TellCalcThread.java

```
1 package tellscopeV4;
 3 //import libraries
 5 import java.net.*;
 8 public class TellCalcThread implements Runnable {
        //default constructor
       public TellCalcThread(Socket socket)
12
13
            this.s = socket;
16
17
        //socket for incoming data from the client
18
19
       private Socket s;
20
       //array to store incoming calculation values
double[] calcValues = new double[3];
21
22
23
24
        //counter for input array - used as index to the array
2.5
        int i = 0;
26
27
        //private attributes to pass to TellsCalculation object
28
29
       private double focalRatio; //store focal ratio
30
       private double lensDiameter;
                                                   //store lens diameter
       private double eyePieceFocalLength; //store eyepiece focal length
31
32
33
       private Scanner in;
34
3.5
       private boolean type;
36
38
        // {\tt variables} \ {\tt to} \ {\tt store} \ {\tt calculated} \ {\tt results}
39
        private static double focalLength;
40
        private static double tubeLength;
       private static double distToSecond;
       private static double secondarySizeMinor;
4.3
        private static double secondarySizeMajor;
44
        private static double minMagnitude;
45
       private static double minResolution;
       private static double maxVisibleMagnification;
47
       private static double minVisibleMagnification;
48
       private static double eyePieceMagnification;
49
50
52
5.3
54
       public void run() {
55
56
            String client = s.getInetAddress().toString();
            TellServerGuiO4.consoleView.append("\nConnected to " + client);
TellServerGuiO4.consoleView.append("\nRunning Calculations");
System.out.println("Connected to " + client);
57
58
59
60
61
            //try creating a new scanner
62
            try
63
                  //create new scanner object and set it to get input stream from socket
65
                 in = new Scanner(s.getInputStream());
66
67
68
                 // 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
70
71
72
                 // ready for sending to client
73
                 while(true)
74
                      //store next line in "input" string
75
                      //check for equality with string "calculate"
76
                      if(input.equalsIgnoreCase("calculate"))
79
                           //add input values to correct attributes for calculations
8.0
                           focalRatio = calcValues[0];
81
                           lensDiameter = calcValues[1];
82
83
                           eyePieceFocalLength = calcValues[2];
84
8.5
86
                           /* write inputs to input text fields */
                           \label{tempt} TellServerGui04.setInputs (String.format ("%.2f", lensDiameter), String.format ("%.2f", focalRatio), String.format ("%.2f", eyePieceFocalLength));
88
89
90
92
93
                           /* moved below for test
94
                           //create new TellsCalculations object and pass user input attributes
TeleScopeReflect testCalcs = new TeleScopeReflect(focalRatio, lensDiameter, eyePieceFocalLength);
95
```

TellCalcThread.java

```
97
 98
                                     /* check the telescope type (reflect == true, refract == false) */
                                     if(type == true)
 99
100
101
                                          //create new TellsCalculations object and pass user input attributes
                                         TeleScopeReflect reflecting = new TeleScopeReflect(focalRatio, lensDiameter,
102
    eyePieceFocalLength);
103
                                         //call calculation methods to get values //convert to string with 2 decimal places % \left( 1\right) =\left( 1\right) ^{2}
104
105
                                          //store in TellServer.calcResults array
106
                                         TellServerGui04.calcResults[0] = String.format("%.2f", reflecting.calcFocalLength());
107
                                         TellServerGui04.calcResults[1] = String.format("%.2f", reflecting.calcTubeLength());
TellServerGui04.calcResults[2] = String.format("%.2f", reflecting.calcDistToSecond());
108
109
                                         TellServerGui04.calcResults[3] = String.format("%.2f",
110
    reflecting.calcSecondarySizeMinor());
111
                                         TellServerGuiO4.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());
112
113
114
115
                                         TellServerGui04.calcResults[9] = String.format("%.2f",
116
    reflecting.calcEvepieceMagnification());
118
119
                                          //once array is populated - add string "result" to inform client to display the results
120
                                         TellServerGui04.calcResults[10] = "result";
121
122
                                          /* place results in server results area */
123
                                         TellServerGui04.setResults();
124
                                          //once calculations are done and stored - create a new thread to pass results to client
125
                                         Thread sendResults = new Thread(new SendResultsThread(s));
126
127
                                         sendResults.start();
128
                                         i = 0;
//DO I NEED A BREAK HERE???????
129
130
                                         continue;
131
132
133
134
                                              else if(type == false)
135
                                                    /create new TellsCalculations object and pass user input attributes
                                                   TeleScopeRefract refracting = new TeleScopeRefract(focalRatio, lensDiameter,
137
    eyePieceFocalLength);
138
139
                                                   //call calculation methods to get values
                                                   //convert to string with 2 decimal places
140
141
                                                    //store in TellServer.calcResults array
142
                                                   TellServerGuiO4.calcResults[0] = String.format("%.2f",
    refracting.calcFocalLength());
143
                                                   TellServerGuiO4.calcResults[1] = String.format("%.2f",
    refracting.calcTubeLength());
144
                                                   TellServerGuiO4.calcResults[2] = "Not Required":
                                                   TellServerGui04.calcResults[3] = "Not Required";
145
146
                                                   TellServerGui04.calcResults[4] = "Not Required";
                                                   TellServerGui04.calcResults[5] = String.format("%.2f",
    refracting.calcMinMagnitude());
148
                                                   TellServerGuiO4.calcResults[6] = String.format("%.2f",
    refracting.calcMinResolution());
149
                                                   TellServerGuiO4.calcResults[7] = String.format("%.2f",
    refracting.calcMaxVisibleMag());
150
                                                   TellServerGui04.calcResults[8] = String.format("%.2f",
    refracting.calcMinVisibleMag());
                                                   TellServerGui04.calcResults[9] = String.format("%.2f",
    refracting.calcEyepieceMagnification());
152
153
                                                   //once array is populated - add string "result" to inform client to display the
154
   results
155
                                                   TellServerGui04.calcResults[10] = "result";
156
157
                                                   //TellServer.printResults();
                                                   //TellServer.sendResultsToClient();
158
159
160
                                                   /* place results in server results area */
161
                                                   TellServerGuiO4.setResults();
162
163
                                                   //once calculations are done and stored - create a new thread to pass results to
   client
164
                                                   Thread sendResults = new Thread(new SendResultsThread(s));
165
                                                   sendResults.start():
166
168
                                                   //DO I NEED A BREAK HERE???????
169
                                                   continue;
170
                                              }
171
172
173
174
175
                      else if(input.equalsIgnoreCase("reflect"))
```

```
TellCalcThread.java
```

```
177
178
                             type = true;
                        }
179
                        else if(input.equalsIgnoreCase("refract"))
180
181
                             type = false;
182
183
184
185
186
187
                             //if input != "calculate"
188
                             else
189
190
191
192
                                  //try parse the input to a double
193
                                  try{
194
                                       //store in calcValues array for sorting later
calcValues[i] = Double.parseDouble(input);
195
196
197
                                        //increment counter for array index
198
199
200
                                       //catch exceptions
catch(NumberFormatException nfe)
201
202
                                            //if exception caught - print "Failed" and exception to console System.out.println("Failed: " + nfe.toString());
203
204
205
206
                                             System.exit(1);
207
                                        }//end nfe exception catch
208
209
                             }//end else
210
211
                   }//end while loop
212
213
214
              }//end top try
catch(Exception e)
215
216
                   //if exception caught - do nothing!
217
218
219
              in.close();
220
221
222
         }//end run
224 }//end class
225
```