Final Project

Accounting Program

Authors: D0969611 Chien Chih Lin

D0965719 Chen Hsin Yu

Instructor: Prof Avinash Shankaranarayanan

Table of Contents

1.Introduction to the Program

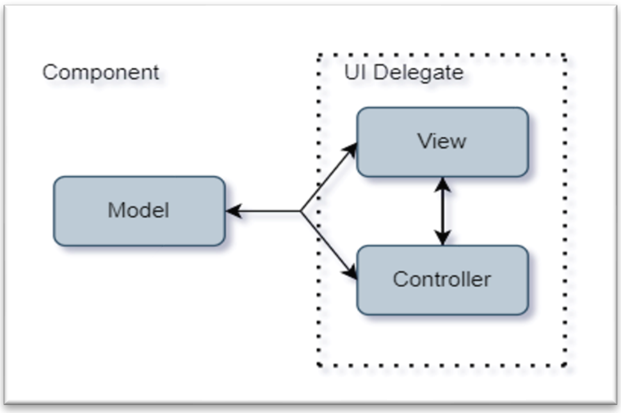
Financial management plays a vital role in our lives. Recording our income and expenditure is a good habit. In this research paper, we focus on the need for an accounting program, which can be used every day and is a very effective method for tracking users' spending. The advantage is fast and timesaving.

We divided this program into three function:
1. **List**, this is used to store the data, users can add the item and choose the classification.
2. **Reminder**, users can set the reminder, if the settlement exceeds the budget, it will remind the users that spending too much.
3. **Graph analysis**, using the data to perform graph analysis by JFreeChart, and making it more visual.

2.Methodology

2.1 Java Swing

Swing is a set of toolkits provided by Java for the development of graphical interface applications and contains various elements for building Graphical User Interfaces (GUI), such as windows, labels, and buttons. It provides many screen display elements that are better than Abstract Window Toolkit (AWT). To distinguish from AWT components, Swing components are under the javax.swing.* package, and the class names all start with J, for example: JFrame, JLabel, JButton.
We choose to use Graphical User Interface (GUI) based on it can design the customized visualization and is easy to view and operate. We design our accounting program by WindowsBuilder from the marketplace.

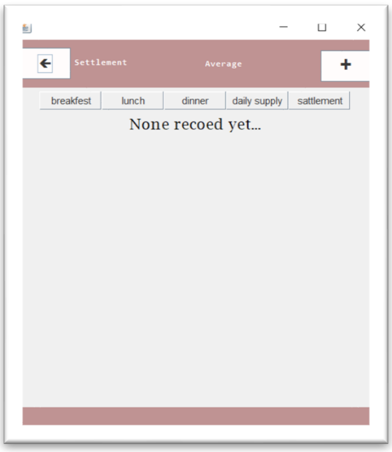


«Java Swing MVC – Model Delegate»

2.2 JTable

The JTable provides a simple mechanism to display large amounts of data. JTable has many things for data generation and editing, many of which can also be customized, such as its layout and size. We make JTable have the following functions:

1. Able to calculate settlement and average
2. Able to judge whether the input is a number
3. Can be arranged from large amount to small amount
4. Data can be saved and exported as charts.



None recoed yet...

«Schematic diagram of JTable»

2.3 JOptionPane

JOptionPane is a mandatory dialog box, you must press the option button to close the dialog box. And we use UIManager to set the appearance. The JOptionPane class has four different dialog boxes:

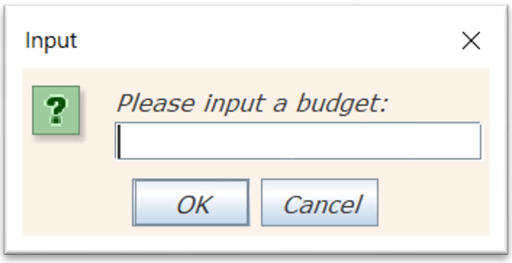
1.ConfirmDialog: Ask the question and user must press the button (Yes/No).

2.InputDialog: Prompt to enter text. 3.MessageDialog: Display information.

4. OptionDialog: Combine the other three dialog types.

We use JOptionPane to do two things:

1. Set reminders, pop out windows and let users enter their budget. 2.Notify, if the settlement succeeds the budget. It will pop up and remind the user.

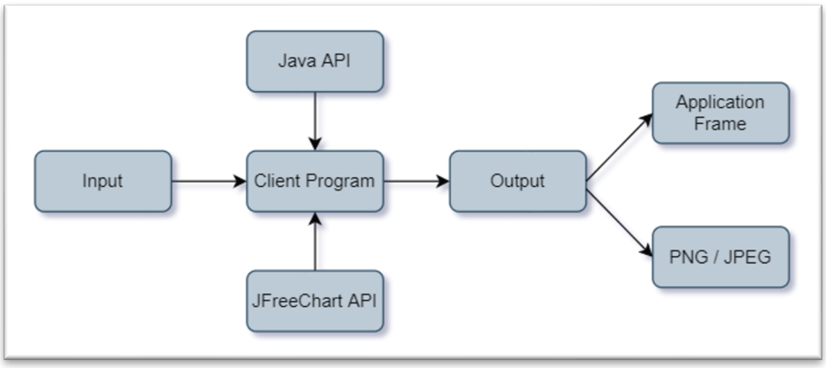


Please input a budget:

«Schematic diagram of JOptionPane»

2.4 JFreeChart

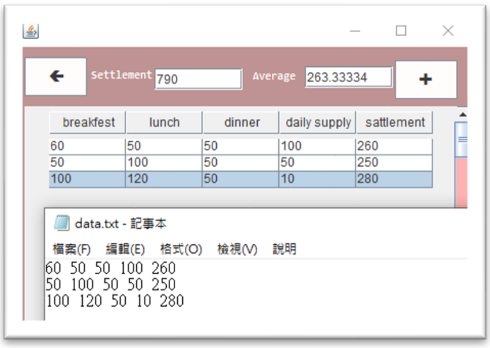
The JFreeChart class is a graphing object, which represents a graphing type. By import external jar file, JFreeChart can generate various charts such as pie charts and bar charts, and it can generate output in PNG and JPEG formats.
In accounting program, we use JFreeChart to illustrate the pie chart given data by JTable and put the chart in JPanel.
1. PieDataset --- Set value in chart
2. ChartFactory--- A collection of utility methods
3. StandardChartTheme --- Set font


$$\begin{tikzcd}
 & \text{Java API} \arrow[d] & & \text{Application Frame} \\
\text{Input} \arrow[r] & \text{Client Program} \arrow[r] & \text{Output} \arrow[ur] \arrow[dr] & \\
 & \text{JFreeChart API} \arrow[u] & & \text{PNG / JPEG}
\end{tikzcd}$$

«JFreeChart library inside the Java application»

2.5 Storing data

Java provides multiple APIs to read a text file. Java calls the stream object created by the Reader abstract subclass as the input stream (FileReader) and calls the stream object created by the Writer abstract subclass as the output stream (FileWriter). In this program, we use the FileWriter and FileReader to read and write the data, such as output forms, types, and the settlement to set reminder and generate pie charts for graph analysis. And in the graph analysis part, we must convert numbers to percentages so that it can generate the pie chart.



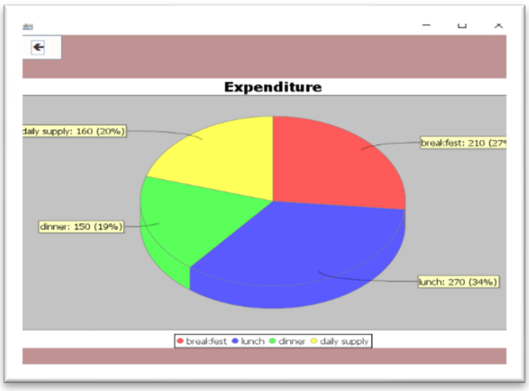
Settlement 790 Average 263.33334

breakfest	lunch	dinner	daily supply	sattlement
60	50	50	100	260
50	100	50	50	250
100	120	50	10	280

data.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

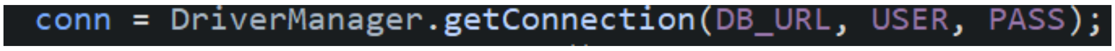
```
60  50  50  100  260
50  100  50  50  250
100  120  50  10  280
```



2.6 MySQL

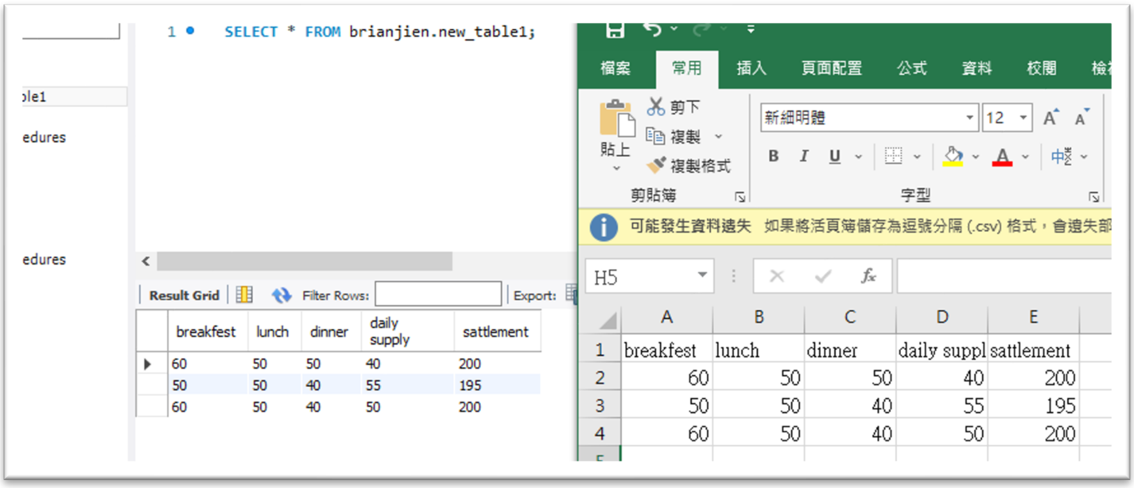
MySQL is a server database that can store many types of data online. Is a fully managed database service to deploy cloud-native applications.

we export our data into csv. File and import table let MYSQL can read the data. Connect to JDBC server by using USERNAME , PASSWORD and URL



```
conn = DriverManager.getConnection(DB_URL, USER, PASS);
```

```
data666.csv          2021/6/6 下午 06:18     Microsoft Excel ...     1 KB
```



```
1 • SELECT * FROM brianjien.new_table1;
```

可能發生資料遺失 如果將活頁簿儲存為逗號分隔 (.csv) 格式，會遺失部

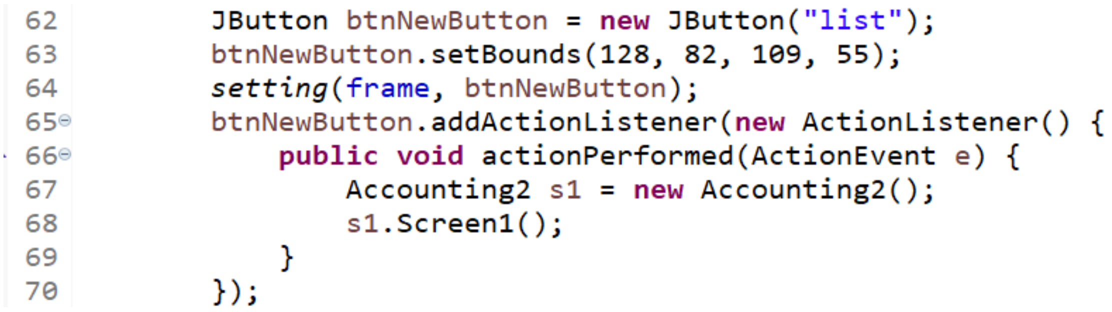
	breakfest	lunch	dinner	daily supply	sattlement
▶	60	50	50	40	200
	50	50	40	55	195
	60	50	40	50	200

	A	B	C	D	E
1	breakfest	lunch	dinner	daily suppl	sattlement
2	60	50	50	40	200
3	50	50	40	55	195
4	60	50	40	50	200

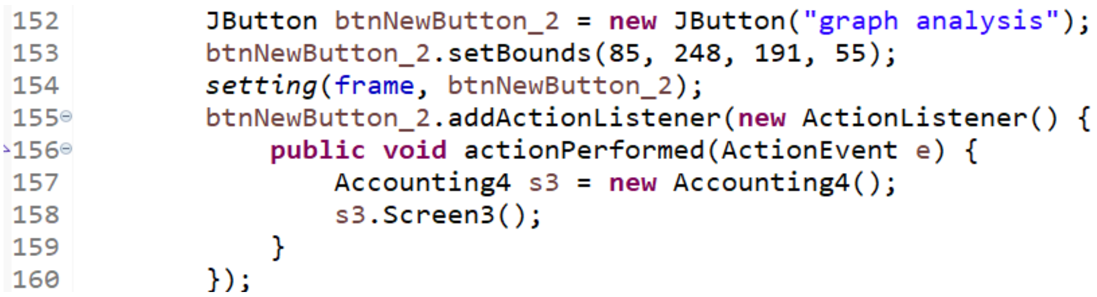
2.7 Code Analysis

Accounting.java

Line 62~70, 152~160: Create the button, set its bounds, call the setting function, and add the ActionListener. If press these two buttons, it will pop up another window.

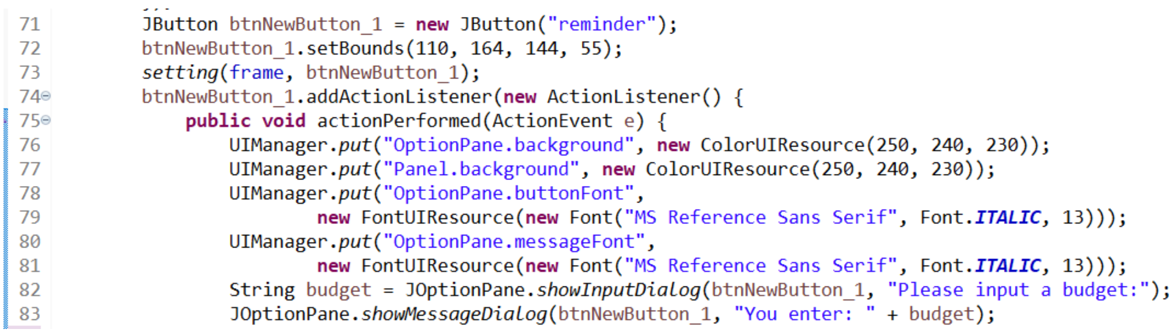


```
62          JButton btnNewButton = new JButton("list");
63          btnNewButton.setBounds(128, 82, 109, 55);
64          setting(frame, btnNewButton);
65          btnNewButton.addActionListener(new ActionListener() {
66              public void actionPerformed(ActionEvent e) {
67                  Accounting2 s1 = new Accounting2();
68                  s1.Screen1();
69              }
70          });
```



```
152         JButton btnNewButton_2 = new JButton("graph analysis");
153         btnNewButton_2.setBounds(85, 248, 191, 55);
154         setting(frame, btnNewButton_2);
155         btnNewButton_2.addActionListener(new ActionListener() {
156             public void actionPerformed(ActionEvent e) {
157                 Accounting4 s3 = new Accounting4();
158                 s3.Screen3();
159             }
160         });
```

Line 71~83: Set the reminder button, we use the JOptionPane to enter the budget, show the budget and notify if settlement exceeds the budget. Then, Use UIManager to set the appearance of JOptionPane.



```
71          JButton btnNewButton_1 = new JButton("reminder");
72          btnNewButton_1.setBounds(110, 164, 144, 55);
73          setting(frame, btnNewButton_1);
74          btnNewButton_1.addActionListener(new ActionListener() {
75              public void actionPerformed(ActionEvent e) {
76                  UIManager.put("OptionPane.background", new ColorUIResource(250, 240, 230));
77                  UIManager.put("Panel.background", new ColorUIResource(250, 240, 230));
78                  UIManager.put("OptionPane.buttonFont",
79                          new FontUIResource(new Font("MS Reference Sans Serif", Font.ITALIC, 13)));
80                  UIManager.put("OptionPane.messageFont",
81                          new FontUIResource(new Font("MS Reference Sans Serif", Font.ITALIC, 13)));
82                  String budget = JOptionPane.showInputDialog(btnNewButton_1, "Please input a budget:");
83                  JOptionPane.showMessageDialog(btnNewButton_1, "You enter: " + budget);
```

Line 162~166: Use JLabel to present the homepage.

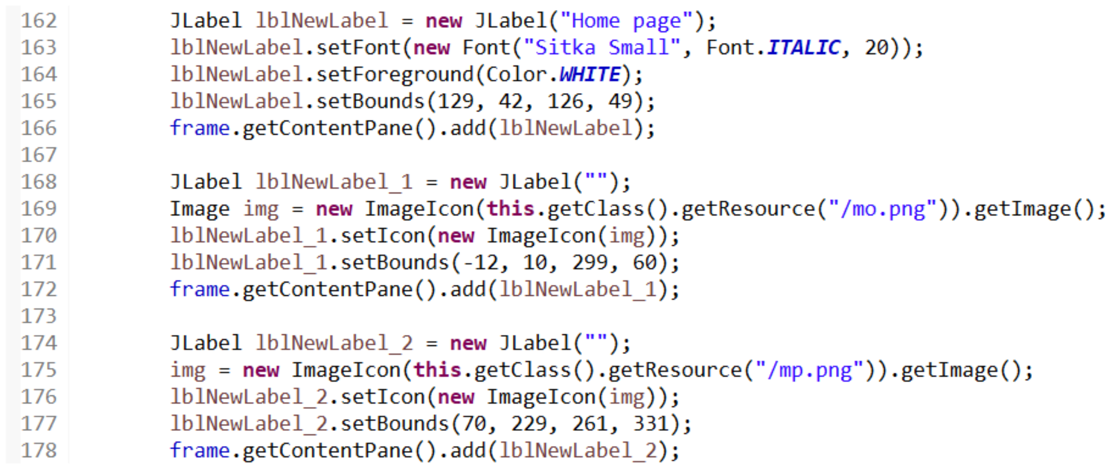
Line 168~178: Use JLabel to put the image and set its bounds, add the JLabel into frame.

Line 169,175:

getClass(): Return a Class object corresponding to your object, this return object holds the class information of your original object.

getResource(): Return the resources of the module in which this class exists.

getImage(): Return an image that gets pixel data from the specified file.



```
162         JLabel lblNewLabel = new JLabel("Home page");
163         lblNewLabel.setFont(new Font("Sitka Small", Font.ITALIC, 20));
164         lblNewLabel.setForeground(Color.WHITE);
165         lblNewLabel.setBounds(129, 42, 126, 49);
166         frame.getContentPane().add(lblNewLabel);
167
168         JLabel lblNewLabel_1 = new JLabel("");
169         Image img = new ImageIcon(this.getClass().getResource("/mo.png")).getImage();
170         lblNewLabel_1.setIcon(new ImageIcon(img));
171         lblNewLabel_1.setBounds(-12, 10, 299, 60);
172         frame.getContentPane().add(lblNewLabel_1);
173
174         JLabel lblNewLabel_2 = new JLabel("");
175         img = new ImageIcon(this.getClass().getResource("/mp.png")).getImage();
176         lblNewLabel_2.setIcon(new ImageIcon(img));
177         lblNewLabel_2.setBounds(70, 229, 261, 331);
178         frame.getContentPane().add(lblNewLabel_2);
```

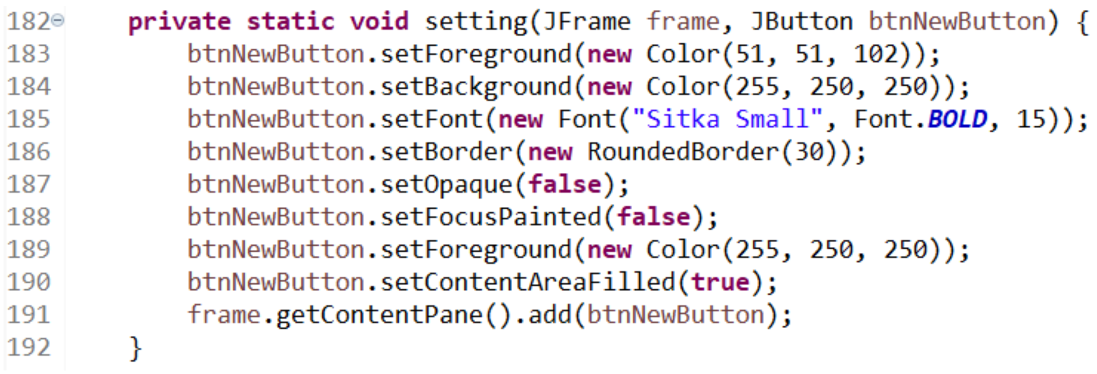
Line 182~192:

setting(): We set up our homepage's background to pink, Color stands for Red, Green, Blue, and add the button to the frame.

setOpaque(): We set this method false, so the button may not paint some or all of its pixel, allowing the underlying pixels to show through.

setFocusPainted(): This is to set whether to draw the focus. For example, a light-colored dashed frame or a bold frame indicates that the button currently has focus.

setContentAreaFilled(): This is to set whether to fill or not, we set it to false, so this button look transparent.

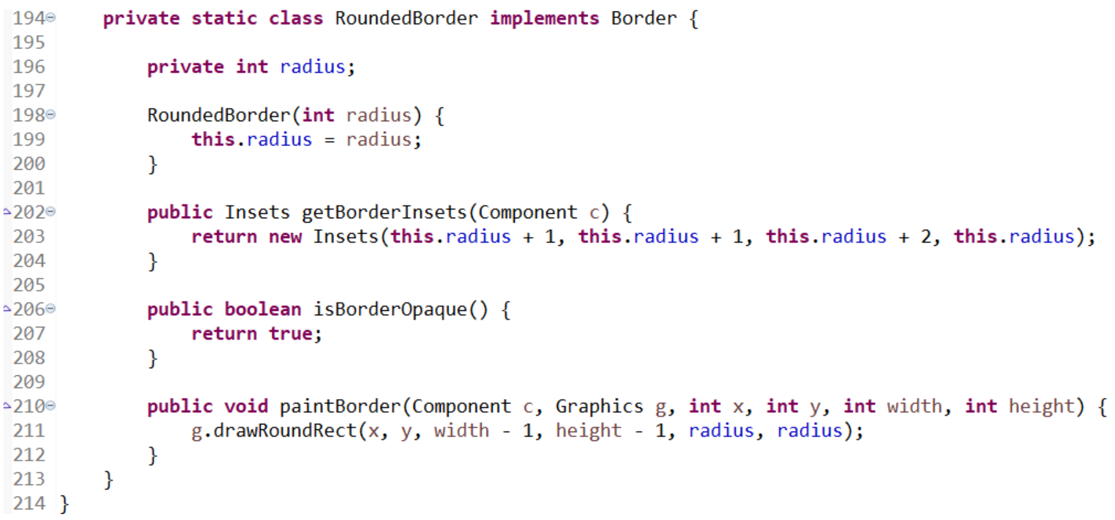


```
182     private static void setting(JFrame frame, JButton btnNewButton) {
183         btnNewButton.setForeground(new Color(51, 51, 102));
184         btnNewButton.setBackground(new Color(255, 250, 250));
185         btnNewButton.setFont(new Font("Sitka Small", Font.BOLD, 15));
186         btnNewButton.setBorder(new RoundedBorder(30));
187         btnNewButton.setOpaque(false);
188         btnNewButton.setFocusPainted(false);
189         btnNewButton.setForeground(new Color(255, 250, 250));
190         btnNewButton.setContentAreaFilled(true);
191         frame.getContentPane().add(btnNewButton);
192     }
```

Line 194~214: Class RoundedBorder, we change the radius by extending the border, creates and initializes a new insets object with the specified top, left, bottom, and right insets.

A JComponent is a void Bounded Box that can be added into swing containers, and Graphics is a package includes how to draw lines and shapes, draw text and images and fill shapes.

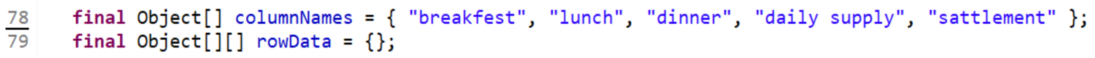
We use this class to make the border of button round.



```
194     private static class RoundedBorder implements Border {
195 
196         private int radius;
197 
198         RoundedBorder(int radius) {
199             this.radius = radius;
200         }
201 
202         public Insets getBorderInsets(Component c) {
203             return new Insets(this.radius + 1, this.radius + 1, this.radius + 2, this.radius);
204         }
205 
206         public boolean isBorderOpaque() {
207             return true;
208         }
209 
210         public void paintBorder(Component c, Graphics g, int x, int y, int width, int height) {
211             g.drawRoundRect(x, y, width - 1, height - 1, radius, radius);
212         }
213     }
214 }
```

Accounting2.java

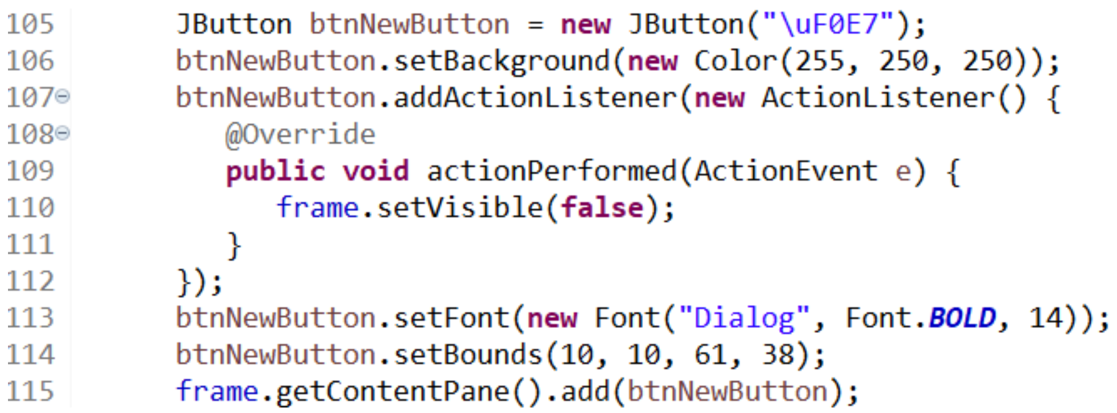
Line 78~79: Create a column and empty row data and type in the classification.



```java
78    final Object[] columnNames = { "breakfest", "lunch", "dinner", "daily supply", "sattlement" };
79    final Object[][] rowData = {};
```

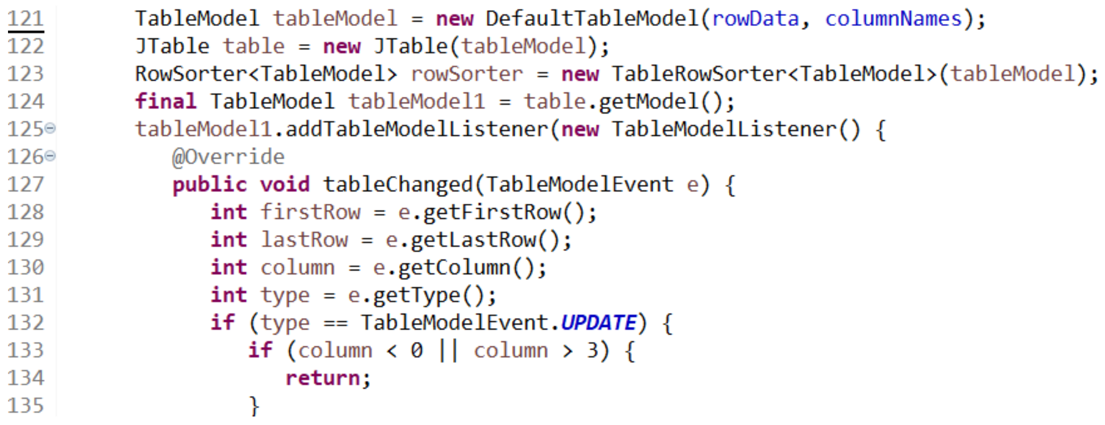
Line 105~115: If press this button, this frame will be hide.

setVisible() is to set the visibility of components. It is used to go back to the last page.



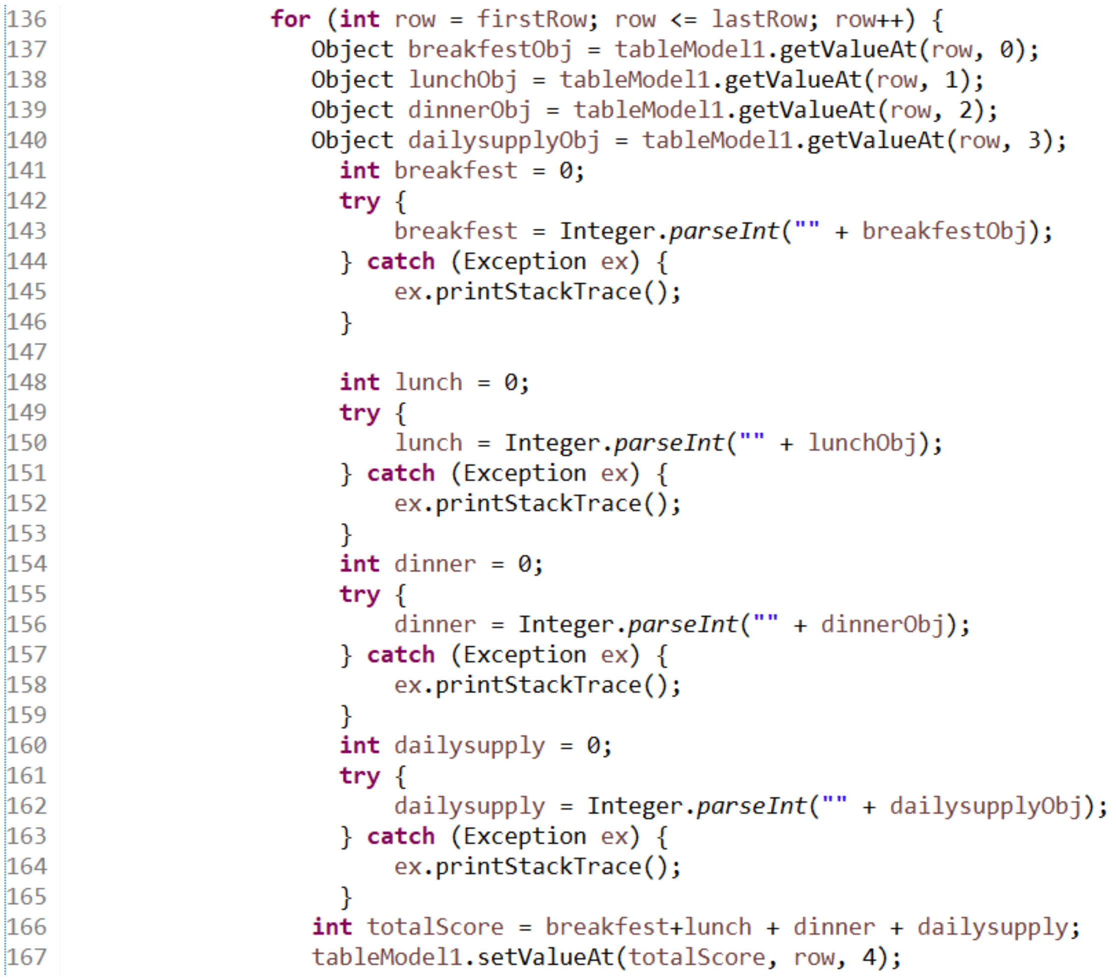
```java
105        JButton btnNewButton = new JButton("\uF0E7");
106        btnNewButton.setBackground(new Color(255, 250, 250));
107        btnNewButton.addActionListener(new ActionListener() {
108           @Override
109           public void actionPerformed(ActionEvent e) {
110              frame.setVisible(false);
111           }
112        });
113        btnNewButton.setFont(new Font("Dialog", Font.BOLD, 14));
114        btnNewButton.setBounds(10, 10, 61, 38);
115        frame.getContentPane().add(btnNewButton);
```

Line 121~135: Create a table with data and column. Use TableModelEvent() to get first row to furth row's data., and update data demonically.



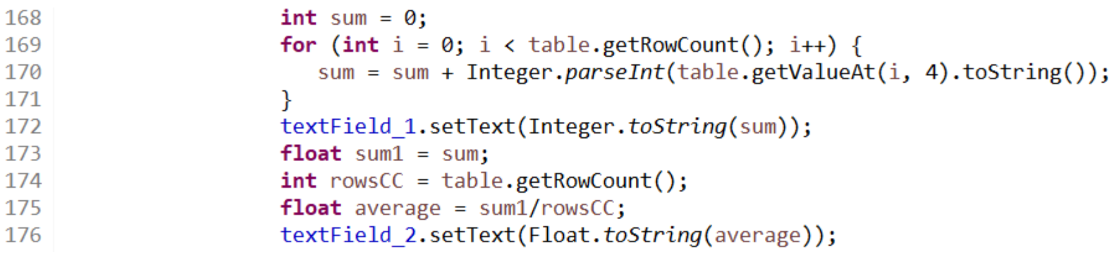
```java
121        TableModel tableModel = new DefaultTableModel(rowData, columnNames);
122        JTable table = new JTable(tableModel);
123        RowSorter<TableModel> rowSorter = new TableRowSorter<TableModel>(tableModel);
124        final TableModel tableModel1 = table.getModel();
125        tableModel1.addTableModelListener(new TableModelListener() {
126           @Override
127           public void tableChanged(TableModelEvent e) {
128              int firstRow = e.getFirstRow();
129              int lastRow = e.getLastRow();
130              int column = e.getColumn();
131              int type = e.getType();
132              if (type == TableModelEvent.UPDATE) {
133                 if (column < 0 || column > 3) {
134                    return;
135                 }
```

Line 136~167: Set up the object of columns and get value at each row by using loop. Change the value from object to integer. Use try and catch to catch the errors. Add the value of four columns value and set the value to row 4(that means in position 5)



```
136         for (int row = firstRow; row <= lastRow; row++) {
137            Object breakfestObj = tableModel1.getValueAt(row, 0);
138            Object lunchObj = tableModel1.getValueAt(row, 1);
139            Object dinnerObj = tableModel1.getValueAt(row, 2);
140            Object dailysupplyObj = tableModel1.getValueAt(row, 3);
141              int breakfest = 0;
142              try {
143                  breakfest = Integer.parseInt("" + breakfestObj);
144              } catch (Exception ex) {
145                  ex.printStackTrace();
146              }
147
148              int lunch = 0;
149              try {
150                  lunch = Integer.parseInt("" + lunchObj);
151              } catch (Exception ex) {
152                  ex.printStackTrace();
153              }
154              int dinner = 0;
155              try {
156                  dinner = Integer.parseInt("" + dinnerObj);
157              } catch (Exception ex) {
158                  ex.printStackTrace();
159              }
160              int dailysupply = 0;
161              try {
162                  dailysupply = Integer.parseInt("" + dailysupplyObj);
163              } catch (Exception ex) {
164                  ex.printStackTrace();
165              }
166            int totalScore = breakfest+lunch + dinner + dailysupply;
167            tableModel1.setValueAt(totalScore, row, 4);
```

Line 168~176: Sum up the all the value of row 4(that means in position 5) and calculate the average.



```
168         int sum = 0;
169         for (int i = 0; i < table.getRowCount(); i++) {
170            sum = sum + Integer.parseInt(table.getValueAt(i, 4).toString());
171         }
172         textField_1.setText(Integer.toString(sum));
173         float sum1 = sum;
174         int rowsCC = table.getRowCount();
175         float average = sum1/rowsCC;
176         textField_2.setText(Float.toString(average));
```

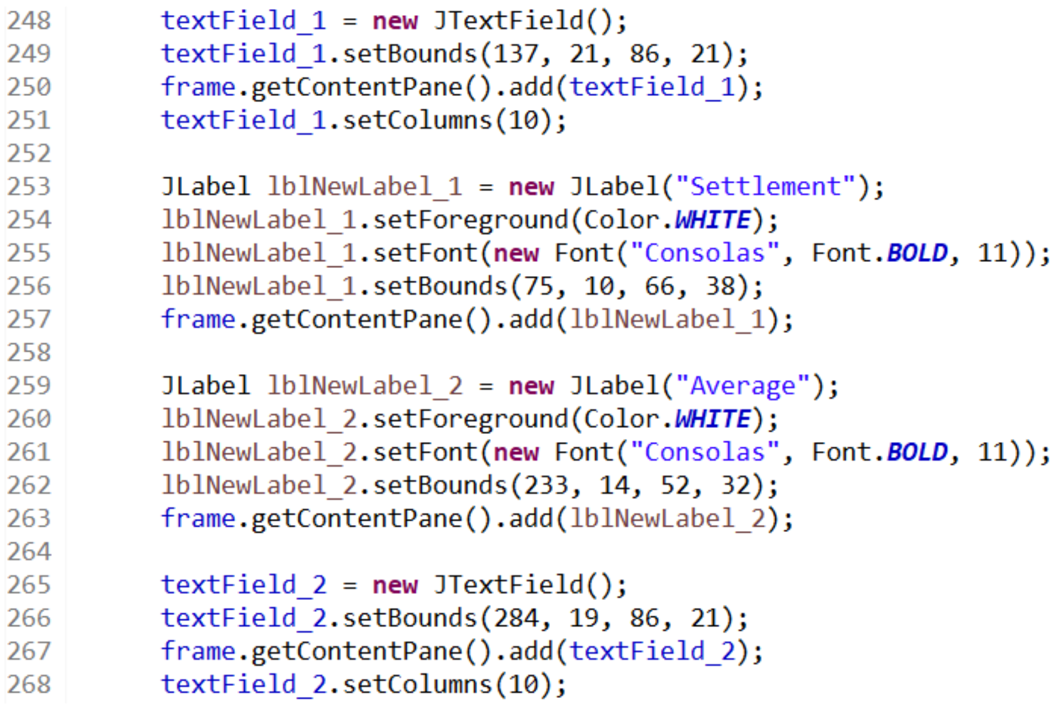
Line 178~208: Get the sum of each column and export the text file with the FileWriter (for the pie chart).

```
178     int sumbr = 0;
179     for (int i = 0; i < table.getRowCount(); i++) {
180        sumbr = sumbr + Integer.parseInt(table.getValueAt(i, 0).toString());
181     }
182     int sumlun = 0;
183     for (int i = 0; i < table.getRowCount(); i++) {
184         sumlun = sumlun + Integer.parseInt(table.getValueAt(i, 1).toString());
185     }
186     int sumdin = 0;
187     for (int i = 0; i < table.getRowCount(); i++) {
188         sumdin = sumdin + Integer.parseInt(table.getValueAt(i, 2).toString());
189     }
190     int sumsu = 0;
191     for (int i = 0; i < table.getRowCount(); i++) {
192         sumsu = sumsu + Integer.parseInt(table.getValueAt(i, 3).toString());
193     }
194
195     String filePath1 = "C:\\Users\\USER\\Downloads\\code\\hello\\data1.txt";
196     File file1 = new File(filePath1);
197     try {
198         FileWriter fw = new FileWriter(file1);
199         BufferedWriter bw = new BufferedWriter(fw);
200
201        bw.write(sumbr+"\n"+sumlun+"\n"+sumdin+"\n"+ +sumsu+"\n"+sum);
202
203         bw.close();
204         fw.close();
205
206     } catch (IOException ex) {
207
208     }
```

Line 230~246: Press plus bottom to add row in the JTable. When add a row in the JTable the" None record yet..." will disappear.

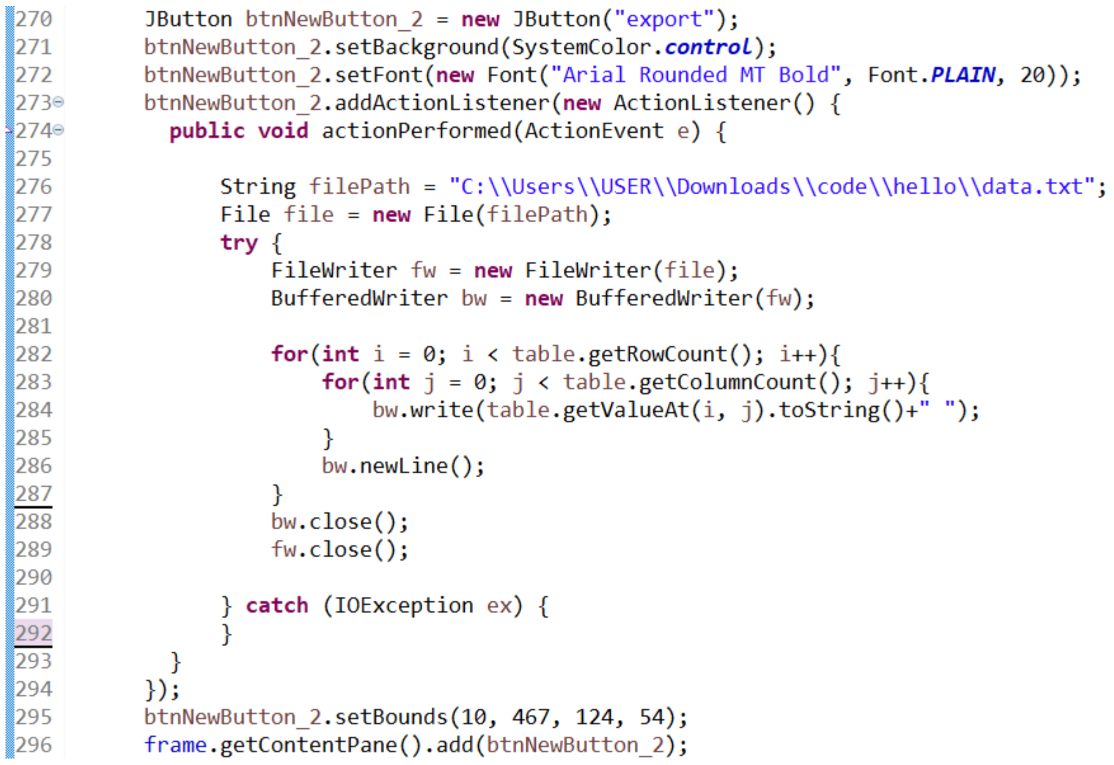
```
230             JLabel lblNewLabel = new JLabel("None recoed yet...");
231             panel.add(lblNewLabel);
232             lblNewLabel.setFont(new Font("Lucida Bright", Font.PLAIN, 18));
233             lblNewLabel.setForeground(Color.BLACK);
234         panel.add(table, BorderLayout.CENTER);
235         JButton btnNewButton_1 = new JButton("+");
236         btnNewButton_1.addActionListener(new ActionListener() {
237             @Override
238             public void actionPerformed(ActionEvent e) {
239                 ((DefaultTableModel) tableModel1).addRow(new Object[] { "" });
240                 lblNewLabel.setVisible(false);
241             }
242         });
243         btnNewButton_1.setFont(new Font("Dialog", Font.BOLD, 24));
244         btnNewButton_1.setBackground(new Color(255, 250, 250));
245         btnNewButton_1.setBounds(372, 13, 61, 38);
246         frame.getContentPane().add(btnNewButton_1);
```

Line 248~268: Use sum and average in line 172, 176 and appear in JTextField.



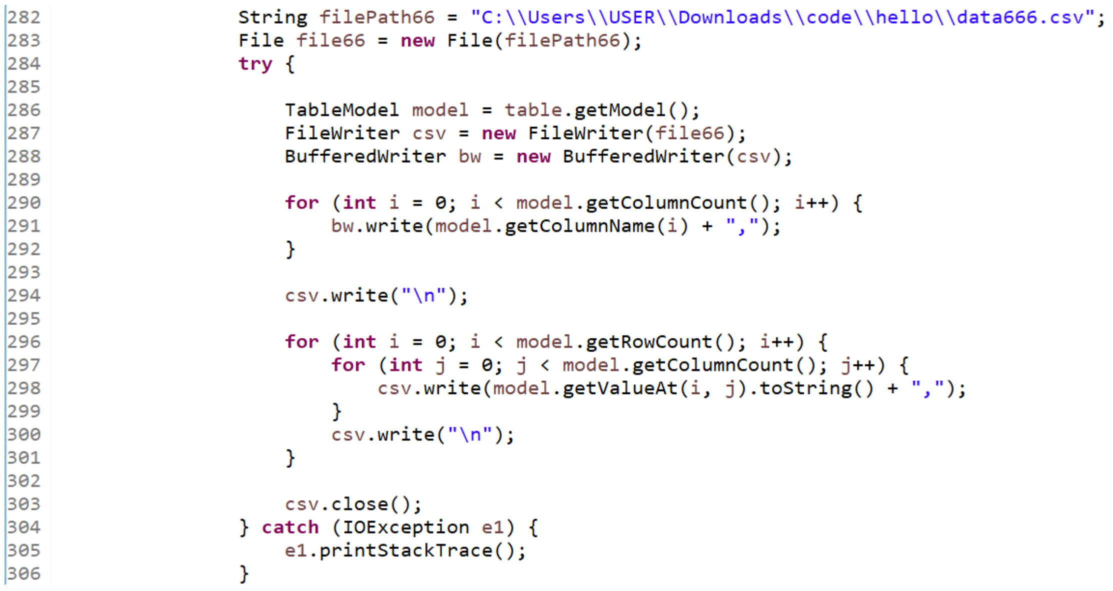
```
248         textField_1 = new JTextField();
249         textField_1.setBounds(137, 21, 86, 21);
250         frame.getContentPane().add(textField_1);
251         textField_1.setColumns(10);
252 
253         JLabel lblNewLabel_1 = new JLabel("Settlement");
254         lblNewLabel_1.setForeground(Color.WHITE);
255         lblNewLabel_1.setFont(new Font("Consolas", Font.BOLD, 11));
256         lblNewLabel_1.setBounds(75, 10, 66, 38);
257         frame.getContentPane().add(lblNewLabel_1);
258 
259         JLabel lblNewLabel_2 = new JLabel("Average");
260         lblNewLabel_2.setForeground(Color.WHITE);
261         lblNewLabel_2.setFont(new Font("Consolas", Font.BOLD, 11));
262         lblNewLabel_2.setBounds(233, 14, 52, 32);
263         frame.getContentPane().add(lblNewLabel_2);
264 
265         textField_2 = new JTextField();
266         textField_2.setBounds(284, 19, 86, 21);
267         frame.getContentPane().add(textField_2);
268         textField_2.setColumns(10);
```

Line 270~296: Use export button to export text file . Using loop to write all the value in text fileby using Filewriter and BufferedWriter.



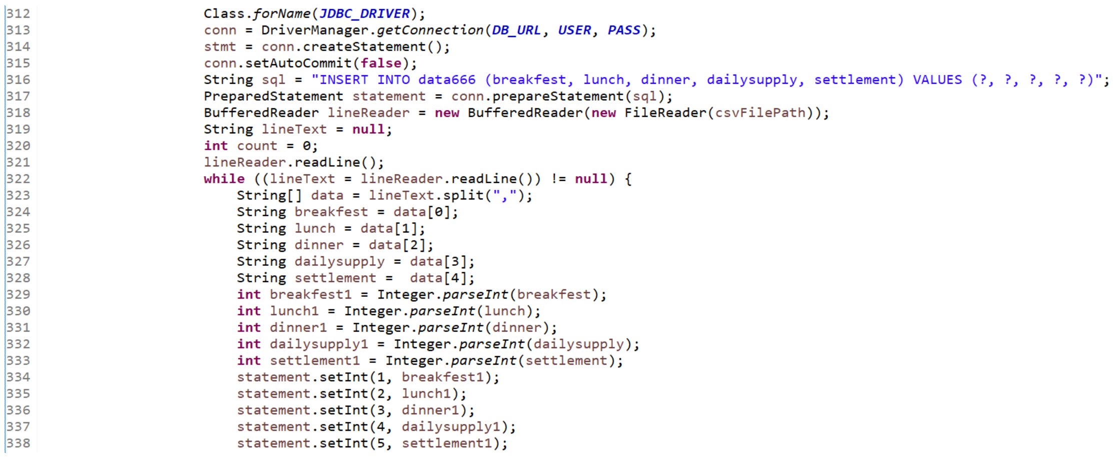
```
270         JButton btnNewButton_2 = new JButton("export");
271         btnNewButton_2.setBackground(SystemColor.control);
272         btnNewButton_2.setFont(new Font("Arial Rounded MT Bold", Font.PLAIN, 20));
273         btnNewButton_2.addActionListener(new ActionListener() {
274             public void actionPerformed(ActionEvent e) {
275 
276                 String filePath = "C:\\Users\\USER\\Downloads\\code\\hello\\data.txt";
277                 File file = new File(filePath);
278                 try {
279                     FileWriter fw = new FileWriter(file);
280                     BufferedWriter bw = new BufferedWriter(fw);
281 
282                     for(int i = 0; i < table.getRowCount(); i++){
283                         for(int j = 0; j < table.getColumnCount(); j++){
284                             bw.write(table.getValueAt(i, j).toString()+" ");
285                         }
286                         bw.newLine();
287                     }
288                     bw.close();
289                     fw.close();
290 
291                 } catch (IOException ex) {
292                 }
293             }
294         });
295         btnNewButton_2.setBounds(10, 467, 124, 54);
296         frame.getContentPane().add(btnNewButton_2);
```

Line 282~386: we export our JTable to csv.Add comma next to the data and can be readable by excel and MySQL



```
282                 String filePath66 = "C:\\Users\\USER\\Downloads\\code\\hello\\data666.csv";
283                 File file66 = new File(filePath66);
284                 try {
285 
286                     TableModel model = table.getModel();
287                     FileWriter csv = new FileWriter(file66);
288                     BufferedWriter bw = new BufferedWriter(csv);
289 
290                     for (int i = 0; i < model.getColumnCount(); i++) {
291                         bw.write(model.getColumnName(i) + ",");
292                     }
293 
294                     csv.write("\n");
295 
296                     for (int i = 0; i < model.getRowCount(); i++) {
297                         for (int j = 0; j < model.getColumnCount(); j++) {
298                             csv.write(model.getValueAt(i, j).toString() + ",");
299                         }
300                         csv.write("\n");
301                     }
302 
303                     csv.close();
304                 } catch (IOException e1) {
305                     e1.printStackTrace();
306                 }
```

Line 312~338:Use Diver Manger to get the connection and setting MySQL. Storing data into the database by fit into the right position, Change data type to Interger.



```
312                 Class.forName(JDBC_DRIVER);
313                 conn = DriverManager.getConnection(DB_URL, USER, PASS);
314                 stmt = conn.createStatement();
315                 conn.setAutoCommit(false);
316                 String sql = "INSERT INTO data666 (breakfest, lunch, dinner, dailysupply, settlement) VALUES (?, ?, ?, ?, ?)";
317                 PreparedStatement statement = conn.prepareStatement(sql);
318                 BufferedReader lineReader = new BufferedReader(new FileReader(csvFilePath));
319                 String lineText = null;
320                 int count = 0;
321                 lineReader.readLine();
322                 while ((lineText = lineReader.readLine()) != null) {
323                     String[] data = lineText.split(",");
324                     String breakfest = data[0];
325                     String lunch = data[1];
326                     String dinner = data[2];
327                     String dailysupply = data[3];
328                     String settlement =  data[4];
329                     int breakfest1 = Integer.parseInt(breakfest);
330                     int lunch1 = Integer.parseInt(lunch);
331                     int dinner1 = Integer.parseInt(dinner);
332                     int dailysupply1 = Integer.parseInt(dailysupply);
333                     int settlement1 = Integer.parseInt(settlement);
334                     statement.setInt(1, breakfest1);
335                     statement.setInt(2, lunch1);
336                     statement.setInt(3, dinner1);
337                     statement.setInt(4, dailysupply1);
338                     statement.setInt(5, settlement1);
```

Line 298~325:Use import button to import txt file. When we import text file ,we have to remove the row we entered. So that the value will not be repeat.

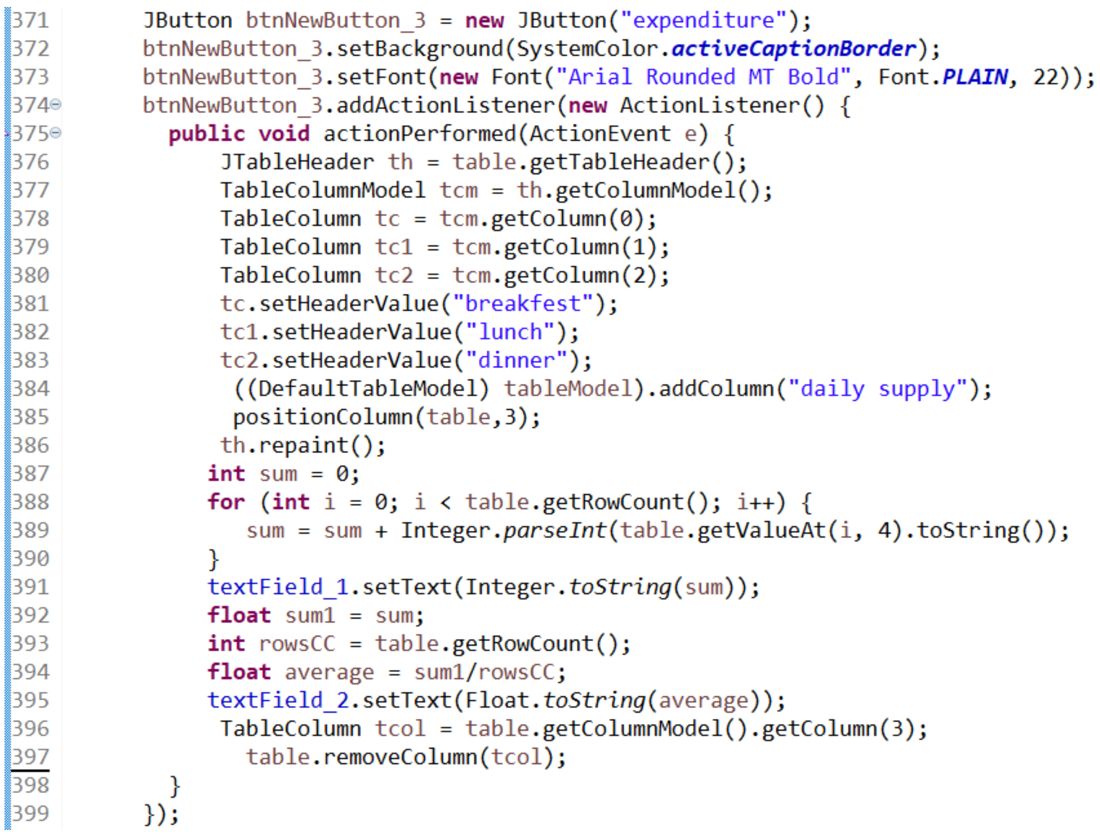
Line 326~330:when we import our data ,we still have to calculate the sum and average in Line:172,176 and appear in textfeild.

```
298        JButton btnNewButton_2_1 = new JButton("import");
299        btnNewButton_2_1.setBackground(SystemColor.control);
300        btnNewButton_2_1.setFont(new Font("Arial Rounded MT Bold", Font.PLAIN, 20));
301        btnNewButton_2_1.addActionListener(new ActionListener() {
302          public void actionPerformed(ActionEvent e) {
303              lblNewLabel.setVisible(false);
304              DefaultTableModel dm = (DefaultTableModel)table.getModel();
305              int rowCount = dm.getRowCount();
306              //Remove rows one by one from the end of the table
307              for (int i = rowCount - 1; i >= 0; i--) {
308                  dm.removeRow(i);
309              }
310              String filePath = "C:\\Users\\USER\\Downloads\\code\\hello\\data.txt";
311              File file = new File(filePath);
312
313              try {
314                  FileReader fr = new FileReader(file);
315                  BufferedReader br = new BufferedReader(fr);
316
317                  DefaultTableModel model = (DefaultTableModel)table.getModel();
318                  Object[] lines = br.lines().toArray();
319
320                  for(int i = 0; i < lines.length; i++){
321                      String[] row = lines[i].toString().split(" ");
322                      model.addRow(row);
323                  }
324              } catch (FileNotFoundException ex) {
325              }
326              int sum = 0;
327              for (int i = 0; i < table.getRowCount(); i++) {
328                 sum = sum + Integer.parseInt(table.getValueAt(i, 4).toString());
329              }
330              textField_1.setText(Integer.toString(sum));
```

Line:340~350 Press delete button to delete all the rows in JTable.Using getRowcount() to get all the row .Use remove rows the delete all the rows in Jtable.

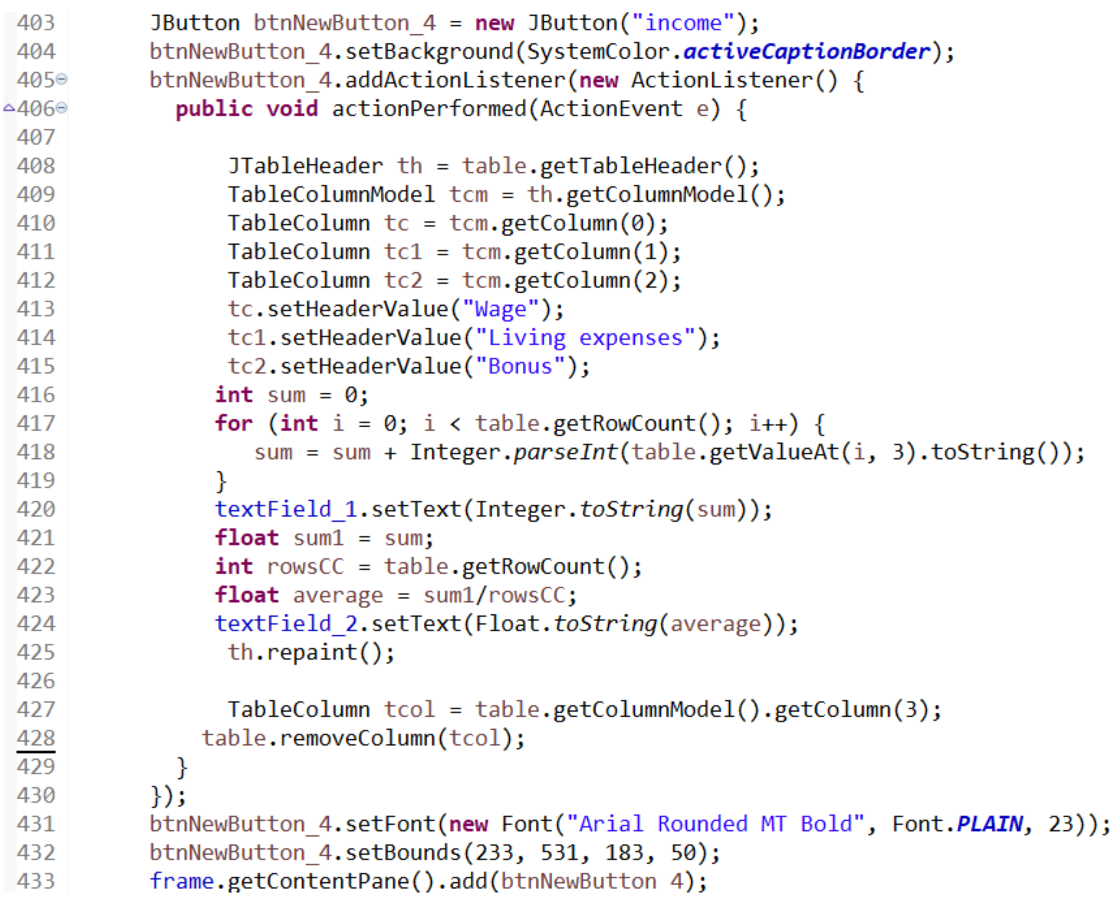
```
340        JButton btnNewButton_2_1_1 = new JButton("delete");
341        btnNewButton_2_1_1.setBackground(SystemColor.control);
342        btnNewButton_2_1_1.setFont(new Font("Arial Rounded MT Bold", Font.PLAIN, 20));
343        btnNewButton_2_1_1.addActionListener(new ActionListener() {
344          public void actionPerformed(ActionEvent e) {
345              DefaultTableModel dm = (DefaultTableModel)table.getModel();
346              int rowCount = dm.getRowCount();
347              //Remove rows one by one from the end of the table
348              for (int i = rowCount - 1; i >= 0; i--) {
349                  dm.removeRow(i);
350              }
351              int sum = 0;
352              for (int i = 0; i < table.getRowCount(); i++) {
353                 sum = sum + Integer.parseInt(table.getValueAt(i, 4).toString());
354              }
355              textField_1.setText(Integer.toString(sum));
356              float sum1 = sum;
357              int rowsCC = table.getRowCount();
358              float average = sum1/rowsCC;
359              textField_2.setText(Float.toString(average));
360          }
361        });
```

Line:371~390: Change column to expenditure. Use getTableHeader() to change column .



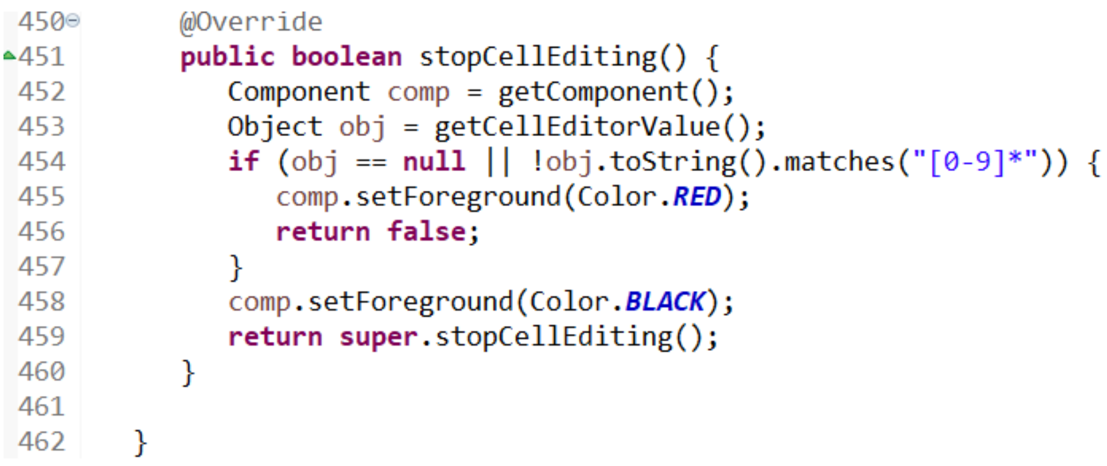
```
371        JButton btnNewButton_3 = new JButton("expenditure");
372        btnNewButton_3.setBackground(SystemColor.activeCaptionBorder);
373        btnNewButton_3.setFont(new Font("Arial Rounded MT Bold", Font.PLAIN, 22));
374        btnNewButton_3.addActionListener(new ActionListener() {
375          public void actionPerformed(ActionEvent e) {
376              JTableHeader th = table.getTableHeader();
377              TableColumnModel tcm = th.getColumnModel();
378              TableColumn tc = tcm.getColumn(0);
379              TableColumn tc1 = tcm.getColumn(1);
380              TableColumn tc2 = tcm.getColumn(2);
381              tc.setHeaderValue("breakfest");
382              tc1.setHeaderValue("lunch");
383              tc2.setHeaderValue("dinner");
384               ((DefaultTableModel) tableModel).addColumn("daily supply");
385               positionColumn(table,3);
386              th.repaint();
387             int sum = 0;
388             for (int i = 0; i < table.getRowCount(); i++) {
389                sum = sum + Integer.parseInt(table.getValueAt(i, 4).toString());
390             }
391             textField_1.setText(Integer.toString(sum));
392             float sum1 = sum;
393             int rowsCC = table.getRowCount();
394             float average = sum1/rowsCC;
395             textField_2.setText(Float.toString(average));
396              TableColumn tcol = table.getColumnModel().getColumn(3);
397                table.removeColumn(tcol);
398          }
399        });
```

Line:403~433:same as Line:371~350 change the column to income(wage,living expenses,bonus)



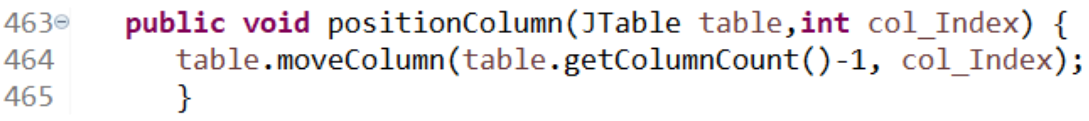
```
403        JButton btnNewButton_4 = new JButton("income");
404        btnNewButton_4.setBackground(SystemColor.activeCaptionBorder);
405        btnNewButton_4.addActionListener(new ActionListener() {
406          public void actionPerformed(ActionEvent e) {
407
408              JTableHeader th = table.getTableHeader();
409              TableColumnModel tcm = th.getColumnModel();
410              TableColumn tc = tcm.getColumn(0);
411              TableColumn tc1 = tcm.getColumn(1);
412              TableColumn tc2 = tcm.getColumn(2);
413              tc.setHeaderValue("Wage");
414              tc1.setHeaderValue("Living expenses");
415              tc2.setHeaderValue("Bonus");
416             int sum = 0;
417             for (int i = 0; i < table.getRowCount(); i++) {
418                sum = sum + Integer.parseInt(table.getValueAt(i, 3).toString());
419             }
420             textField_1.setText(Integer.toString(sum));
421             float sum1 = sum;
422             int rowsCC = table.getRowCount();
423             float average = sum1/rowsCC;
424             textField_2.setText(Float.toString(average));
425              th.repaint();
426
427              TableColumn tcol = table.getColumnModel().getColumn(3);
428            table.removeColumn(tcol);
429          }
430        });
431        btnNewButton_4.setFont(new Font("Arial Rounded MT Bold", Font.PLAIN, 23));
432        btnNewButton_4.setBounds(233, 531, 183, 50);
433        frame.getContentPane().add(btnNewButton_4);
```

Line450~462: Use stopcellEditing() to get the component of JTable,and if the value in the table do not match 0~9.The Foreground will be RED, if matches. The color will become BLACK.



```
450      @Override
451      public boolean stopCellEditing() {
452          Component comp = getComponent();
453          Object obj = getCellEditorValue();
454          if (obj == null || !obj.toString().matches("[0-9]*")) {
455              comp.setForeground(Color.RED);
456              return false;
457          }
458          comp.setForeground(Color.BLACK);
459          return super.stopCellEditing();
460      }
461
462  }
```

Line463~465:Use function position to add column in direct position



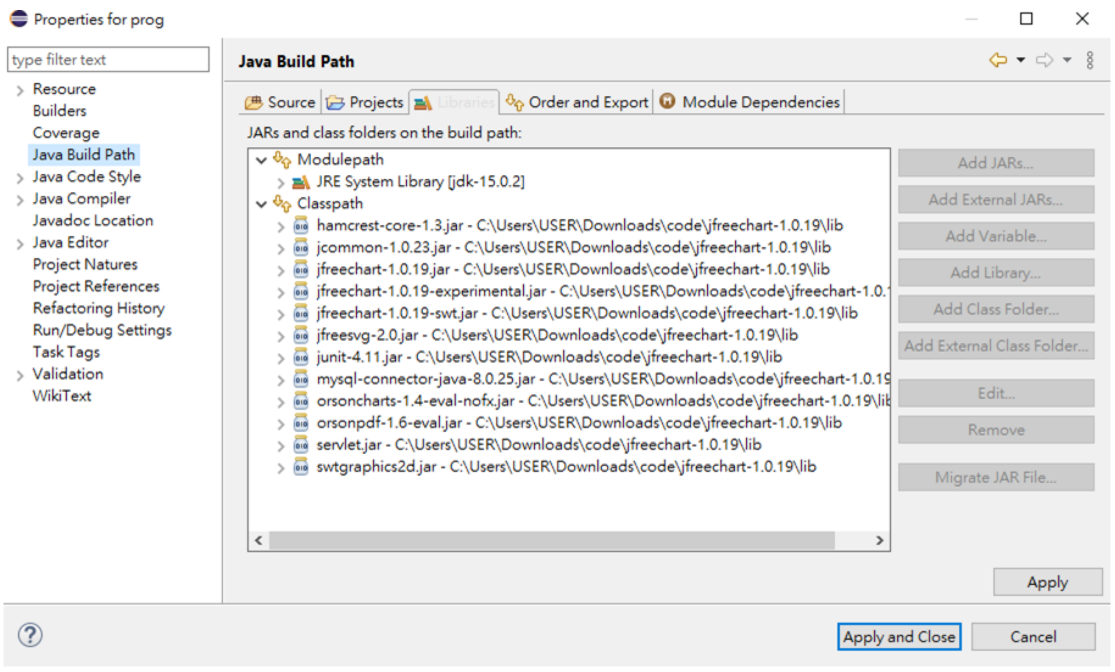
```
463  public void positionColumn(JTable table,int col_Index) {
464      table.moveColumn(table.getColumnCount()-1, col_Index);
465      }
```

Accounting4.java

Line18~26:Import org.jfree.chart my add external jar in library

```
18 import org.jfree.chart.ChartFactory;
19 import org.jfree.chart.ChartPanel;
20 import org.jfree.chart.JFreeChart;
21 import org.jfree.chart.StandardChartTheme;
22 import org.jfree.chart.labels.PieSectionLabelGenerator;
23 import org.jfree.chart.labels.StandardPieSectionLabelGenerator;
24 import org.jfree.chart.plot.PiePlot;
25 import org.jfree.chart.title.TextTitle;
26 import org.jfree.data.general.DefaultPieDataset;
```

Right Click the project find properity and go to java build path

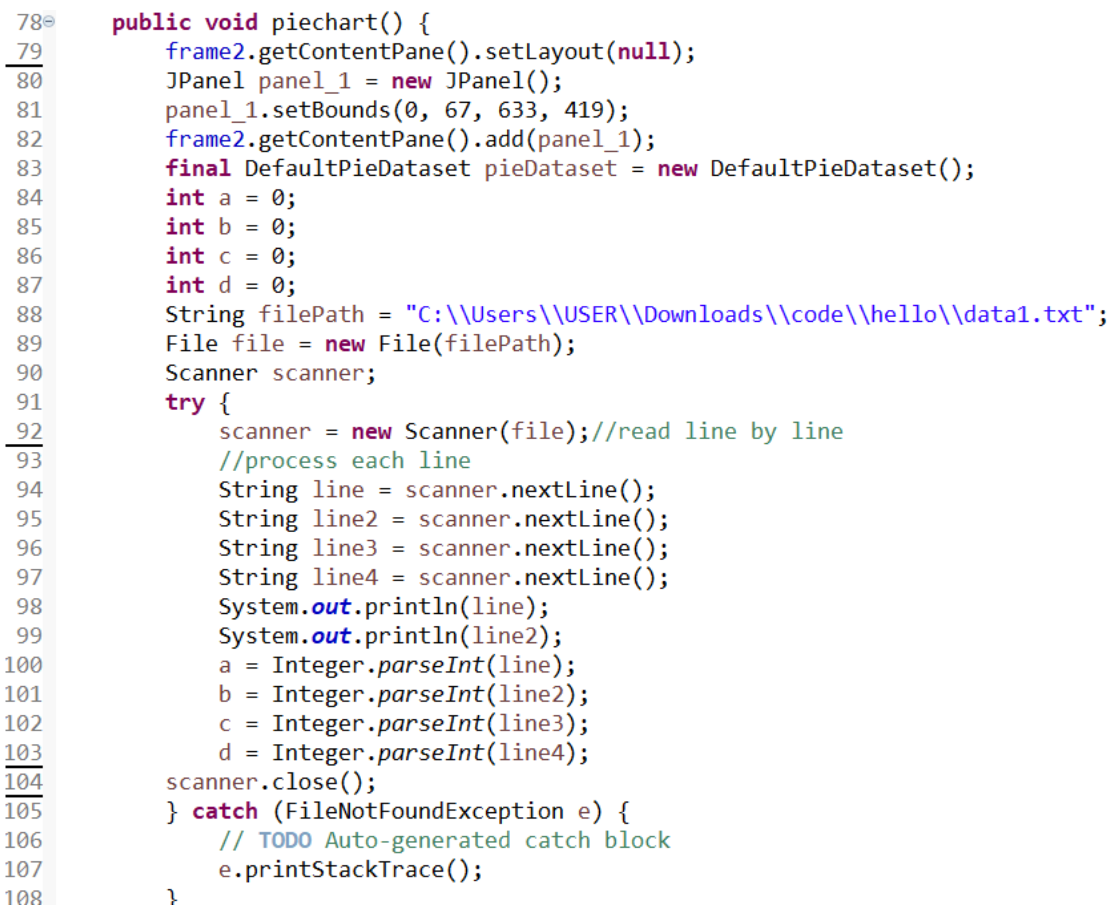


JARs and class folders on the build path:

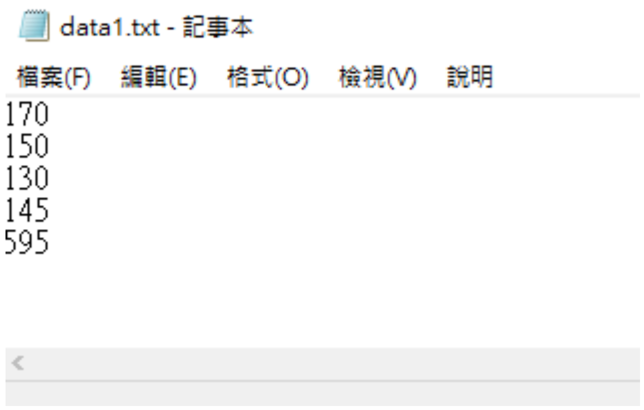
```
Modulepath
  JRE System Library [jdk-15.0.2]
Classpath
  hamcrest-core-1.3.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
  jcommon-1.0.23.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
  jfreechart-1.0.19.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
  jfreechart-1.0.19-experimental.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.1
  jfreechart-1.0.19-swt.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
  jfreesvg-2.0.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
  junit-4.11.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
  mysql-connector-java-8.0.25.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19
  orsoncharts-1.4-eval-nofx.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
  orsonpdf-1.6-eval.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
  servlet.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
  swtgraphics2d.jar - C:\Users\USER\Downloads\code\jfreechart-1.0.19\lib
```

Add external jars

Line78~108: Firstly, we create a DefaultPieSet,and import our data from Line:270~296,use scanner to scan the file from every layer



```
 78      public void piechart() {
 79          frame2.getContentPane().setLayout(null);
 80          JPanel panel_1 = new JPanel();
 81          panel_1.setBounds(0, 67, 633, 419);
 82          frame2.getContentPane().add(panel_1);
 83          final DefaultPieDataset pieDataset = new DefaultPieDataset();
 84          int a = 0;
 85          int b = 0;
 86          int c = 0;
 87          int d = 0;
 88          String filePath = "C:\\Users\\USER\\Downloads\\code\\hello\\data1.txt";
 89          File file = new File(filePath);
 90          Scanner scanner;
 91          try {
 92              scanner = new Scanner(file);//read line by line
 93              //process each line
 94              String line = scanner.nextLine();
 95              String line2 = scanner.nextLine();
 96              String line3 = scanner.nextLine();
 97              String line4 = scanner.nextLine();
 98              System.out.println(line);
 99              System.out.println(line2);
100              a = Integer.parseInt(line);
101              b = Integer.parseInt(line2);
102              c = Integer.parseInt(line3);
103              d = Integer.parseInt(line4);
104          scanner.close();
105          } catch (FileNotFoundException e) {
106              // TODO Auto-generated catch block
107              e.printStackTrace();
108          }
```



```
170
150
130
145
595
```

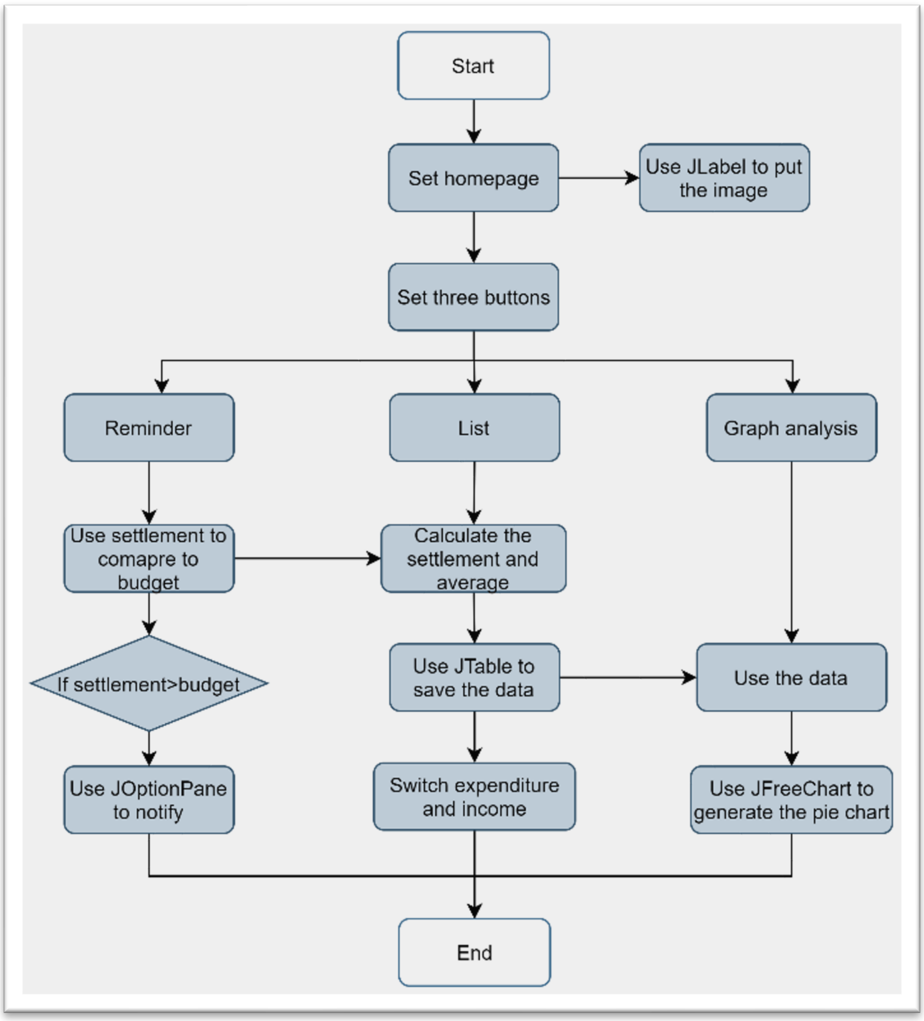
Line 109~136: we create a PieChart3D,use the data we get from text file,chane eah data to percentage.Print it on the chart

```
        pieDataset.setValue("breakfest", a);
        pieDataset.setValue("lunch", b);
        pieDataset.setValue("dinner", c);
        pieDataset.setValue("daily supply", d);
        JFreeChart chart = ChartFactory.createPieChart3D("Expenditure", pieDataset, true, true, false);
        PiePlot plot = (PiePlot) chart.getPlot();
          PieSectionLabelGenerator gen = new StandardPieSectionLabelGenerator(
                    "{0}: {1} ({2})", new DecimalFormat("0"), new DecimalFormat("0%"));
                plot.setLabelGenerator(gen);
        panel_1.setLayout(null);
        ChartPanel frame = new ChartPanel(chart);
        frame.setBounds(-14, 0, 680, 420);
        panel_1.add(frame);
        frame.setLayout(null);

         JButton btnNewButton_1 = new JButton("  ");
         btnNewButton_1.setBounds(0, 0, 61, 38);
         frame2.getContentPane().add(btnNewButton_1);
         btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
              frame2.setVisible(false);
            }
         });
         btnNewButton_1.setFont(new Font("Dialog", Font.BOLD, 14));
         btnNewButton_1.setBackground(new Color(255, 250, 250));


        frame.setVisible(true);
```

2.8 FlowChart


Start
Set homepage
Use JLabel to put the image
Set three buttons
Reminder
List
Graph analysis
Use settlement to comapre to budget
Calculate the settlement and average
If settlement>budget
Use JTable to save the data
Use the data
Use JOptionPane to notify
Switch expenditure and income
Use JFreeChart to generate the pie chart
End

3.Results / findings

Through this program we can get:
1. Users can enter the cost or income and choose the classification.
2. Data can be accessed and output
3. Can be analyzed with graphs.
4. Can set the reminder to inform the user over the budget.
5. we can store data in to local file.
6.Use Mysql to let our data into database.

4.Conclusion

This research paper presents an introduction to our accounting program and how to do this program.
We can use Graphically User Interfaces (GUI) to design the customized visualization, use JTable to store data, use JOptionPane to design the notification, and use JFreeChart to generate the chart to visualize the data.
Storing data is important in this finical program,so we export our data into csv, and txt. File respectively.
From what has been discussed above, we can use this accounting program to achieve the effect of financial management.

5.Limitations

This research report shows how to overcome the limitation of time, and technical problem. Because of time limitations, we only can finish the program part. If we have enough time we can do some research and use android studio.

6.References /Bibliography

https://www.jfree.org/jfreechart/api/javadoc/org/jfree/chart/ChartFactory.html
https://1bestcsharp.blogspot.com/2015/04/java-io-how-to-export-jtable-data-to-txt-File-In-Java.html
https://commons.apache.org/proper/commons-configuration/userguide/howto_properties.html
https://examples.javacodegeeks.com/core-java/sql/import-csv-file-to-mysql-table-java-example/
https://www.jfree.org/jfreechart/
https://puremonkey2010.blogspot.com/2011/06/java-jfreechart-part-1.html
https://www.1ju.org/jfreechart/jfreechart-pie-chart
https://stackoverflow.com/questions/13579810/how-to-import-data-from-text-file-

to-mysql-database
https://www.itread01.com/p/1123099.html
https://topic.alibabacloud.com/tc/a/importing-and-exporting-txt-files-in-mysql_1_41_32056352.html
https://dev.mysql.com/doc/refman/8.0/en/loading-tables.html
https://xken831.pixnet.net/blog/post/427790957-%5Bjava%5D%E4%BD%BF%E7%94%A8-java-%E9%80%A3%E7%B5%90-mysql
https://blog.judysocute.com/2020/06/08/%E7%AC%AC-18-%E9%80%B1-java-%E9%80%A3%E6%8E%A5-mysql/
http://www.tsnien.idv.tw/Java2_WebBook/chap11/11-4%20Java%20+%20MySQL%20%E9%80%A3%E7%B5%90.html
https://dev.mysql.com/downloads/connector/j/
https://stackoverflow.com/questions/6232355/deleting-all-the-rows-in-a-jtable
https://stackoverflow.com/questions/4577792/how-to-clear-jtable/4578501
https://www.c-sharpcorner.com/UploadFile/fd0172/display-records-from-database-using-jtable-in-java/
https://www.youtube.com/watch?v=frafcK6fhdQ