```java
package tellscopeV4;

//import libraries
import java.net.*;

public class TellCalcThread implements Runnable {


    //default constructor
    public TellCalcThread(Socket socket)
    {
        this.s = socket;

    }

    //socket for incoming data from the client
    private Socket s;

    //array to store incoming calculation values
    double[] calcValues = new double[3];

    //counter for input array - used as index to the array
    int i = 0;


    //private attributes to pass to TellsCalculation object
    private double focalRatio;          //store focal ratio
    private double lensDiameter;        //store lens diameter
    private double eyePieceFocalLength; //store eyepiece focal length

    private Scanner in;

    private boolean type;

    /*
    //variables to store calculated results
    private static double focalLength;
    private static double tubeLength;
    private static double distToSecond;
    private static double secondarySizeMinor;
    private static double secondarySizeMajor;
    private static double minMagnitude;
    private static double minResolution;
    private static double maxVisibleMagnification;
    private static double minVisibleMagnification;
    private static double eyePieceMagnification;

    */



    public void run() {

        String client = s.getInetAddress().toString();
        TellServerGui04.consoleView.append("\nConnected to " + client);
        TellServerGui04.consoleView.append("\nRunning Calculations");
        System.out.println("Connected to " + client);

        //try creating a new scanner
        try
        {
            //create new scanner object and set it to get input stream from socket
            in = new Scanner(s.getInputStream());



            // loop to test if the socket input reads "calculate"
            // if it does not - add values to array for processing later
            // if it does - sort values, perform calculations and store to TellServer calcResults array
            // ready for sending to client
            while(true)
            {
                //store next line in "input" string
                String input = in.nextLine();
                //check for equality with string "calculate"
                if(input.equalsIgnoreCase("calculate"))
                {
                    //add input values to correct attributes for calculations
                    focalRatio = calcValues[0];
                    lensDiameter = calcValues[1];
                    eyePieceFocalLength = calcValues[2];

                    /*
                    /* write inputs to input text fields */

                    TellServerGui04.setInputs(String.format("%.2f", lensDiameter),String.format
                            ("%.2f", focalRatio), String.format("%.2f", eyePieceFocalLength));



                    /* moved below for test
                    //create new TellsCalculations object and pass user input attributes
                    TeleScopeReflect testCalcs = new TeleScopeReflect(focalRatio, lensDiameter, eyePieceFocalLength);
                    */
```

```java

                            /* check the telescope type (reflect == true, refract == false) */
                            if(type == true)
                            {
                                //create new TellsCalculations object and pass user input attributes
                                TeleScopeReflect reflecting = new TeleScopeReflect(focalRatio, lensDiameter, eyePieceFocalLength);

                                //call calculation methods to get values
                                //convert to string with 2 decimal places
                                //store in TellServer.calcResults array
                                TellServerGui04.calcResults[0] = String.format("%.2f", reflecting.calcFocalLength());
                                TellServerGui04.calcResults[1] = String.format("%.2f", reflecting.calcTubeLength());
                                TellServerGui04.calcResults[2] = String.format("%.2f", reflecting.calcDistToSecond());
                                TellServerGui04.calcResults[3] = String.format("%.2f", reflecting.calcSecondarySizeMinor());
                                TellServerGui04.calcResults[4] = String.format("%.2f", reflecting.calcSecondarySizeMajor());
                                TellServerGui04.calcResults[5] = String.format("%.2f", reflecting.calcMinMagnitude());
                                TellServerGui04.calcResults[6] = String.format("%.2f", reflecting.calcMinResolution());
                                TellServerGui04.calcResults[7] = String.format("%.2f", reflecting.calcMaxVisibleMag());
                                TellServerGui04.calcResults[8] = String.format("%.2f", reflecting.calcMinVisibleMag());
                                TellServerGui04.calcResults[9] = String.format("%.2f", reflecting.calcEyepieceMagnification());


                                //once array is populated - add string "result" to inform client to display the results
                                TellServerGui04.calcResults[10] = "result";

                                /* place results in server results area */
                                TellServerGui04.setResults();

                                //once calculations are done and stored - create a new thread to pass results to client
                                Thread sendResults = new Thread(new SendResultsThread(s));
                                sendResults.start();

                                i = 0;
                                //DO I NEED A BREAK HERE????????
                                continue;
                            }

                                    else if(type == false)
                                    {
                                        //create new TellsCalculations object and pass user input attributes
                                        TeleScopeRefract refracting = new TeleScopeRefract(focalRatio, lensDiameter, eyePieceFocalLength);

                                        //call calculation methods to get values
                                        //convert to string with 2 decimal places
                                        //store in TellServer.calcResults array
                                        TellServerGui04.calcResults[0] = String.format("%.2f", refracting.calcFocalLength());
                                        TellServerGui04.calcResults[1] = String.format("%.2f", refracting.calcTubeLength());
                                        TellServerGui04.calcResults[2] = "Not Required";
                                        TellServerGui04.calcResults[3] = "Not Required";
                                        TellServerGui04.calcResults[4] = "Not Required";
                                        TellServerGui04.calcResults[5] = String.format("%.2f", refracting.calcMinMagnitude());
                                        TellServerGui04.calcResults[6] = String.format("%.2f", refracting.calcMinResolution());
                                        TellServerGui04.calcResults[7] = String.format("%.2f", refracting.calcMaxVisibleMag());
                                        TellServerGui04.calcResults[8] = String.format("%.2f", refracting.calcMinVisibleMag());
                                        TellServerGui04.calcResults[9] = String.format("%.2f", refracting.calcEyepieceMagnification());


                                        //once array is populated - add string "result" to inform client to display the results
                                        TellServerGui04.calcResults[10] = "result";

                                        //TellServer.printResults();
                                        //TellServer.sendResultsToClient();

                                        /* place results in server results area */
                                        TellServerGui04.setResults();

                                        //once calculations are done and stored - create a new thread to pass results to client
                                        Thread sendResults = new Thread(new SendResultsThread(s));
                                        sendResults.start();

                                        i = 0;
                                        //DO I NEED A BREAK HERE????????
                                        continue;
                                    }

                }


                else if(input.equalsIgnoreCase("reflect"))
                {
                    type = true;
                }

                else if(input.equalsIgnoreCase("refract"))
                {
                    type = false;
                }



                    //if input != "calculate"
                    else
                    {


                        //try parse the input to a double
                        try{

                            //store in calcValues array for sorting later
                            calcValues[i] = Double.parseDouble(input);
                            //increment counter for array index
                            i++;
                            }
                            //catch exceptions
                            catch(NumberFormatException nfe)
                            {
                                //if exception caught - print "Failed" and exception to console
                                System.out.println("Failed: " + nfe.toString());
                                //exit
                                System.exit(1);
                            }//end nfe exception catch

                    }//end else

            }//end while loop

        }//end top try
        catch(Exception e)
        {
            //if exception caught - do nothing!
        }

        in.close();


    }//end run

}//end class

```