Δίκτυα Υπολογιστών 1

Πηγαίος κώδικας εργασίας

Ονοματεπώνυμο: Φόρογλου Γεώργιος-Βησσαρίων
ΑΕΜ: 9557

```
package diktia;

import java.lang.*;
import ithakimodem.Modem;
import java.io.*;
import java.util.Date;
import java.util.concurrent.TimeUnit;

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

public class virtualModem {
	public static void main(String[] param) {

	//Oi sinartiseis trexoyn ksexorista
	//Opoia den trexei mpainei san sxollio

		(new virtualModem()).demo();
		(new virtualModem()).echo();
		(new virtualModem()).image();
		(new virtualModem()).imageWithError();
		(new virtualModem()).gps();
		(new virtualModem()).acknack();
	}

	public void demo() {
		int k;
		Modem modem = new Modem();
		modem.setSpeed(80000);
		modem.setTimeout(2000);
		modem.open("ithaki");
		for (;;) {
			try {
				k = modem.read();
				if (k == -1)
```

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

        public void echo() {

                String command = new String();
                command = "E2283\r";
                Modem modem = new Modem();
                modem.setSpeed(80000);
                modem.setTimeout(2000);
                modem.open("ithaki");
                int a;
                for (;;) {
                        a = modem.read();
                        if (a == -1)
                                break;
                        System.out.print((char) a);
                }

                int k;
                float time;
                PrintWriter responseTimesEcho=null;
                try{
                        responseTimesEcho=new
PrintWriter("responseTimesEcho.txt","UTF-8");
                }catch(Exception x){
                        System.out.println("Raise exception");
                        System.exit(1);
                }
                String echoPacket="";
                long t1,t2,tStart;
                tStart=System.currentTimeMillis();
                while((System.currentTimeMillis()-tStart)<240000){
                        tic=System.currentTimeMillis();
                        modem.write(command.getBytes());
                        for(;;){
                                try{
                                        k=modem.read();
                                        echoPacket=echoPacket+(char)k;
                                        System.out.print((char)k);
                                        if(echoPacket.endsWith("PSTOP")){
                                                t2=System.currentTimeMillis();
                                                time=(float)((float)(t2-t1)/1000);

        responseTimesEcho.write(System.lineSeparator());
```

```
responseTimesEcho.write(Float.toString(time));

responseTimesEcho.write(System.lineSeparator());
                                        break;
                                }
                        }catch(Exception x){
                                System.exit(1);
                        }
                }
                echoPacket="";
        }
        try{
                responseTimesEcho.close();
        }catch(Exception x){
                System.exit(1);
        }
}

public void image() {

        String command = new String();
        command = "M0099\r";
        Modem modem = new Modem();
        modem.setSpeed(80000);
        modem.setTimeout(2000);
        modem.open("ithaki");
        int a;
        for (;;) {
                a = modem.read();
                if (a == -1)
                        break;
                System.out.print((char) a);
        }
        int k;
        boolean counter=false;
        OutputStream out=null;
        try{
                out=new FileOutputStream("E1.jpg");
        }catch(Exception x){
                System.out.println("Openning ERROR!");
                System.exit(1);
                }
        try{
                modem.write(command.getBytes());
        }catch(Exception x){
                System.exit(1);
        }
        for(;;){
                try{
                        k=modem.read();
```

```
                        if(k==-1)break;
                        if(k==0xFF){
                                for(;;){
                                        out.write(k);
                                        k=modem.read();
                                        if(k==0xFF){
                                                out.write(k);
                                                k=modem.read();
                                                if(k==0XD9){
                                                        out.write(k);
                                                        counter=true;
                                                }
                                        }
                                        if(counter) break;
                                }
                        }
                }catch(Exception x){
                        System.exit(1);
                }
                if(counter) break;
        }
        try{
                out.close();
        }catch(Exception x){
                System.exit(1);
        }
}

public void imageWithError() {
        String command = new String();
        command = "G1313\r";
        Modem modem = new Modem();
        modem.setSpeed(80000);
        modem.setTimeout(2000);
        modem.open("ithaki");
        int a;
        for (;;) {
                a = modem.read();
                if (a == -1)
                        break;
                System.out.print((char) a);
        }
        int k;
        boolean counter=false;
        OutputStream out=null;
        try{
                out=new FileOutputStream("E2.jpg");
        }catch(Exception x){
                System.out.println("Openning ERROR!");
                System.exit(1);
                }
```

```
		try{
			modem.write(command.getBytes());
		}catch(Exception x){
			System.exit(1);
		}
		for(;;){
			try{
				k=modem.read();
				if(k==-1)break;
				if(k==0xFF){
					for(;;){
						out.write(k);
						k=modem.read();
						if(k==0xFF){
							out.write(k);
							k=modem.read();
							if(k==0XD9){
								out.write(k);
								counter=true;
							}
						}
						if(counter) break;
					}
				}
			}catch(Exception x){
				System.exit(1);
			}
			if(counter) break;
		}
		try{
			out.close();
		}catch(Exception x){
			System.exit(1);
		}

}

public void gps(){
		String GPS = new String();
		GPS = "P6030=1000199\r";
		Modem modem=new Modem();
		modem.setSpeed(80000);
		modem.setTimeout(2000);
		modem.open("ithaki");
		int e;
		for(;;){
			try{
				e=modem.read();
				if(e==-1)break;
			}catch(Exception x){
				System.exit(1);
```

```
        }
}
OutputStream op=null;
PrintWriter coordinates=null;
boolean counter=false;
OutputStream out=null;
int k;
try{
        out=new FileOutputStream("M1.jpg");
}catch(Exception x){
        System.out.println("Raise Exception");
        System.exit(1);
}
try{
        op=modem.getOutputStream();
}catch (Exception x){
        System.out.println("Raise Exception");
        System.exit(1);
}
try{
        coordinates=new PrintWriter("Coordinates.txt","UTF-8");
}catch(Exception x){
        System.out.println("Creation Error");
        System.exit(1);
}
try{
        op.write(GPS.getBytes());
}catch(Exception x){
        System.out.println("Raise Exception");
        System.exit(1);
}
for(;;){
        try{
                k=modem.read();
                if(k==-1)break;
                coordinates.write((char)k);
                System.out.print((char)k);
        }catch(Exception x){
                System.out.println("Raise Exception");
                System.exit(1);
        }
}
coordinates.close();
try{
        op.close();
        modem.close();
}catch(Exception x){
        System.out.println("Exception Occured ");
        System.exit(1);
}
String line="";
```

```
		BufferedReader reader1=null;
		PrintWriter onlyCoordinates=null;
		try{
			onlyCoordinates=new
PrintWriter("onlyCoordinates.txt","UTF-8");
		}catch(Exception x){
			System.out.println("Exception Occured");
			System.exit(1);
		}
		try{
			reader1=new BufferedReader(new
FileReader("Coordinates.txt"));
		}catch (Exception x){
			System.out.println("Exception Occured");
			System.exit(1);
		}
		try{
			while((line=reader1.readLine())!=null){
				if(line.startsWith("$GPGGA")){
					onlyCoordinates.write(line);

	onlyCoordinates.write(System.lineSeparator());
				}
			}
		}catch(Exception x){
			System.exit(1);
		}
		try{
			reader1.close();
		}catch(Exception x){
			System.out.println("Raise Exception");
			System.exit(1);
		}
		try{
			onlyCoordinates.close();
		}catch(Exception x){
			System.out.println("Raise Exception");
			System.exit(1);
		}
		BufferedReader reader2=null;
		line="";
		try{
			reader2=new BufferedReader(new
FileReader("onlyGPS.txt"));
		}catch(Exception x){
			System.exit(1);
		}
		int Counter=0;
		double[] longtitude=new double[5];
		double[] latitude=new double[5];
		int[] time=new int[5];
```

```
        int sec,min,temp,temp2;
        String TEMP;
        String[][] data=new String[99][15];
        try{
                while((line=reader2.readLine())!=null){
                        if(Counter==99)break;
                        data[Counter]=line.split(",");
                        Counter=Counter+1;
                }
        }catch(Exception x){
                System.out.println("Raise Exception");
                System.exit(1);
        }
        try{
                reader2.close();
        }catch(Exception x){
                System.exit(1);
        }
        Counter=0;
        for(int j=0;j<data.length;j++){
                TEMP=data[j][1].substring(2,6);
                temp=Integer.parseInt(TEMP);
                sec=temp%100;
                min=(temp%10000)-sec;
                min=min/100;
                temp2=(min*60);
                temp2+=sec;//temp2 time in sec
                if(Counter==0){
                        latitude[Counter]=Double.parseDouble(data[j][2]);

longtitude[Counter]=Double.parseDouble(data[j][4]);
                        time[Counter]=temp2;
                        Counter+=1;
                }
                else if(Counter<5 && Counter>0){
                        if(temp2-time[Counter-1]>18){

latitude[Counter]=Double.parseDouble(data[j][2]);

longtitude[Counter]=Double.parseDouble(data[j][4]);
                                time[Counter]=temp2;
                                Counter=Counter+1;
                        }
                }
                else break;
        }
        String cmd=GPS;
        long a,b;
        int aa,bb;
        for(int j=0;j<5;j++){
                a=(long)(longtitude[j]);
```

```
		b=(long)(latitude[j]);
		aa=(int)((longtitude[j]-a)*60);
		bb=(int)((latitude[j]-b)*60);
		cmd=cmd+"T="+a+aa+b+bb;
	}
	cmd=cmd+"\r\n";
	System.out.println(cmd);
	Modem modem2=new Modem();
	modem2.setSpeed(80000);
	modem2.setTimeout(2000);
	modem2.open("ithaki");
	for(;;){
		try{
			k=modem2.read();
			if(k==-1)break;
			System.out.print((char)k);
		}catch(Exception x){
			System.out.println("Raise Exception");
			System.exit(1);
		}
	}
	for(;;){
		try{
			modem2.write(cmd.getBytes());
		}catch(Exception x){
			System.out.println("Raise Exception");
			System.exit(1);
		}
		try{
			k=modem2.read();
			if(k==-1)break;
			System.out.print((char)k);
			if(k==0xFF){
				for(;;){
					out.write(k);
					k=modem2.read();
					if(k==0xFF){
						out.write(k);
						k=modem2.read();
						if(k==0xD9){
							out.write(k);
							counter=true;
						}
					}
					if(counter)break;
				}
			}
		}catch(Exception x){
			System.exit(1);
		}
		if(counter)break;
```

```
		}
		modem2.close();
		try{
			out.close();
		}catch(Exception x){
			System.exit(1);
		}

}

public void acknack() {
		String ACK = new String();
		String NACK = new String;
		ACK = "Q8448\r";
		NACK = "R4166\r";
		Modem modem = new Modem();
		modem.setSpeed(80000);
		modem.setTimeout(2000);
		modem.open("ithaki");
		int a;
		for (;;) {
			a = modem.read();
			if (a == -1)
				break;
			System.out.print((char) a);
		}
		int k ;
		PrintWriter response=null;
		PrintWriter retrans=null;
		try{
			response=new PrintWriter("ACK.txt","UTF-8");
			retrans=new PrintWriter("ReACK.txt", "UTF-8");
		}catch(Exception x){
			System.out.println("Raise Exception");
			System.exit(1);
		}
		long t1,t2;
		long total=0;
		String currentPacket="";
		String code,result;
		int xorResult;
		char xor;
		float responseTime;
		long correct=0;
		long wrong=0;
		long retransCount=0;
		long tStart;
		tStart=System.currentTimeMillis();
		t1=System.currentTimeMillis();
		try{
			modem.write(ACK.getBytes());
```

```
	}catch(Exception x){
		System.exit(1);
	}
	while((System.currentTimeMillis()-tStart)<240000){
		for(;;){
			try{
				k=modem.read();
				if(k==-1)break;
				currentPacket+=(char)k;
				if(currentPacket.endsWith("PSTOP")){
					code=currentPacket.substring(31,47);
					result=currentPacket.substring(49,52);
					xorResult=Integer.parseInt(result);
					xor=code.charAt(0);
					for(int i=1;i<16;i++){

xor=(char)(xor^(code.charAt(i)));
					}
					if((int)xor==xorResult){

retrans.write(Long.toString(retransCount));

retrans.write(System.lineSeparator());
						if(retransCount > 0) {
							retransCount = 0;
						}
						t2=System.currentTimeMillis();

responseTime=(float)((float)(t2-t1)/1000);

response.write(Float.toString(responseTime));

response.write(System.lineSeparator());
					correct++;
					t1=System.currentTimeMillis();
					try{
						modem.write(ACK.getBytes());
					}catch(Exception x){
						System.exit(1);
					}
					}else{
						wrong++;
						retransCount++;
						try{

modem.write(NACK.getBytes());
						}catch(Exception x){
							System.exit(1);
						}
					}
					break;
```

```
                                        }
                                } catch(Exception x){
                                        System.exit(1);
                                }
                        }
                        total++;
                        currentPacket="";
                }
                System.out.println("Total Packets: " + total);
                System.out.println("Corrent Packets: " + correct);
                System.out.println("Wrong Packets: " + wrong);
                retrans.close();
                response.close();
        }
}
```