Exception Handling and File Processing

Exception Handling and File Processing

Thomas Culpepper

```
100  /**
101 * The CalcTaxes program implements an application that requests tax information
102 * from the user and then calculates taxes due.
103 *
104 * Assignment: CSC310 Mod 1 Case
105 *
106 * @author  Thomas Culpepper
107 * @version 1.0
108 * @since   2021-12-28
109 */
110 import javax.swing.JOptionPane;
111 import java.util.regex.*;
112
113 public class CalcTaxes {
114     public static void main(String[] args) {
115         boolean calcAgain = true; //control to run again or exit
116
117         JOptionPane.showMessageDialog(
118             null, "This program will calculate your federal and state taxes");
119
120         while (calcAgain) {
121             String validationPattern = null;
122             String errorMessage = null;
123             String [] userInputs = new String[4]; //array to hold user input str
124             String [][] inputRequests = { //input requests and expected type (nu
125                 {"Please enter your name:","str"},
126                 {"Enter your yearly income","num"},
127                 {"Enter your Federal tax rate (%):","num"},
128                 {"Enter your State tax rate (%):","num"}
129             };
130             for (int i=0; i < inputRequests.length; i++) {
131                 userInputs[i] = JOptionPane.showInputDialog(inputRequests[i][0])
132                 if (userInputs[i] == null) { // Exit cleanly if user hits cancel
133                     System.exit(0);
```

```
134                 }
135                 else if (inputRequests[i][1].equals("num")){
136                     validationPattern = "^\\d+$|^-?\\d*\\.\\d{2}$"; // match int
137                     errorMessage = "Please enter a number \n(000 or 00.00)";
138                 }
139                 else if (inputRequests[i][1].equals("str")) {
140                     validationPattern = "^[A-Za-z]+\\s*[A-Za-z]*$"; // match nam
141                     errorMessage = "Try again./nNo numbers or symbols please";
142                 }
143                 else {
144                     JOptionPane.showMessageDialog(
145                         null, "Illegal Input type Option. Please contact the dev
146                     throw new IllegalArgumentException(
147                         "Input type option invalid. Only 'num' or 'str' allowed"
148                 }
149 
150                 Pattern p = Pattern.compile(validationPattern);
151                 Matcher m = p.matcher(userInputs[i]);
152                 if (!m.find()){
153                     JOptionPane.showMessageDialog(null, errorMessage);
154                     i--;
155                 }
156             }
157 
158             double yearlyIncome = Double.parseDouble(userInputs[1]);
159             double fedTaxRate = Double.parseDouble(userInputs[2]) / 100;
160             double stateTaxRate = Double.parseDouble(userInputs[3]) / 100;
161             double fedTaxDue = yearlyIncome * fedTaxRate;
162             double stateTaxDue = yearlyIncome * stateTaxRate;
163 
164             JOptionPane.showMessageDialog(null, userInputs[0] + "\nYour Federal 
165                 + fedTaxDue + "\nYour State taxes are: $" + stateTaxDue);
166 
167             int reply = JOptionPane.showConfirmDialog(null, "Would you like to c
168                 "Run Again?",  JOptionPane.YES_NO_OPTION);
169             if (reply == JOptionPane.NO_OPTION) {
170                     calcAgain = false;
171             }
172         }
173     }
174 }
```

User Screens

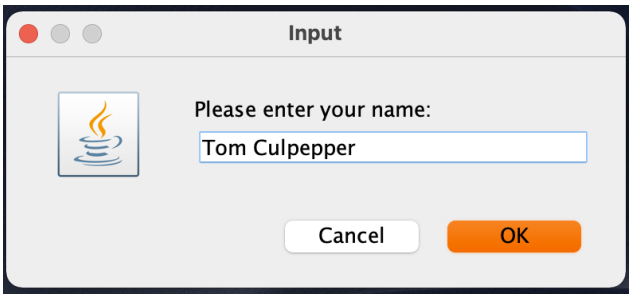

Figure 1: CalcTaxes Screen 1

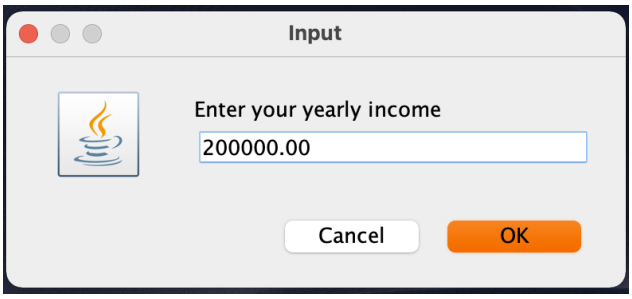

Figure 2: CalcTaxes Screen 2

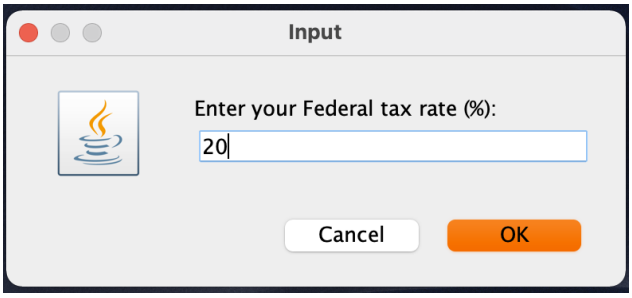

Figure 3: CalcTaxes Screen 3

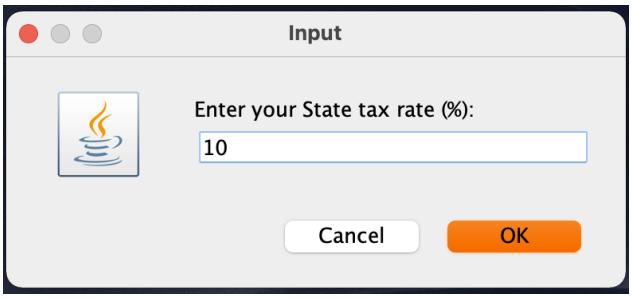

Figure 4: CalcTaxes Screen 4

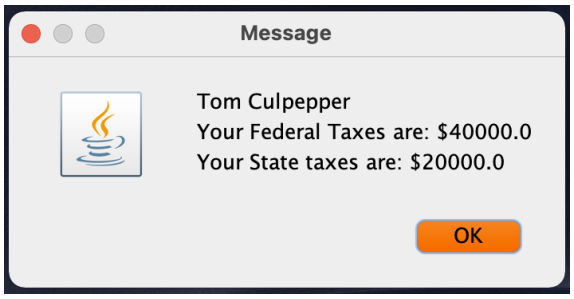

Figure 5: CalcTaxes Screen 5

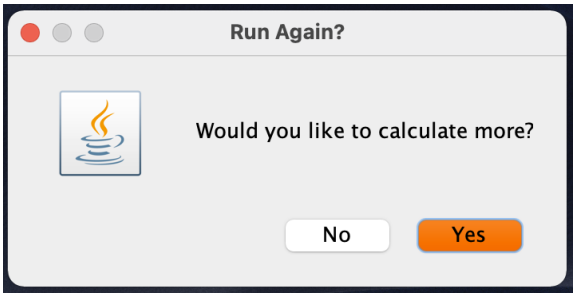

Figure 6: CalcTaxes Screen 6