

```

2  * Class Name: TellScopeGui04
30 package tellscopeV4;
31
32 /* import required library's */
33 import javax.swing.*;
34
35
36
37
38 /* define class and inherit from JFrame */
39 public class TellScopeGui04 extends JFrame {
40
41     /** RESULTS STORAGE *****/
42     //array list to store calculation results
43     protected static ArrayList<String> results = new ArrayList<String>();
44     //array to copy results to before clearing array list
45     protected static String[] calcResults = new String[13];
46
47     /** END RESULTS STORAGE *****/
48
49     /* set serial id */
50     private static final long serialVersionUID = 1L;
51
52
53
54     /* main method */
55     public static void main(String[] args) {
56         // TODO Auto-generated method stub
57
58         /* call default constructor */
59         new TellScopeGui04();
60
61     }
62
63     /** JLABELS *****/
64     /* Create JLabel attributes for INPUTS */
65     private static JLabel lblLensDiameter; //label to display title Lens Diameter
66     private static JLabel lblFocalRatio; //label to display title Focal Ratio
67     private static JLabel lblEyeFocalLength; //label to display title Eyepiece Focal Length
68
69     /* Create JLabel attributes for RESULTS */
70     private static JLabel lblLensInput; //label to display title Submitted Lens Diameter Input
71     private static JLabel lblFocalInput; //label to display title Submitted Focal Ratio Input
72     private static JLabel lblEyeFocalInput; //label to display title Submitted Eyepiece focal length
73
74     private static JLabel lblFocalLength; //label to display title Focal Length
75     private static JLabel lblTubeDiameter; //label to display title Tube Diameter
76     private static JLabel lblDistToSecond; //label to display title Distance to Second
77     private static JLabel lblSecondSizeMinor; //label to display title Secondary Size Minor
78     private static JLabel lblSecondSizeMajor; //label to display title Secondary Size Major
79     private static JLabel lblMinMagnitude; //label to display title Minimum Magnitude
80     private static JLabel lblMinResolution; //label to display title Minimum Resolution
81     private static JLabel lblMaxVisibleMag; //label to display title Maximum Visible Magnification
82     private static JLabel lblMinVisibleMag; //label to display title Minimum Visible Magnification
83     private static JLabel lblEyePieceMag; //label to display title Eyepiece Magnification
84
85     /** END JLABELS *****/
86
87     /** JTEXTFIELDS *****/
88     /* Create JTextField attributes for INPUTS */
89     protected static JTextField txtLensDiameter; //text field to take lens diameter input
90     protected static JTextField txtFocalRatio; //text field to take focal ratio input
91     protected static JTextField txtEyeFocalLength; //text field to take eyepiece magnification input
92
93     /* Create JTextField attributes for RESULTS */
94     private static JTextField txtLensInput; //text field to display Lens Diameter Input
95     private static JTextField txtFocalInput; //text field to display Focal Ratio Input
96     private static JTextField txtEyeFocalInput; //text field to display Eyepiece Magnification Input
97
98     private static JTextField txtFocalLength; //text field to display Focal Length
99     private static JTextField txtTubeDiameter; //text field to display Tube Diameter
100    private static JTextField txtDistToSecond; //text field to display Distance to Second
101    private static JTextField txtSecondSizeMinor; //text field to display Secondary Size Minor
102    private static JTextField txtSecondSizeMajor; //text field to display Secondary Size Major
103    private static JTextField txtMinMagnitude; //text field to display Minimum Magnitude
104    private static JTextField txtMinResolution; //text field to display Minimum Resolution
105    private static JTextField txtMaxVisibleMag; //text field to display Maximum Visible Magnification
106    private static JTextField txtMinVisibleMag; //text field to display Minimum Visible Magnification
107    private static JTextField txtEyePieceMag; //text field to display Eyepiece Magnification
108
109    /** END JTEXTFIELDS *****/
110
111    /** BUTTONS *****/
112    /* JButton */

```

TellScopeGui04.java

```

129 private static JButton btnSubmit;           //button to submit users inputs to server
130 private static JButton btnSave;             //button to save the results of the calculations
131 private static JButton btnLoad;             //button to load previously calculated results
132 private static JButton btnClear;            //button to clear the results set
133
134 /* JRadioButton */
135
136 protected static JRadioButton reflect, refract; //radio buttons to select between reflecting or refracting telescope
137
138 /** END BUTTONS *****/
139
140
141
142 /**** INPUT PANEL SECTION *****/
143
144
145
146 /* TellScopeGUI4 Constructor */
147
148 public TellScopeGui04()
149 {
150
151
152     /* set up JFrame */
153     this.setSize(1100,600);                 //set default JFrame size
154     this.setTitle("Tell Scope");             //set JFrame title
155     this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //set default close operation
156     this.setLocationRelativeTo(null);        //center the frame on screen
157
158
159     /* create JPanels */
160     JPanel mainPanel = new JPanel();          //create main panel
161     mainPanel.setLayout(new GridBagLayout()); //set layout for main panel
162     //mainPanel.setBackground(Color.BLACK);
163
164     JPanel inputPanel = new JPanel();          //create panel for inputs
165     inputPanel.setLayout(new GridBagLayout()); //set layout for input panel
166     //inputPanel.setForeground(Color.WHITE);
167
168     JPanel resultsPanel = new JPanel();        //create panel for outputs
169     resultsPanel.setLayout(new GridBagLayout()); //set layout for results panel
170     //resultsPanel.setBackground(Color.WHITE);
171
172     JPanel saveLoadPanel = new JPanel();       //create panel for save and load buttons
173     saveLoadPanel.setLayout(new GridBagLayout()); //set layout for saveLoad panel
174
175     /* create borders for panels */
176     Border inputBorder = BorderFactory.createTitledBorder("Inputs"); //create border for inputs panel
177     inputPanel.setBorder(inputBorder);         //add the border to the panel
178
179     Border resultsBorder = BorderFactory.createTitledBorder("Results"); //create border for results panel
180     resultsPanel.setBorder(resultsBorder);      //add border to results panel
181
182     /**** INPUT PANEL SECTION *****/
183
184     /* create input labels */
185     lblLensDiameter = new JLabel("Lens Diameter"); //create new label with title Lens Diameter
186     lblFocalRatio = new JLabel("Focal Ratio");    //create new label with title Focal Ratio
187     lblEyeFocalLength = new JLabel("Eyepiece Focal Length"); //create new label with title Eyepiece Focal Length
188
189     /* create input text fields */
190     txtLensDiameter = new JTextField(10);         //create new text field, set number of columns
191     txtFocalRatio = new JTextField(10);           //create new text field, set number of columns
192     txtEyeFocalLength = new JTextField(10);       //create new text field, set number of columns
193
194     /* create radio buttons */
195
196     refract = new JRadioButton("Refracting");     //create new radio button with title Refracting
197     reflect = new JRadioButton("Reflecting");     //create new radio button with title Reflecting
198     reflect.setSelected(true);
199
200
201     ButtonGroup teleType = new ButtonGroup();     //create new radio button group called teleType
202     teleType.add(reflect);                        //add the reflect radio button to the group
203     teleType.add(refract);                       //add the refract radio button to the group
204
205
206     /* create submit button */
207
208     btnSubmit = new JButton("Submit");            //create submit button
209     btnSubmit.setFont(new Font("Tahoma", Font.PLAIN, 9)); //set font and font size of button
210
211     SubmitListener submitListen = new SubmitListener();
212     btnSubmit.addActionListener(submitListen);
213

```

```

214
215 /**** GRIDBAG LAYOUT CONSTRAINTS *****/
216
217 /* create new grid bag constraints object to help set components in place */
218 GridBagConstraints gc = new GridBagConstraints(); //create new grid constraints object
219 gc.gridx = 0; //set default grid x
220 gc.gridy = 0; //set default grid y
221 gc.gridwidth = 1; //set default grid width
222 gc.gridheight = 1; //set default grid height
223 gc.weightx = 100.0; //set default row width
224 gc.weighty = 100.0; //set default row height
225 gc.insets = new Insets(5,25,15,25); //set default padding
226 gc.anchor = GridBagConstraints.CENTER; //set default alignment if component does not fill space
227 gc.fill = GridBagConstraints.NONE; //set default fill value (?fill available space)
228
229 /*****/
230
231 /* add components to the input panel */
232 //labels
233 inputPanel.add(lblLensDiameter, gc); //add lens diameter label
234 gc.gridx = 1; //set grid x position for focal label
235 inputPanel.add(lblFocalRatio, gc); //add focal ratio label
236 gc.gridx = 2; //set grid x position for eye mag label
237 inputPanel.add(lblEyeFocalLength, gc); //add eye mag label
238
239 //text fields
240 gc.gridx = 0; //set grid x position for lens diam text field
241 gc.gridy = 1; //set grid y position for lens diam text field
242 inputPanel.add(txtLensDiameter, gc); //add lens diam text field to the panel
243 gc.gridx = 1; //set grid x position for focal ratio text field
244 gc.gridy = 1; //set grid y position for focal ratio text field
245 inputPanel.add(txtFocalRatio, gc); //add focal ratio text field to the panel
246 gc.gridx = 2; //set grid x position for eye mag text field
247 gc.gridy = 1; //set grid y position for eye mag text field
248 inputPanel.add(txtEyeFocalLength, gc); //add eye mag text field to the panel
249
250 //radio button group
251 gc.gridx = 3;
252 gc.gridy = 0;
253 inputPanel.add(reflect, gc);
254
255 gc.gridx = 3;
256 gc.gridy = 1;
257 inputPanel.add(refract, gc);
258
259 //buttons
260 gc.gridx = 3; //set grid x position for submit button
261 gc.gridy = 2; //set grid y position for submit button
262 inputPanel.add(btnSubmit, gc); //add the button to the input panel
263
264
265 //reset the grid x and y values for grid bag constraints "gc"
266 gc.gridx = 0;
267 gc.gridy = 0;
268
269
270 /**** END INPUT PANEL SECTION *****/
271
272 /**** RESULTS PANEL SECTION *****/
273
274 /* create results labels */
275
276 //labels for submitted user inputs
277 lblLensInput = new JLabel("Lens Diameter"); //create new label with title lens diameter
278 lblFocalInput = new JLabel("Focal Ratio"); //create new label with title focal ratio
279 lblEyeFocalInput = new JLabel("Eyepiece Focal Length"); //create new label with title eyepiece mag
280
281 //calculation results labels
282 lblFocalLength = new JLabel("Focal Length"); //create new label with title focal length
283 lblTubeDiameter = new JLabel("Tube Diameter"); //create new label with title tube diam
284 lblDistToSecond = new JLabel("Distance to Secondary"); //create new label with title dist to second
285 lblSecondSizeMinor = new JLabel("Secondary Size Min"); //create new label with title second size minor
286 lblSecondSizeMajor = new JLabel("Secondary Size Maj"); //create new label with title second size major
287 lblMinMagnitude = new JLabel("Minimum Magnitude"); //create new label with title minimum mag
288 lblMinResolution = new JLabel("Minimum Resolution"); //create new label with title min resolution
289 lblMaxVisibleMag = new JLabel("Max Visible Magnification"); //create new label with title max visible mag
290 lblMinVisibleMag = new JLabel("Min Visible Magnification"); //create new label with title min visible mag
291 lblEyePieceMag = new JLabel("Eyepiece Magnification"); //create new label with title eyepiece magnification
292
293
294 //create results text fields */
295 //text fields to display submitted user inputs
296 txtLensInput = new JTextField(10); //create new text field for submitted lens input
297 txtLensInput.setEditable(false); //make text field un-editable
298 txtFocalInput = new JTextField(10); //create new text field for submitted focal input
299 txtFocalInput.setEditable(false); //make text field un-editable
300 txtEyeFocalInput = new JTextField(10); //create new text field for submitted eyepiece mag
301

```

```

302         txtEyeFocalInput.setEditable(false); //make text field un-editable
303
304 //calculation results text fields, not editable!!!
305 txtFocalLength = new JTextField(10); //create new text field for focal length
306 txtFocalLength.setEditable(false); //make text field un-editable
307 txtTubeDiameter = new JTextField(10); //create new text field for tube diameter
308 txtTubeDiameter.setEditable(false); //make text field un-editable
309 txtDistToSecond = new JTextField(10); //create new text field for dist to second
310 txtDistToSecond.setEditable(false); //make text field un-editable
311 txtSecondSizeMinor = new JTextField(10); //create new text field for second size minor
312 txtSecondSizeMinor.setEditable(false); //make text field un-editable
313 txtSecondSizeMajor = new JTextField(10); //create new text field for second size major
314 txtSecondSizeMajor.setEditable(false); //make text field un-editable
315 txtMinMagnitude = new JTextField(10); //create new text field for min magnitude
316 txtMinMagnitude.setEditable(false); //make text field un-editable
317 txtMinResolution = new JTextField(10); //create new text field for min resolution
318 txtMinResolution.setEditable(false); //make text field un-editable
319 txtMaxVisibleMag = new JTextField(10); //create new text field for max visible magnification
320 txtMaxVisibleMag.setEditable(false); //make text field un-editable
321 txtMinVisibleMag = new JTextField(10); //create new text field for min visible magnification
322 txtMinVisibleMag.setEditable(false); //make text field un-editable
323 txtEyePieceMag = new JTextField(10); //create new text field for eyepiece
324 txtEyePieceMag.setEditable(false); //make text field un-editable
325
326 /***** SAVE, LOAD AND CLEAR BUTTONS *****/
327
328 /* create save button */
329 btnSave = new JButton("Save"); //create save button
330 btnSave.setFont(new Font("Tahoma", Font.PLAIN, 9)); //set font and font size of button
331
332 /* create load button */
333 btnLoad = new JButton("Load"); //create load button
334 btnLoad.setFont(new Font("Tahoma", Font.PLAIN, 9)); //set font and size of button
335
336 /* create clear button */
337 btnClear = new JButton("Clear"); //create clear button to clear the results set
338 btnClear.setFont(new Font("Tahoma", Font.PLAIN, 9)); //set font and size of button
339
340 /***** SAVE, LOAD AND CLEAR ACTION LISTENERS *****/
341
342 /* add action listener to save button */
343 SaveListener sl = new SaveListener(); //create new SaveListener object
344 btnSave.addActionListener(sl); //add SaveListener object to the save button
345
346 /* add action listener to load button */
347 LoadListener ll = new LoadListener(); //create new LoadListener object
348 btnLoad.addActionListener(ll); //add LoadListener to the load button
349
350 /* add action listener to clear button */
351 ClearListener cl = new ClearListener(); //create new ClearListener object
352 btnClear.addActionListener(cl); //add ClearListener to clear button
353
354 /***** END SAVE, LOAD AND CLEAR ACTION LISTENERS *****/
355
356 /*****
357
358 /* add components to the output panel */
359
360 //labels
361 //submitted inputs labels
362 gc.gridx = 1; //set grid x position
363 resultsPanel.add(lblLensInput, gc); //add submitted lens diameter label
364 gc.gridx = 2; //set grid x position
365 resultsPanel.add(lblFocalInput, gc); //add submitted focal length input
366 gc.gridx = 3; //set grid x position
367 resultsPanel.add(lblEyeFocalInput, gc); //add submitted eyepiece focal length
368
369 //calculation results labels
370 //first row (y = 2 for layout)
371 gc.gridx = 0; //set grid x position
372 gc.gridy = 2; //set grid y position
373 resultsPanel.add(lblFocalLength, gc); //add focal length label
374 gc.gridx = 1; //set grid x position
375 gc.gridy = 2; //set grid y position
376 resultsPanel.add(lblTubeDiameter, gc); //add tube diameter label
377 gc.gridx = 2; //set grid x position
378 gc.gridy = 2; //set grid y position
379 resultsPanel.add(lblDistToSecond, gc); //add dist to second label
380 gc.gridx = 3; //set grid x position
381 gc.gridy = 2; //set grid y position
382 resultsPanel.add(lblSecondSizeMinor, gc); //add second size min label
383 gc.gridx = 4; //set grid x position
384 gc.gridy = 2; //set grid y position
385 resultsPanel.add(lblSecondSizeMajor, gc); //add second size maj label
386
387 //second row (y = 4 for layout)
388 gc.gridx = 0; //set grid x position
389 gc.gridy = 4; //set grid y position

```

```

390     resultsPanel.add(lblMinMagnitude, gc);           //add minimum magnitude label
391     gc.gridx = 1;                                   //set grid x position
392     gc.gridy = 4;                                   //set grid y position
393     resultsPanel.add(lblMinResolution, gc);         //add minimum resolution label
394     gc.gridx = 2;                                   //set grid x position
395     gc.gridy = 4;                                   //set grid y position
396     resultsPanel.add(lblMaxVisibleMag, gc);         //add max visible magnification label
397     gc.gridx = 3;                                   //set grid x position
398     gc.gridy = 4;                                   //set grid y position
399     resultsPanel.add(lblMinVisibleMag, gc);         //add min visible magnification label
400     gc.gridx = 4;                                   //set grid x position
401     gc.gridy = 4;                                   //set grid y position
402     resultsPanel.add(lblEyePieceMag, gc);           //add eyepiece magnification label
403
404
405     //text fields
406     //submitted inputs text fields
407     gc.gridx = 1;                                   //set grid x position
408     gc.gridy = 1;                                   //set grid y position
409
410     resultsPanel.add(txtLensInput, gc);             //add lens input text field to results panel
411     gc.gridx = 2;                                   //set grid x position
412     gc.gridy = 1;                                   //set grid y position
413     resultsPanel.add(txtFocalInput, gc);           //add focal ratio text field to results panel
414     gc.gridx = 3;                                   //set grid x position
415     gc.gridy = 1;                                   //set grid y position
416     resultsPanel.add(txtEyeFocalInput, gc);         //add eye focal length text field to results panel
417
418     //calculation results text fields
419     // row 1 (y = 3)
420     gc.gridx = 0;                                   //set grid x position
421     gc.gridy = 3;                                   //set grid y position
422     resultsPanel.add(txtFocalLength, gc);           //add focal length text field to results panel
423     gc.gridx = 1;                                   //set grid x position
424     gc.gridy = 3;                                   //set grid y position
425     resultsPanel.add(txtTubeDiameter, gc);         //add tube diameter text field to results panel
426     gc.gridx = 2;                                   //set grid x position
427     gc.gridy = 3;                                   //set grid y position
428     resultsPanel.add(txtDistToSecond, gc);         //add dist to second text field to results panel
429     gc.gridx = 3;                                   //set grid x position
430     gc.gridy = 3;                                   //set grid y position
431     resultsPanel.add(txtSecondSizeMinor, gc);       //add secondary size minor text field to results
panel
432     gc.gridx = 4;                                   //set grid x position
433     gc.gridy = 3;                                   //set grid y position
434     resultsPanel.add(txtSecondSizeMajor, gc);       //add secondary size major text field to results
panel
435
436
437     //row 2 (y = 5)
438     gc.gridx = 0;                                   //set grid x position
439     gc.gridy = 5;                                   //set grid y position
440     resultsPanel.add(txtMinMagnitude, gc);         //add minimum magnitude text field to results panel
441     gc.gridx = 1;                                   //set grid x position
442     gc.gridy = 5;                                   //set grid y position
443     resultsPanel.add(txtMinResolution, gc);         //add min resolution text field to results panel
444     gc.gridx = 2;                                   //set grid x position
445     gc.gridy = 5;                                   //set grid y position
446     resultsPanel.add(txtMaxVisibleMag, gc);         //add max visible mag text field to results panel
447     gc.gridx = 3;                                   //set grid x position
448     gc.gridy = 5;                                   //set grid y position
449     resultsPanel.add(txtMinVisibleMag, gc);         //add min visible mag text field to results panel
450     gc.gridx = 4;                                   //set grid x position
451     gc.gridy = 5;                                   //set grid y position
452     resultsPanel.add(txtEyePieceMag, gc);           //add eyepiece mag text field to results panel
453
454
455     //buttons
456     gc.gridx = 0;                                   //set grid x position for save button
457     gc.gridy = 0;                                   //set grid y position for save button
458     gc.insets = new Insets(5,5,5,5);               //add the button to the results panel
459     saveLoadPanel.add(btnSave, gc);
460     gc.gridx = 1;
461     saveLoadPanel.add(btnLoad, gc);
462
463     gc.insets = new Insets(5,5,5,5);
464     gc.gridx = 0;
465     gc.gridy = 6;
466     resultsPanel.add(saveLoadPanel, gc);
467
468
469     gc.gridx = 4;
470     resultsPanel.add(btnClear, gc);
471
472
473     /**** END RESULTS PANEL SECTION *****/
474
475     /**** ADD SUB PANELS TO MAIN PANEL *****/

```

```

476
477     /* add sub panels to main panel */
478     gc.gridx = 0;                                //set grid x position for input panel
479     gc.gridy = 0;                                //set grid y position for input panel
480
481     //add input panel
482     mainPanel.add(inputPanel, gc);                //add input panel to main panel
483
484
485     //add results panel
486     gc.gridx = 0;                                //set grid x position for results panel
487     gc.gridy = 1;                                //set grid y position for results panel
488     mainPanel.add(resultsPanel, gc);              //add results panel to main panel
489
490     /**** END ADD SUB PANELS TO MAIN PANEL *****/
491
492     /**** ADD PANEL TO FRAME AND MAKE FRAME VISIBLE *****/
493
494     /* add the main panel to this JFrame */
495     this.add(mainPanel);                          //add main panel to the frame
496     //this.pack();
497     /* set JFrame visibility to visible */
498     this.setVisible(true);                        //set the frame visibility to true
499
500     /**** END ADD PANEL TO FRAME AND MAKE FRAME VISIBLE *****/
501
502 } //end TellscopeGui04 constructor
503
504
505
506
507 /***** PUBLIC METHODS *****/
508
509 /*
510  * Method name: printResults()
511  * Description: test method that prints the results to the console
512  * Return: void
513  *
514  */
515 public static void printResults()
516 {
517
518     for (String s : results)
519     {
520         System.out.println(s);
521     }
522
523 }
524
525
526
527 /*
528  * Method name: checkInputForNull()
529  * Description: checks if any of the user input boxes are null
530  * Return: boolean --> true if fields not null, false if any fields are null
531  *
532  */
533 public static boolean checkInputForNull()
534 {
535
536     if(txtLensDiameter.getText().equals("") | txtFocalRatio.getText().equals("")
537        | txtEyeFocalLength.getText().equals("")) //check if any of the text fields are empty
538     {
539         return false; //return false if there are empty fields
540     }
541
542     else
543     {
544         return true; //return true if all fields are not null
545     }
546 }
547
548 /*****
549
550
551 /*
552  * Method name: checkInputForChars()
553  * Description: checks if any of the user input boxes contain any values other than numeric values
554  * Return: boolean --> true if fields only contain 1-9, false if any fields contain chars or symbols
555  *
556  */
557 public static boolean checkInputForChars()
558 {
559     /* check if any of the input boxes contain anything other than numeric values */
560     if(txtLensDiameter.getText().matches("[1-9].*") | txtFocalRatio.getText().matches("[1-9].*")
561        | txtEyeFocalLength.getText().matches("[1-9].*"))
562     {
563         return true; //return true if only numeric values

```

```

564     }
565     else
566     {
567         /* clear user input fields */
568         txtFocalRatio.setText(null);           //clear user focal ratio input
569         txtLensDiameter.setText(null);         //clear user lens diam input
570         txtEyeFocalLength.setText(null);       //clear user eye focal length input
571
572         return false;                          //return false if contains chars or symbols
573     }
574 }
575
576
577
578 /***** END PUBLIC METHODS *****/
579
580
581
582
583 /***** PROTECTED METHODS *****/
584
585 /*
586  * Method name: getSocket()
587  * Description: takes the port number as an argument, creates a socket based on the host
588  *              and port number
589  * Return: socket
590  * */
591
592 protected static Socket getSocket(int port)
593 {
594     Socket s;                                //create new socket attribute
595     String host = "localhost";               //create new string "host" and set value to "localhost"
596     InetAddress ip;                          //create new inetAddress object
597
598
599     Scanner sc = new Scanner(System.in);     //create new scanner object
600     while(true)                             //start new while true loop
601     {
602
603
604         try                                //try catch block for getting new socket
605         {
606             ip = InetAddress.getByName(host); //use getByName() method to get ip address of localhost
607             s = new Socket(ip, port);         //create socket object based on ip and port number
608             return s;                         //return the socket
609
610         }
611         catch(IOException e)                //catch exception if
612         {
613             there is a network error
614             System.out.println("Network Error"); //print to console
615             JOptionPane.showMessageDialog(txtFocalInput, "No network connection, make sure you" //open dialog box and
616             display error message to user
617             + " have started the server!");
618
619             break;
620         }
621     }
622     sc.close();
623     return null;
624 } //end getSocket()
625
626 /*****
627
628 /*
629  * Method name: copyResults()
630  * Description: protected method to copy results from array list to array,
631  *              as array list is cleared ready for next operation
632  * Return: void
633  * */
634
635 protected static void copyResults()
636 {
637     int j = 3;                                //create dummy int for counter
638
639     for (String s : results)                  //for each string...loop
640     {
641         calcResults[j] = s;                  //use j as index to copy value
642         j++;                                //increment j
643     }
644 } //end copyResults()
645
646
647 /*****
648

```



```

649  /*
650  * Method name: setResults()
651  * Description: protected method to print results to the appropriate text field
652  * Return: void
653  *
654  * */
655  protected static void setResults()
656  {
657      /* set user input values */
658      txtLensInput.setText(calcResults[0]);           //set user lens diam input to results area
659      txtFocalInput.setText(calcResults[1]);           //set user focal ratio input to results area
660      txtEyeFocalInput.setText(calcResults[2]);        //set user eye focal input to results area
661
662      /* set results values */
663      txtFocalLength.setText(calcResults[3]);          //set focal length result
664      txtTubeDiameter.setText(calcResults[4]);         //set tube diam result
665      txtDistToSecond.setText(calcResults[5]);         //set dist to second result
666      txtSecondSizeMinor.setText(calcResults[6]);      //set second size min result
667      txtSecondSizeMajor.setText(calcResults[7]);      //set second size maj result
668      txtMinMagnitude.setText(calcResults[8]);         //set min magnitude result
669      txtMinResolution.setText(calcResults[9]);        //set min res result
670      txtMaxVisibleMag.setText(calcResults[10]);       //set max vis mag result
671      txtMinVisibleMag.setText(calcResults[11]);       //set min vis mag result
672      txtEyePieceMag.setText(calcResults[12]);         //set eyepiece mag result
673
674
675  } //end setResults
676
677  /*
678  * Method name: resetInputs()
679  * Description: calls private method >>> copyAndStoreInputs
680  * Return: void
681  *
682  * */
683  protected static void resetInputs()
684  {
685      copyAndStoreInputs();                          //call private method >>> copy and store inputs
686  }
687
688  /*****
689
690  /*
691  * Method name: storeResults()
692  * Description: writes the results array to file, user prompted
693  *               for filename in save listener class, which calls
694  *               this method.
695  * Input: String >>> filename
696  * Return: void
697  *
698  * */
699  protected static void storeResults(String s)
700  {
701
702      //create local attribute for filename acquired from user
703      String filename = s;                          //store input filename to filename attribute
704
705      /* validation */
706      /* check there are values to save */
707      if(txtLensInput.getText().equals(""))          //if text field is empty
708      {
709          JOptionPane.showMessageDialog(txtFocalInput, "No values to save!" //display error message in JDialogBox
710          , "Save file error", JOptionPane.ERROR_MESSAGE);
711      }
712
713      else                                           //else if text field is not empty
714      {
715          try                                       //try catch to handle exceptions
716          {
717              FileWriter fw = new FileWriter(filename + ".txt"); //create new file writer object and pass filename
718              PrintWriter pw = new PrintWriter(fw); //create new print writer object and pass file
719
720              writer
721
722              for(int j = 0; j<13; j++)           //for loop to count through array
723              {
724                  pw.println(calcResults[j]);      //write results set to file
725                  //System.out.println(calcResults[j]); //used for testing only! check the values that
726                  are being saved
727              }
728
729              System.out.println("!!! File saved !!!"); //print file saved message
730              pw.close();                          //close the print writer
731
732          }
733          catch(IOException e)                   //catch any exceptions
734          {

```



```

735         System.out.println("Error!!!");           //print error message to console
736     }
737 }
738 }
739 }
740 } //end storeResults()
741
742
743 /*****
744
745  /*
746   * Method Name: loadResults()
747   * Function: loads results from a text file
748   * Input: Filename from inputMessageDialog user entry
749   * Return: void
750   * */
751  protected static void loadResults(String s)           //declare method and arguments
752  {
753      String filename = s;                             //create string to store filename and set == to users
754      filename input
755      int j = 0;                                         //create int to use as counter for the array
756
757      try                                                //try catch to to catch any exceptions
758      {
759          FileReader fr = new FileReader(filename + ".txt"); //create new filereader object and pass filename
760          BufferedReader br = new BufferedReader(fr);        //create new buffered reader object and pass filereader
761          String str ;                                       //create string to store each value from readLine()
762
763          while((str = br.readLine()) != null)             //while loop >>> while input != null, continue
764          {
765              calcResults[j] = str;                       //store input into results array
766              j++;                                         //increment counter used as array index
767          }
768
769          System.out.println("!!! File loaded !!!");       //print "file loaded" message to console
770          setResults();                                   //call set results method to allocate the result set to
771          correct txt fields
772          br.close();                                     //close the buffered reader
773      }
774      catch(IOException e)                               //catch any exceptions
775      {
776          System.out.println("Error!!!");                 //print error message to
777          console
778          JOptionPane.showMessageDialog(txtFocalInput, "File not found, please try again."
779          , "File Loading Error", JOptionPane.ERROR_MESSAGE); //open dialog box to display
780          error msg to user
781      }
782
783 /*****
784
785  *****/
786  /***** END PROTECTED METHODS *****/
787
788  /***** PRIVATE METHODS *****/
789
790  /*
791   * Method Name: setInputs()
792   * Function: Print the user inputs to the inputs text fields in results section,
793   *           Clear the input text fields.
794   * */
795  private static void copyAndStoreInputs()
796  {
797      /* store users input to results array */
798      calcResults[0] = txtLensDiameter.getText();        //get user input for lens diameter and store in local
799      attribute
800      calcResults[1] = txtFocalRatio.getText();           //get user input for lens focal ratio and store in local
801      attribute
802      calcResults[2] = txtEyeFocalLength.getText();       //get user input for lens eyepiece focal length and store in
803      local attribute
804
805      /* clear user input fields */
806      txtFocalRatio.setText(null);                        //clear user focal ratio input
807      txtLensDiameter.setText(null);                     //clear user lens diam input
808      txtEyeFocalLength.setText(null);                   //clear user eye focal length input
809
810      } //end copyInputs
811
812  /***** END PRIVATE METHODS *****/
813  } //end class
814

```