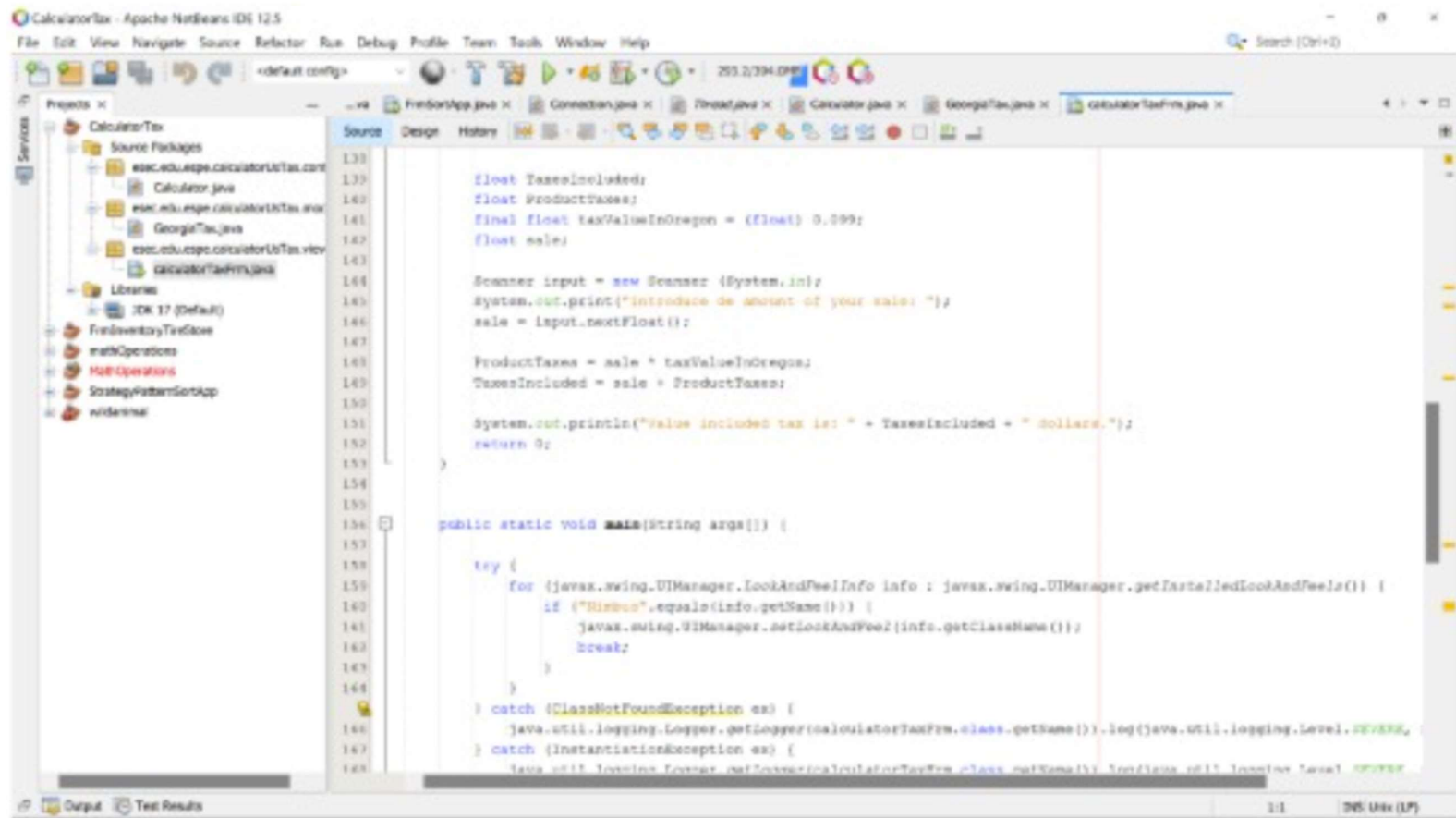


16 DE LA CRUZ QUINGA ALEJANDRO SEBASTIAN

Singleton Pattern es un patrón de creación que garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ella; este patrón rompe el principio de responsabilidad única porque realiza estas dos acciones.



```
130
131
132 float TaxesIncluded;
133 float ProductTaxes;
134 final float taxValueInOregon = (float) 0.099;
135 float sale;
136
137 Scanner input = new Scanner (System.in);
138 System.out.print("Introduce the amount of your sale: ");
139 sale = input.nextFloat();
140
141 ProductTaxes = sale * taxValueInOregon;
142 TaxesIncluded = sale + ProductTaxes;
143
144 System.out.println("Value included tax is: " + TaxesIncluded + " dollars.");
145 return 0;
146
147
148 public static void main(String args[]) {
149
150     try {
151         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
152             if ("Nimbus".equals(info.getName())) {
153                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
154                 break;
155             }
156         }
157     } catch (ClassNotFoundException ex) {
158         java.util.logging.Logger.getLogger(CalculatorTaxFW.class.getName()).log(java.util.logging.Level.SEVERE,
159         null, ex);
160     } catch (InstantiationException ex) {
161         java.util.logging.Logger.getLogger(CalculatorTaxFW.class.getName()).log(java.util.logging.Level.SEVERE,
162         null, ex);
163     }
```