**Cafeteria Management**



Done by Maclean N

Submitted to Junnie Denny Solomon

**Summary**

* The topic of my project is a method for cafeteria billing. It is a tool for vendor-side billing. A list of the cafe's products will be available on the dashboard.
* Just the count needs to be added, and the product needs to be enabled.
* The bill is provided in the text area, and the bill number is produced at random.
* Also, the bill includes the product's name and quantity.
* Finally, there are four buttons: total, receipt, reset bill, and exit.

**Introduction**

The bill session will appear more complicated in the cafeteria. I planned to create a straightforward application for cafeteria since I believed that I could make it simple. It would be quite difficult to maintain a massive application created by a large team to develop a cafe, therefore I choose to keep it straightforward.

I began creating the café management system with the aid of Java and Netbeans.

The topic of my project is a method for cafeteria billing. It is a tool for vendor-side billing. A list of the cafe's products will be available on the dashboard.

Just the count needs to be added, and the product needs to be enabled.

The bill is provided in the text area, and the bill number is produced at random.

Also, the bill includes the product's name and quantity.

Finally, there are four buttons: total, receipt, reset bill, and exit.

**Tools used**

Apache netbeans

**Core language**

Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java, its name was Oak. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

**The main Function used In java :-**

**Java swing**

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

**Java jbutton**

The JButton class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed. It inherits AbstractButton class.

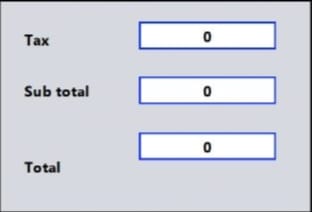


**Java jlabel**

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

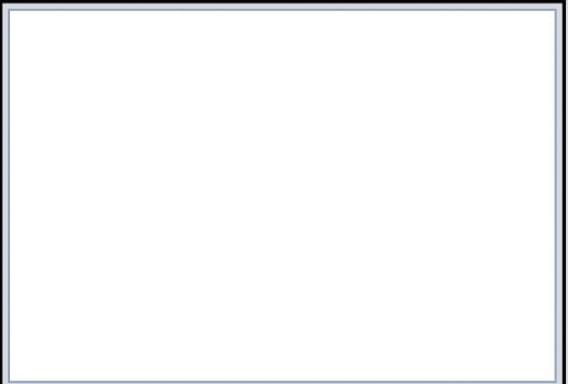
**Java jtextfield**

The object of a JTextField class is a text component that allows the editing of a single line text. It inherits JTextComponent class.



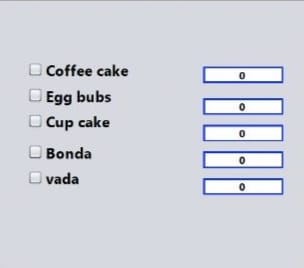
**Java jtextArea**

The object of a JTextArea class is a multi line region that displays text. It allows the editing of multiple line text. It inherits JTextComponent class.



**Java jCheckBox**

The JCheckBox class is used to create a checkbox. It is used to turn an option on (true) or off (false). Clicking on a CheckBox changes its state from "on" to "off" or from "off" to "on ".It inherits [JToggleButton](https://www.javatpoint.com/java-jtogglebutton) class.



**Java JPanel**

The JPanel is a simplest container class. It provides space in which an application can attach any other component. It inherits the JComponents class.

It doesn't have title bar.

**Java JFrame**

The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.

Unlike Frame, JFrame has the option to hide or close the window with the help of setDefaultCloseOperation(int) method.

**Java Action Listener interface**

The Java Action Listener is notified whenever you click on the button or menu item. It is notified against Action Event. The Action Listener interface is found in java.awt.event [package](https://www.javatpoint.com/package). It has only one method: action Performed().

**Java mouse Listener interface**

The Java MouseListener is notified whenever you change the state of mouse. It is notified against MouseEvent. The MouseListener interface is found in java.awt.event package. It has five methods.

**Java Layout manager**

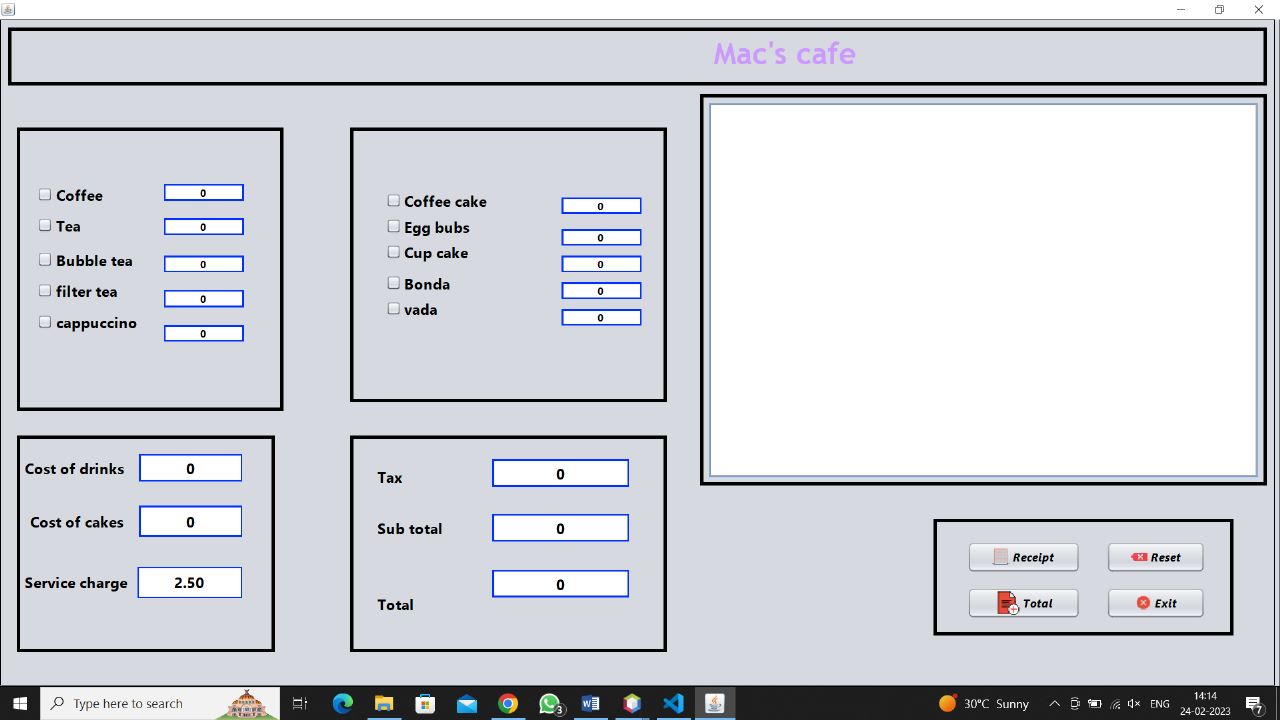
The LayoutManagers are used to arrange components in a particular manner. The **Java LayoutManagers** facilitates us to control the positioning and size of the components in GUI forms. LayoutManager is an interface that is implemented by all the classes of layout managers. There are the following classes that represent the layout managers:

1. java.awt.BorderLayout
2. java.awt.FlowLayout
3. java.awt.GridLayout
4. java.awt.CardLayout
5. java.awt.GridBagLayout
6. javax.swing.BoxLayout
7. javax.swing.GroupLayout
8. javax.swing.ScrollPaneLayout
9. javax.swing.SpringLayout etc.

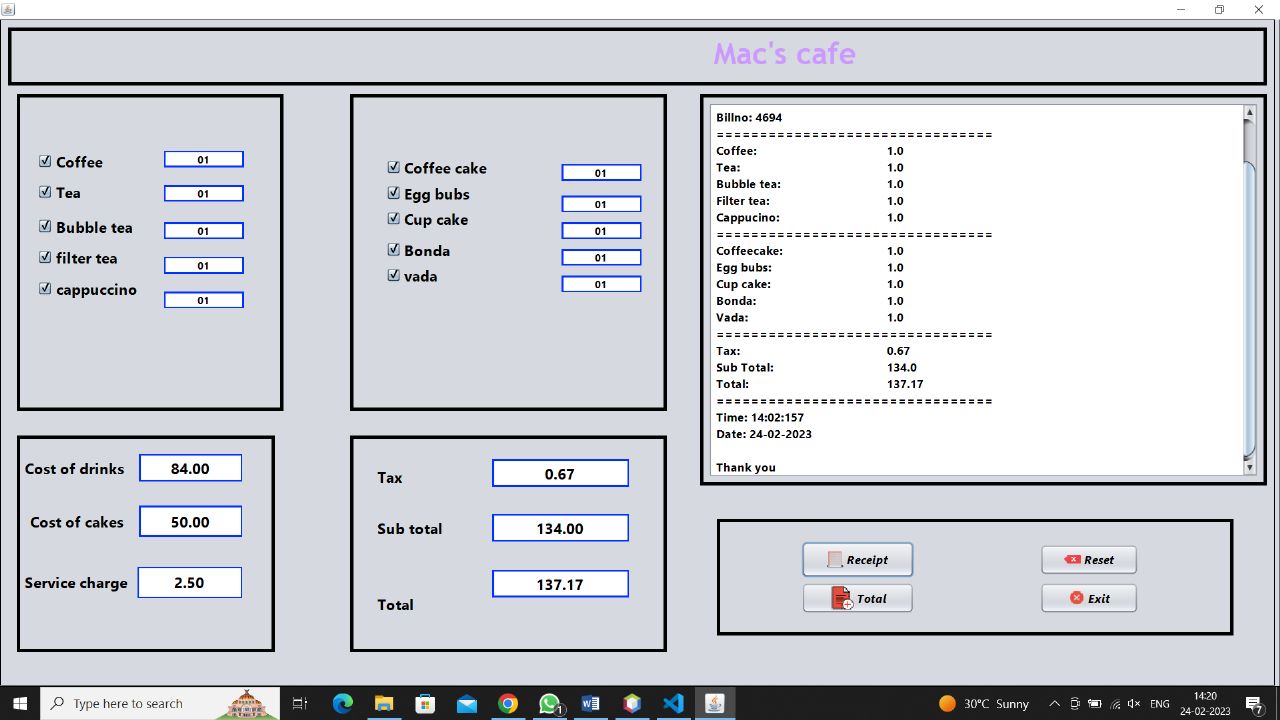
**java Date and time**

The java.time, java.util, java.sql and java.text packages contains classes for representing date and time. Following classes are important for dealing with date in Java.

**Application preview**



**Application preview with output**



**Future work**

In the future, I'll extend the programme to connect to a database, store each segment, and create a report page for this project to means of responding.