFlag=0;
}
fcscounter=0;
fcs=0;
xor=0;
flag=0;
exitFlag=0;


  endTime=(System.nanoTime()- startTime)/1000000;
  filewrite.add("Time to recieve this packet was "+endTime+"\n");
  totalTime+=endTime;
  endloop=(System.nanoTime()-startloop)/1000000;
  System.out.print(" Time to recieve "+samples+" packet was "+endTime+"\n");


}
float ber;
ber=(float)packetErrors/(samples*stringlengh);
System.out.print("\n\nRecieved "+samples+" samples.\n");
System.out.print("Total time of communication : "+(totalTime/60)/1000+" minutes\n");
```

```
  System.out.print("Average time to recieve one packet : "+(totalTime)/samples+"\n");
  System.out.print("There was error in "+packetErrors+" out of "+samples+" packets\n");

  filewrite.add("\n\nRecieve "+samples+" samples.\n");
  filewrite.add("Total time of communication : "+(totalTime/60)/1000+" minutes\n");
  filewrite.add("Average time to recieve one packet : "+(totalTime)/samples+"\n");
  filewrite.add("There was error in "+packetErrors+" out of "+samples+" packets\n");

  for (int i =1;i<10;i++){
    System.out.print("The number of packets that needed "+(i)+" Nack requests was : "+(numTry[i])+"\n");
    filewrite.add("The number of packets that needed "+(i)+" Nack requests was : "+(numTry[i])+"\n");
  }

  System.out.print("Total time of this test "+(endloop/60)/1000+"\n");
  System.out.print("Bit error rate : "+ber);
  filewrite.add("Total time of this test "+(endloop/60)/1000+"\n");
  filewrite.add("Bit error rate : "+ber);
  filewrite.add("\n");

  BufferedWriter bw = null;
  try{
    File file =new File(("aqr"+ack+".txt"));
    bw = new BufferedWriter(new FileWriter(("aqr"+ack+".txt"), true));

    if(!file.exists()){
      file.createNewFile();
    }
    for (int i=0; i <filewrite.size(); i++){

      bw.write(String.valueOf(filewrite.get(i)));
      bw.newLine();
    }
    bw.newLine();
  }catch(IOException ioe){
    ioe.printStackTrace();
  }
  finally{
    try{
      if(bw != null) bw.close();
    }catch(Exception ex){
      System.out.println("Error in closing the BufferedWriter" + ex);
    }
  }
  modem.close();
}

}//virtualmodem class
```