```java
import java.awt.Color;
import java.util.Scanner;
import javax.swing.JOptionPane;

public class RatesMethodsExtreme {
    // initialization of arrays and variable to the object member variable level
    private GTerm gt;
    private String[] names;
    private char[] genders;
    private int[] birthYears;
    private double[] hourlyRates;
    private Boolean[] terms;
    private int currentStaff;
    private int maxNumberStaff;
    private String inputs;
    private Scanner recordScan;

    public RatesMethodsExtreme(int userInterfaceMode) {
        // initial creation of all strings as well as the input to control which
        // interface you want
        this.maxNumberStaff = 1;
        this.names = new String[this.maxNumberStaff];
        this.genders = new char[this.maxNumberStaff];
        this.birthYears = new int[this.maxNumberStaff];
        this.hourlyRates = new double[this.maxNumberStaff];
        this.terms = new Boolean[this.maxNumberStaff];
        if (userInterfaceMode == 0) {
            this.recordScan = new Scanner(System.in);
            console();
        } else if (userInterfaceMode == 1) {
            gterm();
        } else {
            JOptionPane.showMessageDialog(null, "You wood-duck");
        }

    }

    // gterm main that calls on the methods as needed by buttons and text field entry
    public void gterm() {
        this.gt = new GTerm(800, 800);
        this.gt.setXY(70, 50);
        this.gt.addTable(700, 600, "Name\tGender\tBirth Year\tHourly Rate\tTerms Accepted");
        this.gt.setFontSize(16);
        this.gt.setFontColor(Color.RED);
        this.gt.setBackgroundColor(Color.GRAY);
        // makes button go down bottom
        this.gt.println("");
        // adds buttons with a space to move last 2 down added refresh because why not
        this.gt.addButton("Add Record", this, "addRecord");
        this.gt.addButton("Edit", this, "butEdit");
        this.gt.addButton("Remove", this, "butRemove");
        this.gt.println("");
        this.gt.addButton("Console", this, "toConsole");
        this.gt.addButton("Refresh", this, "refreshTable");
        // initializing array lengths to 1
        this.gt.println("");
        this.gt.setFontColor(Color.BLACK);
        // enters line of text on gterm window and a text field under it for inputs
        this.gt.println("Enter Name, Gender, Year of Birth, Hourly rate, True if Correct");
        this.gt.addTextField("", 500);
    }

    // method to swap from console to gterm
    public void toGterm() {
        gterm();
        refreshTable();
    }

    // extra method to add refresh table as couldn't call 2 from button???
    public void butRemove() {
        remove();
        refreshTable();
    }

    // same as above just for edit button
    public void butEdit() {
        edit();
```

```
78          refreshTable();
79      }
80
81      // method to change from gterm to console
82      public void toConsole() {
83          this.gt.close();
84          this.recordScan = new Scanner(System.in);
85          conRefresh();
86          console();
87      }
88
89      public void console() {
90          // console version main page has 4 selections that call on methods depending on
91          // user input
92          System.out.println(
93                  "Welcome to Staff record keeper 2.0\nPlease enter option\n1. Add record\n2. Edit record \n3.
    Delete record \n4. Go to gterm ");
94          int input = this.recordScan.nextInt();
95          if (input == 1) {
96              conAddRecord();
97          } else if (input == 2) {
98              edit();
99              conRefresh();
100             console();
101         } else if (input == 3) {
102             remove();
103             conRefresh();
104             console();
105         } else if (input == 4) {
106             toGterm();
107         }
108     }
109
110     public void conAddRecord() {
111         // adds record for the console version calling 3 methods
112         System.out.println("Enter Name, Gender, Year of Birth, Hourly rate, True if Correct");
113         this.inputs = this.recordScan.next();
114         allData();
115         conRefresh();
116         console();
117     }
118
119     public void addRecord() {
120         // gets user input to be added to table from the text field
121         this.inputs = this.gt.getTextFromEntry(0);
122         allData();
123         System.out.println(maxNumberStaff);
124         System.out.println(currentStaff);
125         refreshTable();
126     }
127
128     public void allData() {
129         if (this.inputs != null) {
130             // splits the user inputs separated by a comma to store in element zones in
131             // inputData array
132             String[] inputData = this.inputs.split(",");
133             String name = inputData[0];
134             while (name.isBlank()) {
135                 name = JOptionPane.showInputDialog("You have not entered name correctly please try again");
136             }
137             char gender = inputData[1].charAt(0);
138             while (gender != 'm' && gender != 'M' && gender != 'f' && gender != 'F') {
139                 gender = JOptionPane.showInputDialog("You entered an incorrect value for gender\nplease input M or
    F")
140                         .charAt(0);
141             }
142             int birthYear = Integer.parseInt(inputData[2]);
143             while (birthYear < 1900) {
144                 birthYear = Integer.parseInt(
145                         JOptionPane.showInputDialog("The entered year Value may not be valid\nplease try again"));
146             }
147             double hourlyRate = Double.parseDouble(inputData[3]);
148             while (hourlyRate < 15.00 || hourlyRate > 99) {
149                 hourlyRate = Double.parseDouble(JOptionPane
150                         .showInputDialog("Please check value of Hourly Rate\nthe entered value was not valid"));
151             }
152             Boolean term = Boolean.parseBoolean(inputData[4]);
```

```
153             while (!term) {
154                 term = Boolean.parseBoolean(JOptionPane.showInputDialog("Too bad you must enter True"));
155             }
156             // increasing size of the arrays if there is not enough room left
157             if (this.currentStaff >= this.maxNumberStaff) {
158                 // increments the strings by 1 to give additional spot for input
159                 this.maxNumberStaff += 1;
160                 // creation of temp used arrays with new length set
161                 String [] longerNames = new String[this.maxNumberStaff];
162                 char [] longerGender = new char[this.maxNumberStaff];
163                 int[] longerBirth = new int[this.maxNumberStaff];
164                 double[] longerWage = new double[this.maxNumberStaff];
165                 Boolean[] longerTerm = new Boolean[this.maxNumberStaff];
166                 // new counter initialization
167                 int j = 0;
168                 // transfers the elemental data from the existing arrays to the longer temp ones
169                 while (j < this.currentStaff) {
170                     longerNames[j] = this.names[j];
171                     longerGender[j] = this.genders[j];
172                     longerBirth[j] = this.birthYears[j];
173                     longerWage[j] = this.hourlyRates[j];
174                     longerTerm[j] = this.terms[j];
175                     j++;
176                 }
177                 // creates the original arrays back from the temp ones so the originals now have
178                 // a longer size with all datum still stored from previous
179                 this.names = longerNames;
180                 this.genders = longerGender;
181                 this.birthYears = longerBirth;
182                 this.hourlyRates = longerWage;
183                 this.terms = longerTerm;
184             }
185             // sends all variables above to the element chosen by counter currentStaff to
186             // there set arrays
187             this.names[this.currentStaff] = name;
188             this.genders[this.currentStaff] = gender;
189             this.birthYears[this.currentStaff] = birthYear;
190             this.hourlyRates[this.currentStaff] = hourlyRate;
191             this.terms[this.currentStaff] = term;
192             this.currentStaff += 1;
193         }
194     }
195
196     public void edit() {
197         // selection of which staff member they wish to edit
198         int selection = Integer.parseInt(JOptionPane.showInputDialog("enter which staff member you want to
    edit:"));
199         // changes the selection given to the elemental number
200         selection = (selection - 1);
201         this.inputs = JOptionPane.showInputDialog("Enter Name, Gender, Year of Birth, Hourly rate, True if
    Correct");
202         // creates string for the input from diag and splits them and from there is
203         // added to individual element zone with values needed has loops for correct
204         // inputs
205         String [] inputData = this.inputs.split(",");
206         String name = inputData[0];
207         while (name.isBlank()) {
208             name = JOptionPane.showInputDialog("You have not entered name correctly please try again");
209         }
210         char gender = inputData[1].charAt(0);
211         while (gender != 'm' && gender != 'M' && gender != 'f' && gender != 'F') {
212             gender = JOptionPane.showInputDialog("You entered an incorrect value for gender\nplease input M or F")
213                     .charAt(0);
214         }
215         int birthYear = Integer.parseInt(inputData[2]);
216         while (birthYear < 1900) {
217             birthYear = Integer
218                     .parseInt(JOptionPane.showInputDialog("The entered year Value may not be valid\nplease try
    again"));
219         }
220         double hourlyRate = Double.parseDouble(inputData[3]);
221         while (hourlyRate < 15.00 || hourlyRate > 99) {
222             hourlyRate = Double.parseDouble(
223                     JOptionPane.showInputDialog("Please check value of Hourly Rate\nthe entered value was not
    valid"));
224         }
225         Boolean term = Boolean.parseBoolean(inputData[4]);
```

```
226         while (!term) {
227             term = Boolean.parseBoolean(JOptionPane.showInputDialog("Too bad you must enter True"));
228         } // changes the given selection(element) and updates it with the new data given
229         this.names[selection] = name;
230         this.genders[selection] = gender;
231         this.birthYears[selection] = birthYear;
232         this.hourlyRates[selection] = hourlyRate;
233         this.terms[selection] = term;
234     }
235 
236     public void remove() {
237         int selection = Integer.parseInt(JOptionPane.showInputDialog("enter which staff member you want to
    remove:"));
238         // makes sure the input is within range so the array data doesnt wipe to null
239         while (selection > this.currentStaff) {
240             selection = Integer.parseInt(
241                     JOptionPane.showInputDialog("out of bounds error!!\nenter which staff member you want to
    remove:"));
242         } // new temp strings minus 1 length of originals
243         String[] removeNames = new String[this.maxNumberStaff - 1];
244         char[] removeGender = new char[this.maxNumberStaff - 1];
245         int[] removeBirth = new int[this.maxNumberStaff - 1];
246         double[] removeWage = new double[this.maxNumberStaff - 1];
247         Boolean[] removeTerm = new Boolean[this.maxNumberStaff - 1];
248         // new counter initialization
249         int j = 0;
250         int i = 0;
251         // transfers the elemental data from the existing arrays to the temp ones
252         if (selection < this.currentStaff) {
253             while (i < this.currentStaff) {
254                 // skips the selected element and continues counter past it
255                 if (i == (selection - 1)) {
256                     i++;
257                 }
258                 removeNames[j] = this.names[i];
259                 removeGender[j] = this.genders[i];
260                 removeBirth[j] = this.birthYears[i];
261                 removeWage[j] = this.hourlyRates[i];
262                 removeTerm[j] = this.terms[i];
263                 j++;
264                 i++;
265             } // wrote in this section to remove the latest input removes out of bound errors
266                 // perhaps better way i havnt thought of yet
267         } else if (selection == this.currentStaff) {
268             while (i < (this.currentStaff - 1)) {
269                 removeNames[j] = this.names[i];
270                 removeGender[j] = this.genders[i];
271                 removeBirth[j] = this.birthYears[i];
272                 removeWage[j] = this.hourlyRates[i];
273                 removeTerm[j] = this.terms[i];
274                 j++;
275                 i++;
276             }
277         }
278         // creates the original arrays back from the temp ones so the originals now have
279         // a shorter size with all datum still stored from previous minus the removed
280         // element still in order
281         this.names = removeNames;
282         this.genders = removeGender;
283         this.birthYears = removeBirth;
284         this.hourlyRates = removeWage;
285         this.terms = removeTerm;
286         this.maxNumberStaff--;
287         this.currentStaff--;
288     }
289 
290     public void conRefresh() {
291         // refreshes the console version same principal as refreshtable
292         int i = 0;
293         while (i < this.currentStaff) {
294             System.out.println("Staff Member " + (i + 1) + ": " + this.names[i] + " " + this.genders[i] + " "
295                     + this.birthYears[i] + " " + this.hourlyRates[i] + " " + this.terms[i]);
296             i += 1;
297         }
298     }
299 
300     public void refreshTable() {
```

```
301         // add rows to the table using a counter will clear table and add all
302         // inputs in the arrays
303         this.gt.clearRowsOfTable(0);
304         int i = 0;
305         while (i < this.currentStaff) {
306             this.gt.addRowToTable(0, this.names[i] + "\t" + this.genders[i] + "\t" + this.birthYears[i] + "\t"
307                     + this.hourlyRates[i] + "\t" + this.terms[i]);
308             i += 1;
309         }
310     }
311 
312     public static void main(String[] args) {
313         RatesMethodsExtreme prmObj = new RatesMethodsExtreme(
314                 Integer.parseInt(JOptionPane.showInputDialog("enter 0 for console\nenter 1 for gTerm")));
315 
316     }
317 
318 }
319 //bob,m,1980,33,true
```