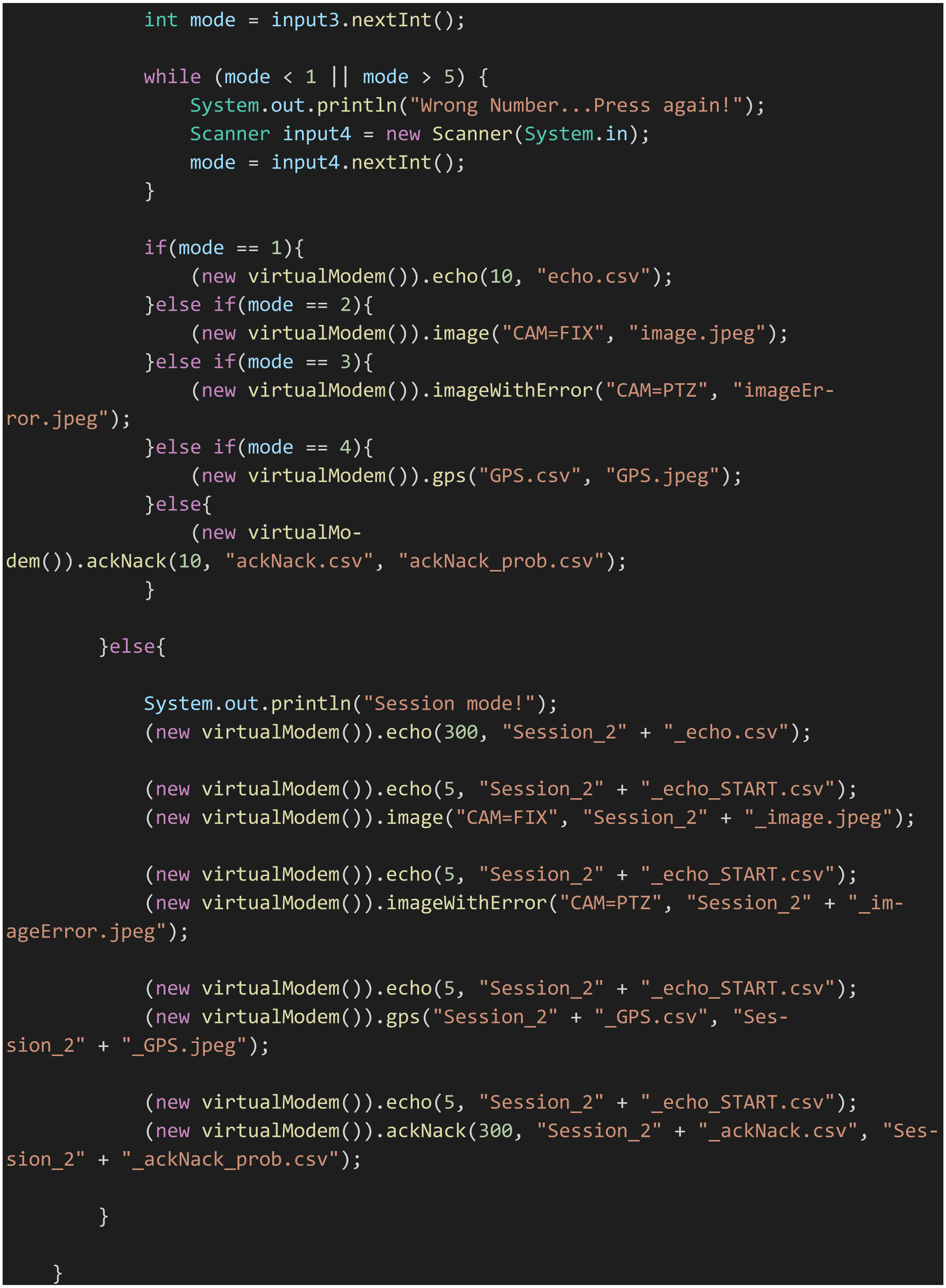
Δίκτυα Υπολογιστών 1

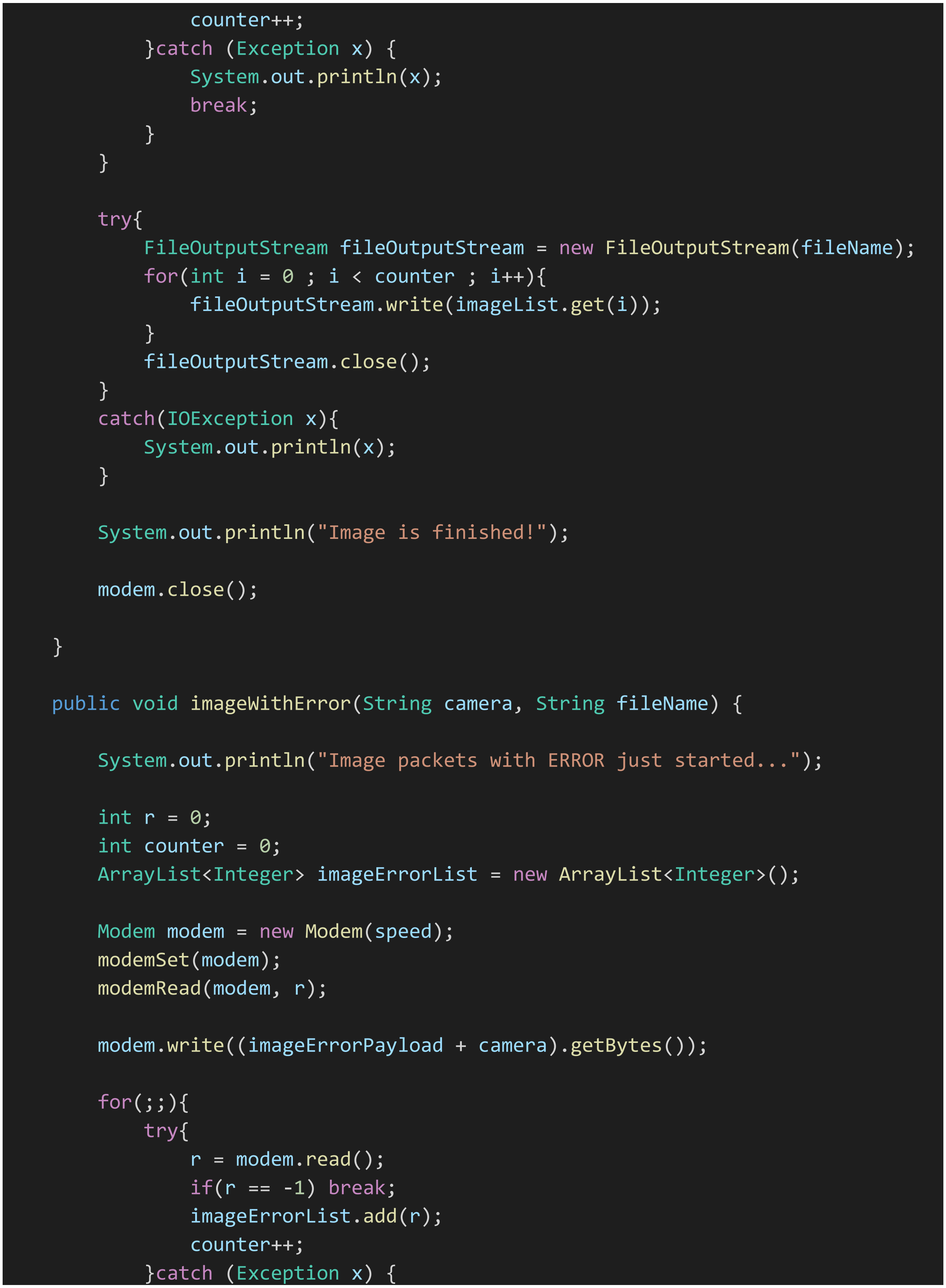
Σωκράτης Κοσέογλου 8837 sokrkose@ece.auth.gr

```java
// Author: Sokratis Koseoglou

import java.util.ArrayList;
import java.util.Scanner;
import java.io.*;
import java.io.IOException;
import ithakimodem.Modem;

public class virtualModem{

    String echoPayload = "E8137\r";
    String imagePayload = "M0739\r";
    String imageErrorPayload = "G3534\r";
    String gpsPayload = "P3154";
    String ackResultPayload = "Q3111\r";
    String nackResultPayload = "R4942\r";

    int speed = 80000;
    int timeout = 1000;
    String callNumber = "atd2310ithaki\r";

    public static void main(String[] param) throws IOException{

        System.out.println("Do you want to run Standalone Packets or Session ? [1, 2]");
        System.out.println("1. Standalone Packets");
        System.out.println("2. Session");
        Scanner input1 = new Scanner(System.in);
        int option = input1.nextInt();

        while (option != 1 && option != 2) {
            System.out.println("Wrong Number...Press again!");
            Scanner input2 = new Scanner(System.in);
            option = input2.nextInt();
        }

        if (option == 1){

            System.out.println("Enter mode of operation: [1, 2, 3, 4, 5]");
            System.out.println("1. Echo Packets");
            System.out.println("2. Image Packets");
            System.out.println("3. Image with ERROR Packets");
            System.out.println("4. GPS Packets");
            System.out.println("5. ACK-NACK Packets");
            Scanner input3 = new Scanner(System.in);
```

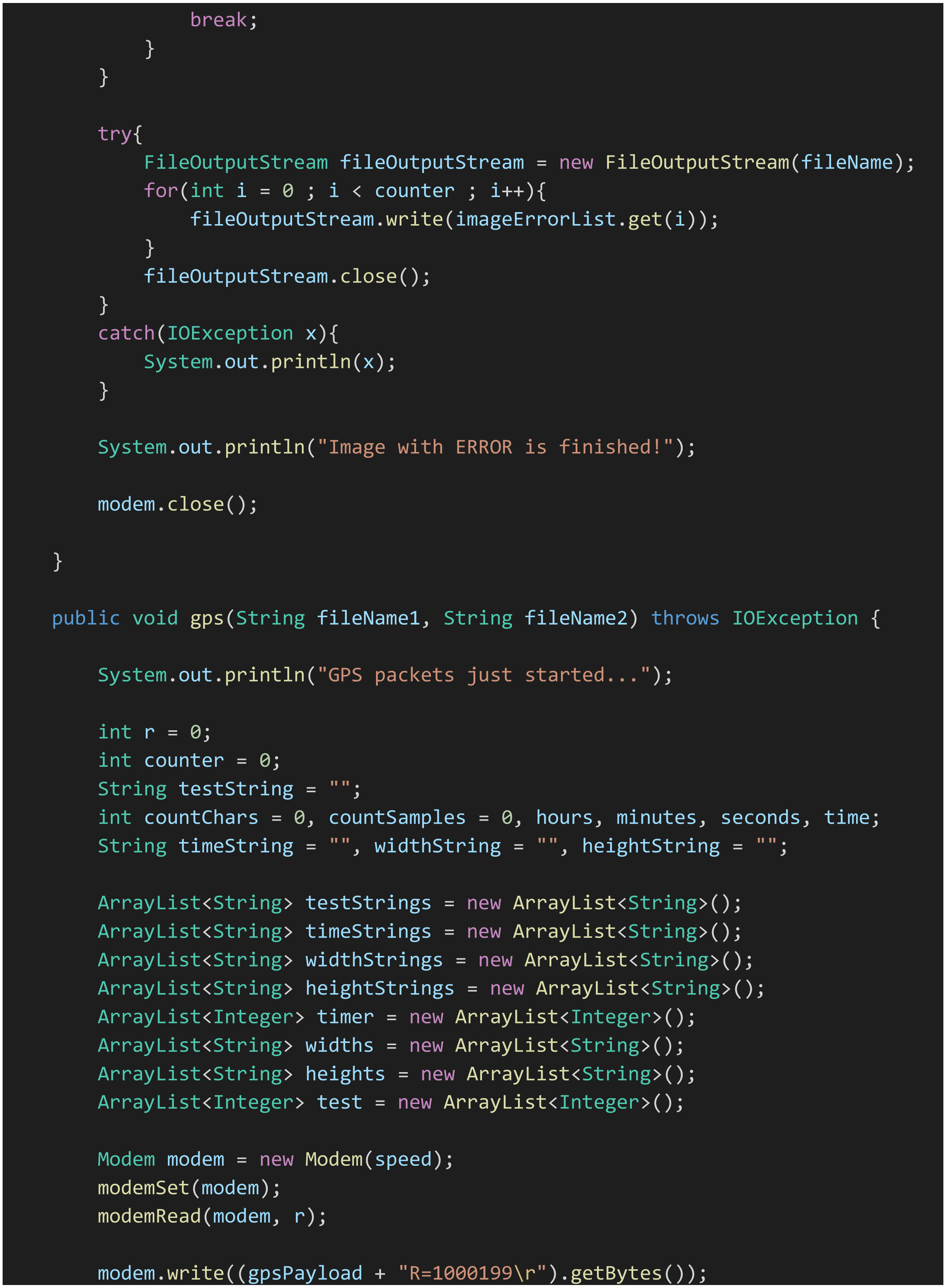


```
            int mode = input3.nextInt();

            while (mode < 1 || mode > 5) {
                System.out.println("Wrong Number...Press again!");
                Scanner input4 = new Scanner(System.in);
                mode = input4.nextInt();
            }

            if(mode == 1){
                (new virtualModem()).echo(10, "echo.csv");
            }else if(mode == 2){
                (new virtualModem()).image("CAM=FIX", "image.jpeg");
            }else if(mode == 3){
                (new virtualModem()).imageWithError("CAM=PTZ", "imageError.jpeg");
            }else if(mode == 4){
                (new virtualModem()).gps("GPS.csv", "GPS.jpeg");
            }else{
                (new virtualModem()).ackNack(10, "ackNack.csv", "ackNack_prob.csv");
            }

        }else{

            System.out.println("Session mode!");
            (new virtualModem()).echo(300, "Session_2" + "_echo.csv");

            (new virtualModem()).echo(5, "Session_2" + "_echo_START.csv");
            (new virtualModem()).image("CAM=FIX", "Session_2" + "_image.jpeg");

            (new virtualModem()).echo(5, "Session_2" + "_echo_START.csv");
            (new virtualModem()).imageWithError("CAM=PTZ", "Session_2" + "_imageError.jpeg");

            (new virtualModem()).echo(5, "Session_2" + "_echo_START.csv");
            (new virtualModem()).gps("Session_2" + "_GPS.csv", "Session_2" + "_GPS.jpeg");

            (new virtualModem()).echo(5, "Session_2" + "_echo_START.csv");
            (new virtualModem()).ackNack(300, "Session_2" + "_ackNack.csv", "Session_2" + "_ackNack_prob.csv");

        }

    }
```

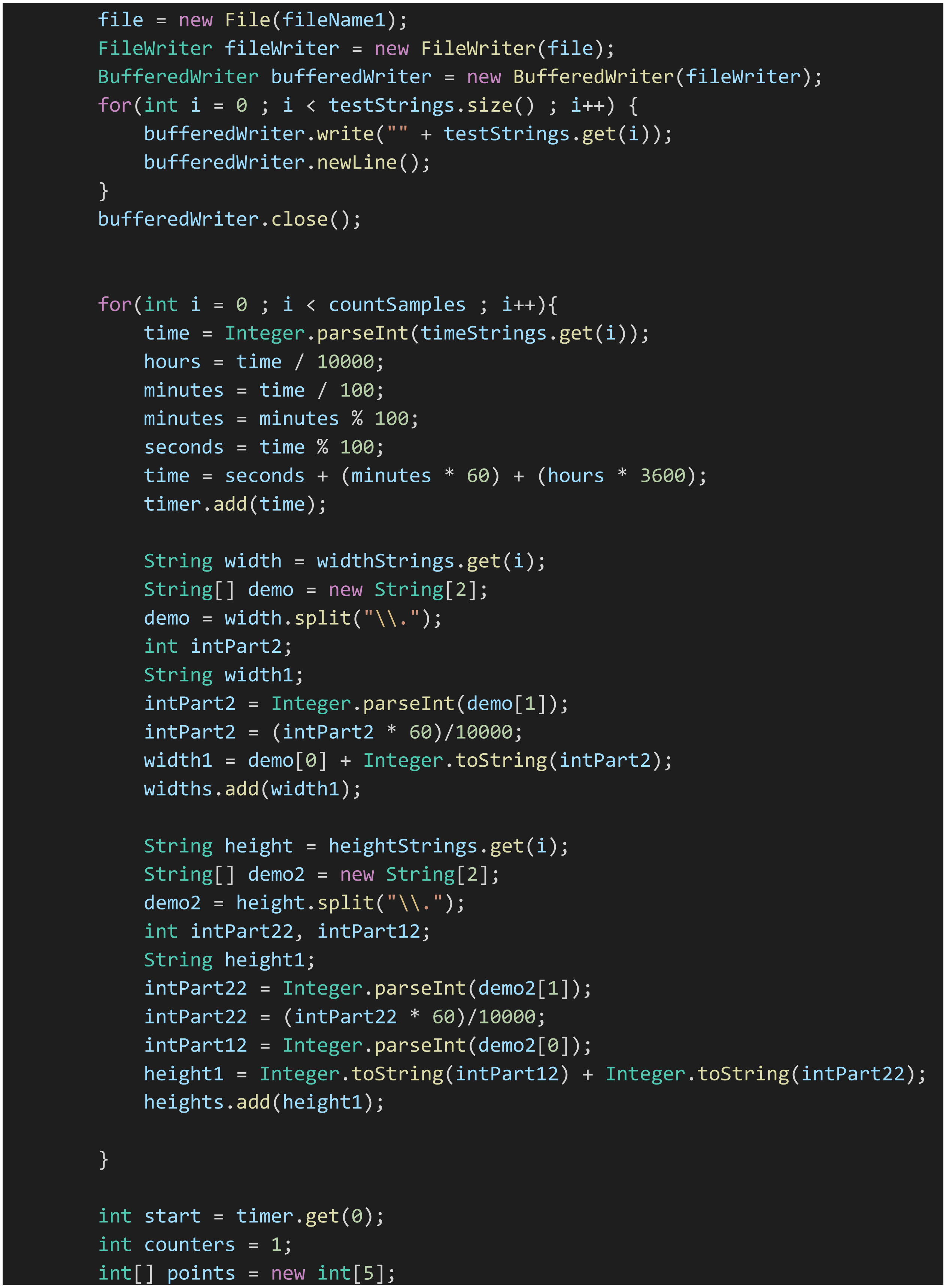
```java
    public void echo(long echoPacketsSeconds, String fileName) throws IOException {

        System.out.println("Echo packets just started...");

        long startLoop = System.currentTimeMillis();
        long endLoop = 0;
        long deltaLoop = 0;
        long s, e, dt = 0;
        int counterP = 0, counterLoop = 0;
        int r = 0;

        File file;
        file = new File(fileName);

        ArrayList<Long> timeSamples = new ArrayList<Long>();

        Modem modem = new Modem(speed);
        modemSet(modem);
        modemRead(modem, r);

        while((deltaLoop/1000) < echoPacketsSeconds) {
            modem.write(echoPayload.getBytes());

            s = System.currentTimeMillis();
            e = 0;
            dt = 0;

            for(;;){
                try{
                    r = modem.read();
                    if((char )r == 'P'){
                        counterP++;
                    }
                    if(counterP == 3){
                        e = System.currentTimeMillis();
                        dt = e - s;
                        counterP = 0;
                    }
                    if(r == -1) break;
                    System.out.print((char )r);
                }catch(Exception x){
                    System.out.println(x);
                    break;
                }
            }
            System.out.println("\n" + dt);

            timeSamples.add(dt);

            endLoop = System.currentTimeMillis();
            deltaLoop = endLoop - startLoop;

            counterLoop++;
        }

        FileWriter fileWriter = new FileWriter(file);
        BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
        for(int i = 0 ; i < counterLoop ; i++) {
            bufferedWriter.write("" + timeSamples.get(i));
            bufferedWriter.newLine();
        }

        System.out.println("Echo packets are finished!");

        bufferedWriter.close();
        modem.close();

    }

    public void image(String camera, String fileName){

        System.out.println("Image packets just started...");

        int r = 0;
        int counter = 0;
        ArrayList<Integer> imageList = new ArrayList<Integer>();

        Modem modem = new Modem(speed);
        modemSet(modem);
        modemRead(modem, r);

        modem.write((imagePayload + camera).getBytes());

        for(;;){
            try{
                r = modem.read();
                if(r == -1) break;
                imageList.add(r);
```



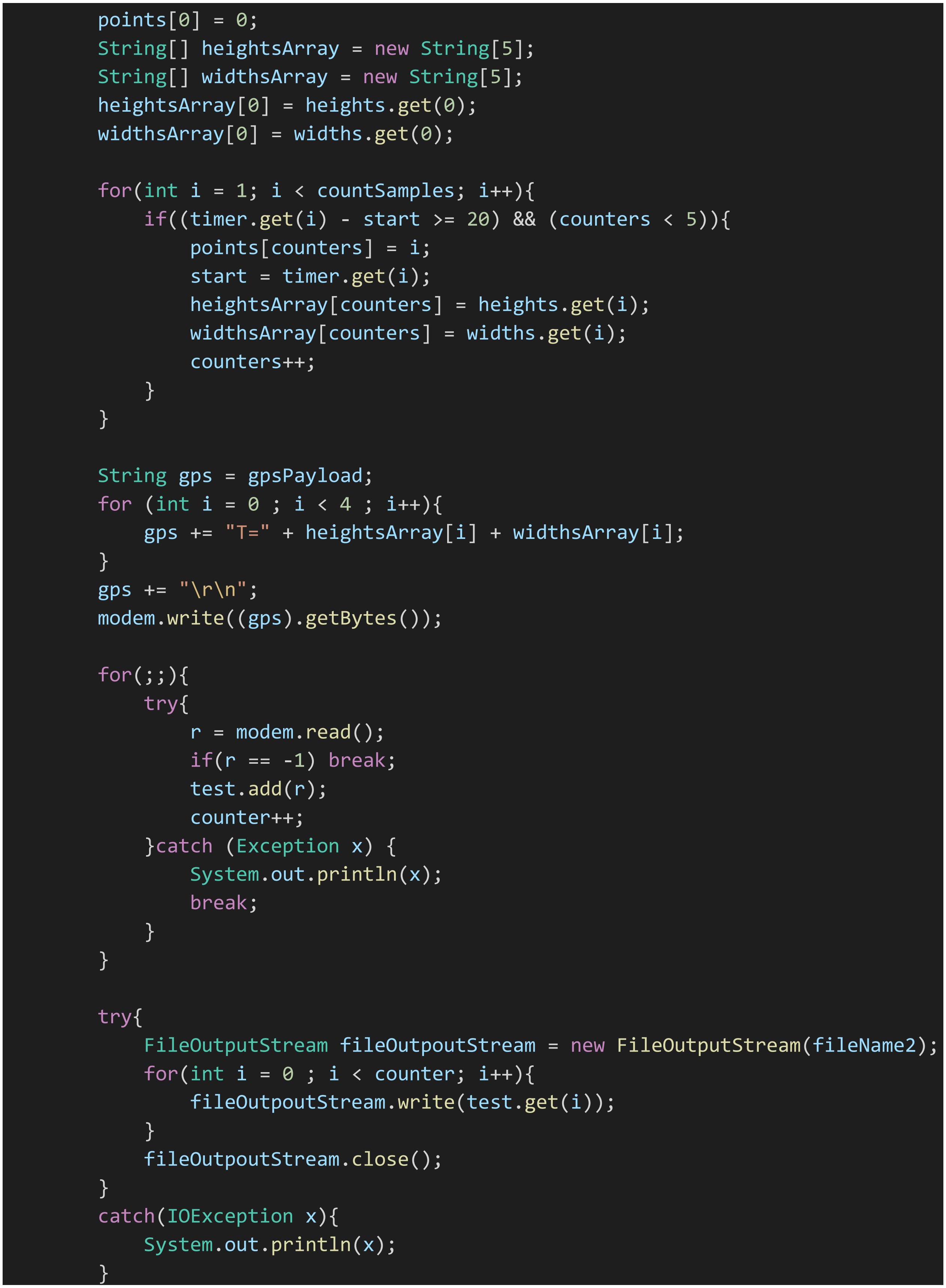
```
                counter++;
            }catch (Exception x) {
                System.out.println(x);
                break;
            }
        }

        try{
            FileOutputStream fileOutputStream = new FileOutputStream(fileName);
            for(int i = 0 ; i < counter ; i++){
                fileOutputStream.write(imageList.get(i));
            }
            fileOutputStream.close();
        }
        catch(IOException x){
            System.out.println(x);
        }

        System.out.println("Image is finished!");

        modem.close();

    }

    public void imageWithError(String camera, String fileName) {

        System.out.println("Image packets with ERROR just started...");

        int r = 0;
        int counter = 0;
        ArrayList<Integer> imageErrorList = new ArrayList<Integer>();

        Modem modem = new Modem(speed);
        modemSet(modem);
        modemRead(modem, r);

        modem.write((imageErrorPayload + camera).getBytes());

        for(;;){
            try{
                r = modem.read();
                if(r == -1) break;
                imageErrorList.add(r);
                counter++;
            }catch (Exception x) {
```



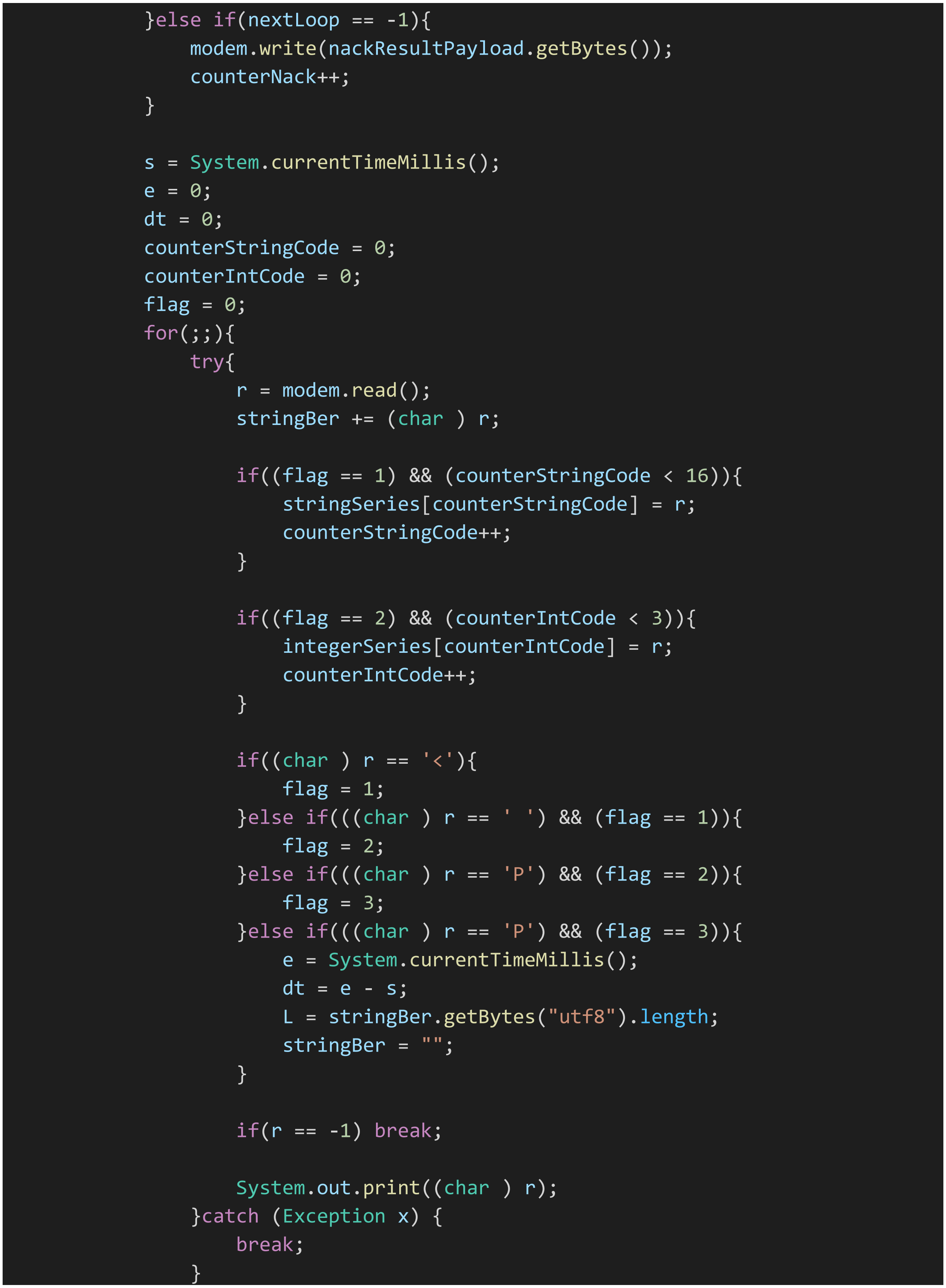
```
                break;
            }
        }

        try{
            FileOutputStream fileOutputStream = new FileOutputStream(fileName);
            for(int i = 0 ; i < counter ; i++){
                fileOutputStream.write(imageErrorList.get(i));
            }
            fileOutputStream.close();
        }
        catch(IOException x){
            System.out.println(x);
        }

        System.out.println("Image with ERROR is finished!");

        modem.close();

    }

    public void gps(String fileName1, String fileName2) throws IOException {

        System.out.println("GPS packets just started...");

        int r = 0;
        int counter = 0;
        String testString = "";
        int countChars = 0, countSamples = 0, hours, minutes, seconds, time;
        String timeString = "", widthString = "", heightString = "";

        ArrayList<String> testStrings = new ArrayList<String>();
        ArrayList<String> timeStrings = new ArrayList<String>();
        ArrayList<String> widthStrings = new ArrayList<String>();
        ArrayList<String> heightStrings = new ArrayList<String>();
        ArrayList<Integer> timer = new ArrayList<Integer>();
        ArrayList<String> widths = new ArrayList<String>();
        ArrayList<String> heights = new ArrayList<String>();
        ArrayList<Integer> test = new ArrayList<Integer>();

        Modem modem = new Modem(speed);
        modemSet(modem);
        modemRead(modem, r);

        modem.write((gpsPayload + "R=1000199\r").getBytes());
        for(;;){
            try{
                r = modem.read();
                testString += (char ) r;

                if((testString.indexOf("GPGGA") > 0) && (testString.indexOf("\r\n") > 0)){
                    testStrings.add(testString);
                    timeStrings.add(timeString);
                    widthStrings.add(widthString);
                    heightStrings.add(heightString);
                    testString = "";
                    timeString = "";
                    widthString = "";
                    heightString = "";
                    countChars = 0;
                    countSamples++;

                }else if((testString.indexOf("GPGGA") < 0) && (testString.indexOf("\r\n") > 0)){
                    testString = "";

                }else if((testString.indexOf("GPGGA") > 0) && (testString.indexOf("\r\n") < 0)){
                    countChars++;
                    if(countChars > 2 && countChars < 9){
                        timeString += (char ) r;
                    }
                    if(countChars > 13 && countChars < 23){
                        widthString += (char ) r;
                    }
                    if(countChars > 25 && countChars < 36){
                        heightString += (char ) r;
                    }
                }

                if(r == -1) break;

            }catch (Exception x) {
                System.out.println(x);
                break;
            }
        }

        File file;
```



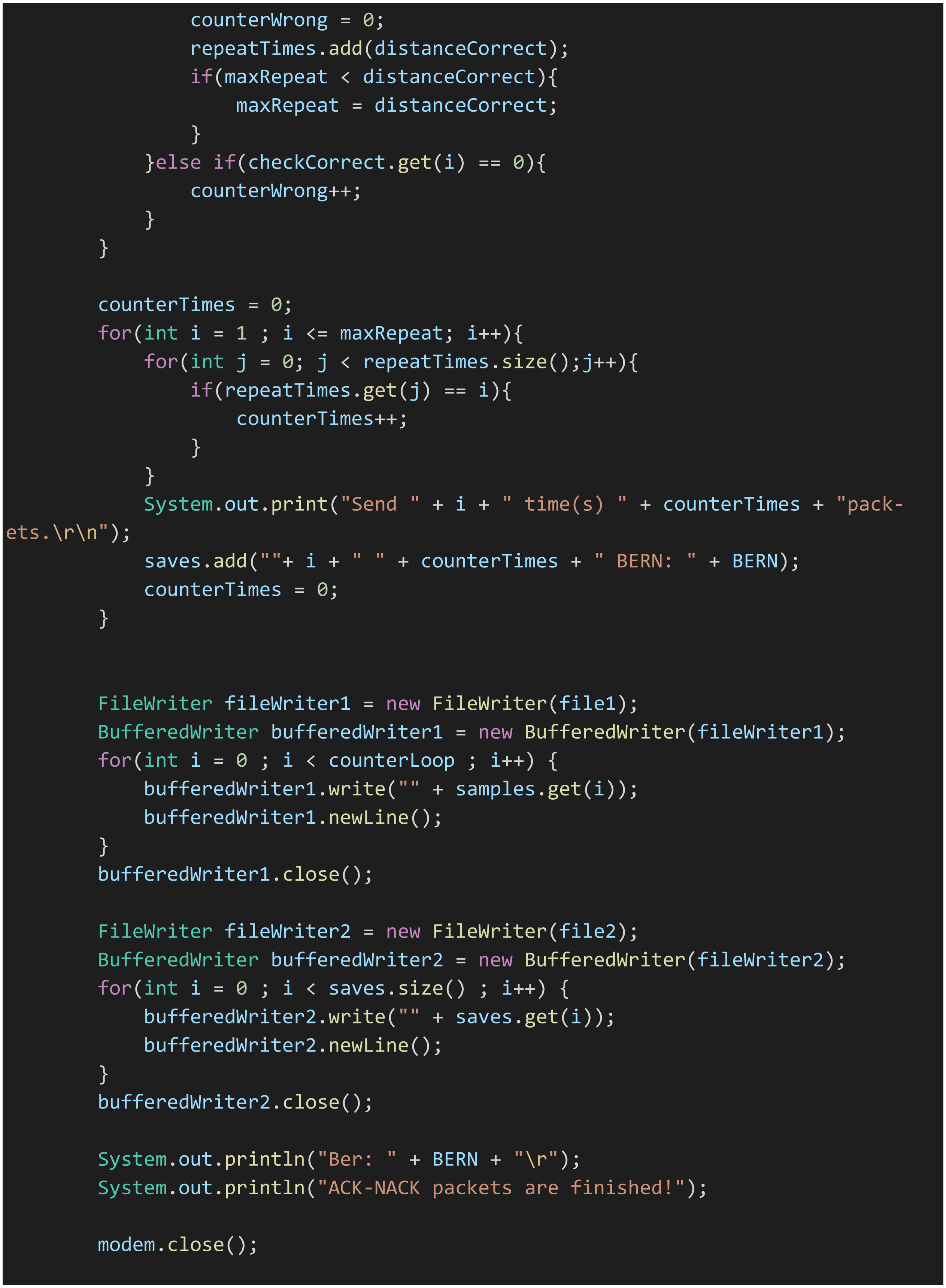
```
        file = new File(fileName1);
        FileWriter fileWriter = new FileWriter(file);
        BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
        for(int i = 0 ; i < testStrings.size() ; i++) {
            bufferedWriter.write("" + testStrings.get(i));
            bufferedWriter.newLine();
        }
        bufferedWriter.close();


        for(int i = 0 ; i < countSamples ; i++){
            time = Integer.parseInt(timeStrings.get(i));
            hours = time / 10000;
            minutes = time / 100;
            minutes = minutes % 100;
            seconds = time % 100;
            time = seconds + (minutes * 60) + (hours * 3600);
            timer.add(time);

            String width = widthStrings.get(i);
            String[] demo = new String[2];
            demo = width.split("\\.");
            int intPart2;
            String width1;
            intPart2 = Integer.parseInt(demo[1]);
            intPart2 = (intPart2 * 60)/10000;
            width1 = demo[0] + Integer.toString(intPart2);
            widths.add(width1);

            String height = heightStrings.get(i);
            String[] demo2 = new String[2];
            demo2 = height.split("\\.");
            int intPart22, intPart12;
            String height1;
            intPart22 = Integer.parseInt(demo2[1]);
            intPart22 = (intPart22 * 60)/10000;
            intPart12 = Integer.parseInt(demo2[0]);
            height1 = Integer.toString(intPart12) + Integer.toString(intPart22);
            heights.add(height1);

        }

        int start = timer.get(0);
        int counters = 1;
        int[] points = new int[5];
```



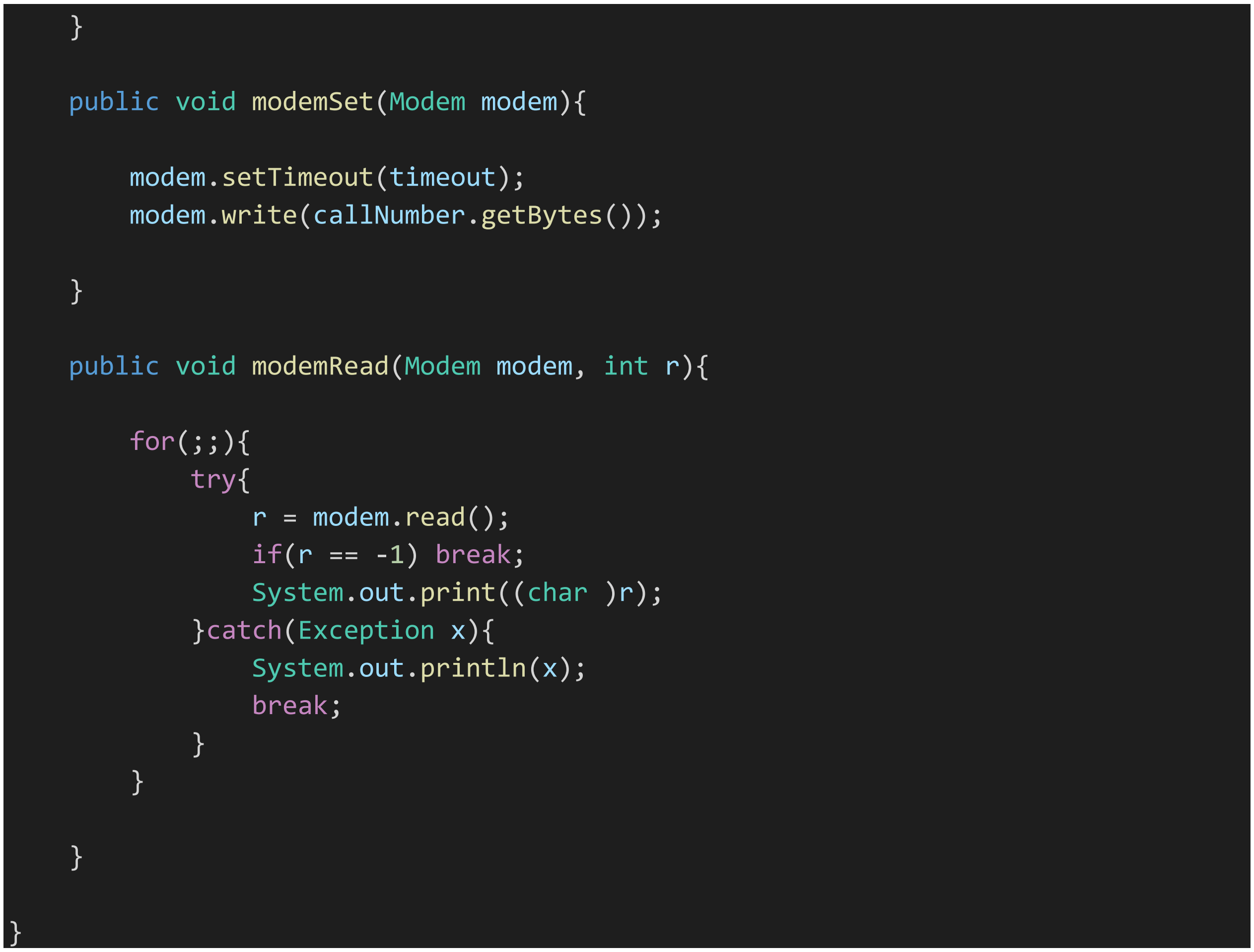
```
        points[0] = 0;
        String[] heightsArray = new String[5];
        String[] widthsArray = new String[5];
        heightsArray[0] = heights.get(0);
        widthsArray[0] = widths.get(0);

        for(int i = 1; i < countSamples; i++){
            if((timer.get(i) - start >= 20) && (counters < 5)){
                points[counters] = i;
                start = timer.get(i);
                heightsArray[counters] = heights.get(i);
                widthsArray[counters] = widths.get(i);
                counters++;
            }
        }

        String gps = gpsPayload;
        for (int i = 0 ; i < 4 ; i++){
            gps += "T=" + heightsArray[i] + widthsArray[i];
        }
        gps += "\r\n";
        modem.write((gps).getBytes());

        for(;;){
            try{
                r = modem.read();
                if(r == -1) break;
                test.add(r);
                counter++;
            }catch (Exception x) {
                System.out.println(x);
                break;
            }
        }

        try{
            FileOutputStream fileOutpoutStream = new FileOutputStream(fileName2);
            for(int i = 0 ; i < counter; i++){
                fileOutpoutStream.write(test.get(i));
            }
            fileOutpoutStream.close();
        }
        catch(IOException x){
            System.out.println(x);
        }
        System.out.println("GPS is finished!");

        modem.close();
    }

    public void ackNack(long ackNackPacketsSeconds, String fileName1, String fileName2) throws IOException {

        System.out.println("ACK-NACK packets just started...");

        int r = 0;
        int[] stringSeries = new int[16];
        int[] integerSeries = new int[3];
        int nextLoop = 1, flag = 0, counterNack = 0, counterStringCode, counterIntCode, counterLoop = 0;
        int distanceCorrect, intCodeNumericValue = 0, counterWrong, counterTimes, temp;
        long startLoop, endLoop, deltaLoop, s, e, dt;
        float BERN, probability, L = 0;
        String stringBer = "";

        ArrayList<Integer> checkCorrect = new ArrayList<Integer>();
        ArrayList<Long> samples = new ArrayList<Long>();
        ArrayList<Integer> stringSeriesList = new ArrayList<Integer>();
        ArrayList<Integer> repeatTimes = new ArrayList<Integer>();
        ArrayList<String> saves = new ArrayList<String>();

        File file1, file2;
        file1 = new File(fileName1);
        file2 = new File(fileName2);

        Modem modem = new Modem(speed);

        modemSet(modem);
        modemRead(modem, r);

        startLoop = System.currentTimeMillis();
        endLoop = 0;
        deltaLoop = 0;

        while(((deltaLoop/1000) < ackNackPacketsSeconds) || (nextLoop == -1)){

            if(nextLoop == 1){
                modem.write(ackResultPayload.getBytes());
```



```
            }else if(nextLoop == -1){
                modem.write(nackResultPayload.getBytes());
                counterNack++;
            }

            s = System.currentTimeMillis();
            e = 0;
            dt = 0;
            counterStringCode = 0;
            counterIntCode = 0;
            flag = 0;
            for(;;){
                try{
                    r = modem.read();
                    stringBer += (char ) r;

                    if((flag == 1) && (counterStringCode < 16)){
                        stringSeries[counterStringCode] = r;
                        counterStringCode++;
                    }

                    if((flag == 2) && (counterIntCode < 3)){
                        integerSeries[counterIntCode] = r;
                        counterIntCode++;
                    }

                    if((char ) r == '<'){
                        flag = 1;
                    }else if(((char ) r == ' ') && (flag == 1)){
                        flag = 2;
                    }else if(((char ) r == 'P') && (flag == 2)){
                        flag = 3;
                    }else if(((char ) r == 'P') && (flag == 3)){
                        e = System.currentTimeMillis();
                        dt = e - s;
                        L = stringBer.getBytes("utf8").length;
                        stringBer = "";
                    }

                    if(r == -1) break;

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

            for(int i = 0; i < 16; i++){
                stringSeriesList.add(stringSeries[i]);
            }

            temp = 0;
            for(int i = 0; i < 16; i++){
                temp = temp^stringSeries[i];
            }

            intCodeNumericValue = 0;
            for(int i = 0; i < 3; i++){
                intCodeNumericValue += ((int)Math.pow(10,2-i))*Character.getNumericValue(integerSeries[i]);
            }

            if(temp == intCodeNumericValue){
                nextLoop = 1;
                checkCorrect.add(1);
            }else{
                nextLoop = -1;
                checkCorrect.add(0);
            }

            System.out.print("\r\n");
            System.out.print(dt + "\r\n");
            samples.add(dt);
            endLoop = System.currentTimeMillis();
            deltaLoop = endLoop - startLoop;

            counterLoop++;
        }

        probability = (float)(counterLoop - counterNack)/counterLoop;
        BERN = (float)(1.0 - Math.pow(probability,1.0/L));
        System.out.print("Ber: " + BERN);
        System.out.print("\r\n");

        distanceCorrect = 0;
        counterWrong = 0;
        int maxRepeat = 1;
        for(int i = 0 ; i < counterLoop; i++){
            if(checkCorrect.get(i) == 1){
                distanceCorrect = 1 + counterWrong;
```



```
                counterWrong = 0;
                repeatTimes.add(distanceCorrect);
                if(maxRepeat < distanceCorrect){
                    maxRepeat = distanceCorrect;
                }
            }else if(checkCorrect.get(i) == 0){
                counterWrong++;
            }
        }

        counterTimes = 0;
        for(int i = 1 ; i <= maxRepeat; i++){
            for(int j = 0; j < repeatTimes.size();j++){
                if(repeatTimes.get(j) == i){
                    counterTimes++;
                }
            }
            System.out.print("Send " + i + " time(s) " + counterTimes + "packets.\r\n");
            saves.add(""+ i + " " + counterTimes + " BERN: " + BERN);
            counterTimes = 0;
        }


        FileWriter fileWriter1 = new FileWriter(file1);
        BufferedWriter bufferedWriter1 = new BufferedWriter(fileWriter1);
        for(int i = 0 ; i < counterLoop ; i++) {
            bufferedWriter1.write("" + samples.get(i));
            bufferedWriter1.newLine();
        }
        bufferedWriter1.close();

        FileWriter fileWriter2 = new FileWriter(file2);
        BufferedWriter bufferedWriter2 = new BufferedWriter(fileWriter2);
        for(int i = 0 ; i < saves.size() ; i++) {
            bufferedWriter2.write("" + saves.get(i));
            bufferedWriter2.newLine();
        }
        bufferedWriter2.close();

        System.out.println("Ber: " + BERN + "\r");
        System.out.println("ACK-NACK packets are finished!");

        modem.close();
```



```java
    }

    public void modemSet(Modem modem){

        modem.setTimeout(timeout);
        modem.write(callNumber.getBytes());

    }

    public void modemRead(Modem modem, int r){

        for(;;){
            try{
                r = modem.read();
                if(r == -1) break;
                System.out.print((char )r);
            }catch(Exception x){
                System.out.println(x);
                break;
            }
        }

    }

}
```