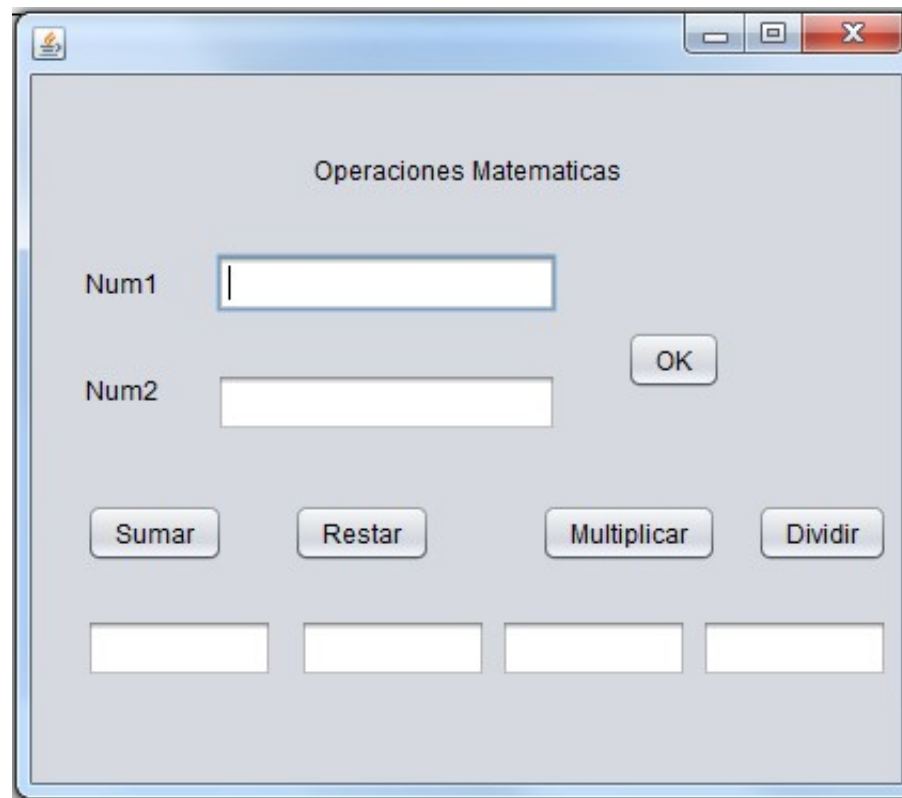


EJERCICIO PRUEBAS CAJA NEGRA Y CAJA BLANCA

Se realizó la prueba Caja Negra y Caja Blanca de una calculadora desarrollada en JAVA que funciona de la siguiente manera:

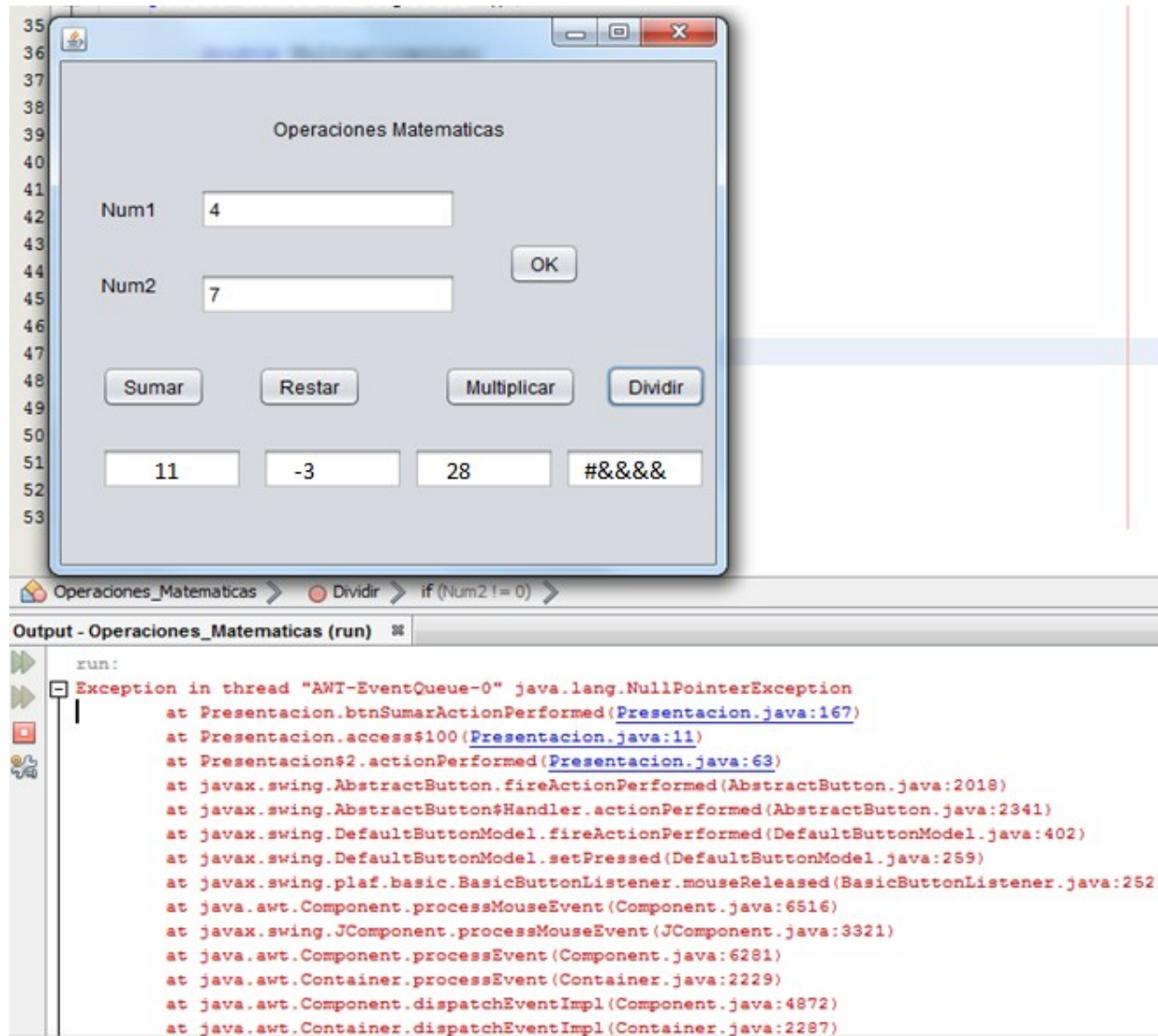
1. Se debe ingresar dos números a las cajas de texto (Num1 y Num2) y paso seguido confirmar los números con el botón OK
2. De acuerdo a la operación que se desee realizar, se visualizara el resultado oprimiendo cualquiera de los botones (Sumar, Restar, Multiplicar, Dividir)



The image shows a Java Swing window titled "Operaciones Matematicas". It features a light blue background and a standard Windows-style title bar with minimize, maximize, and close buttons. The interface includes two text input fields labeled "Num1" and "Num2". To the right of these fields is an "OK" button. Below the input fields, there are four buttons labeled "Sumar", "Restar", "Multiplicar", and "Dividir". At the bottom of the window, there are four empty rectangular boxes, likely intended for displaying the results of the operations.

1. PRUEBA CAJA NEGRA

Al realizar la prueba Caja Negra de la calculadora se obtuvo el siguiente resultado:



1. Las operaciones Sumar, Restar , Multiplicar están correcto pero la división arroja una serie de caracteres incoherentes con respecto a la operación

2. PRUEBA CAJA BLANCA

Razón el cual se procedió a realizar la prueba Caja Blanca. Consiste en realizar pruebas técnicas propiamente del código de desarrollo de la calculadora y se obtuvieron los siguientes resultados:

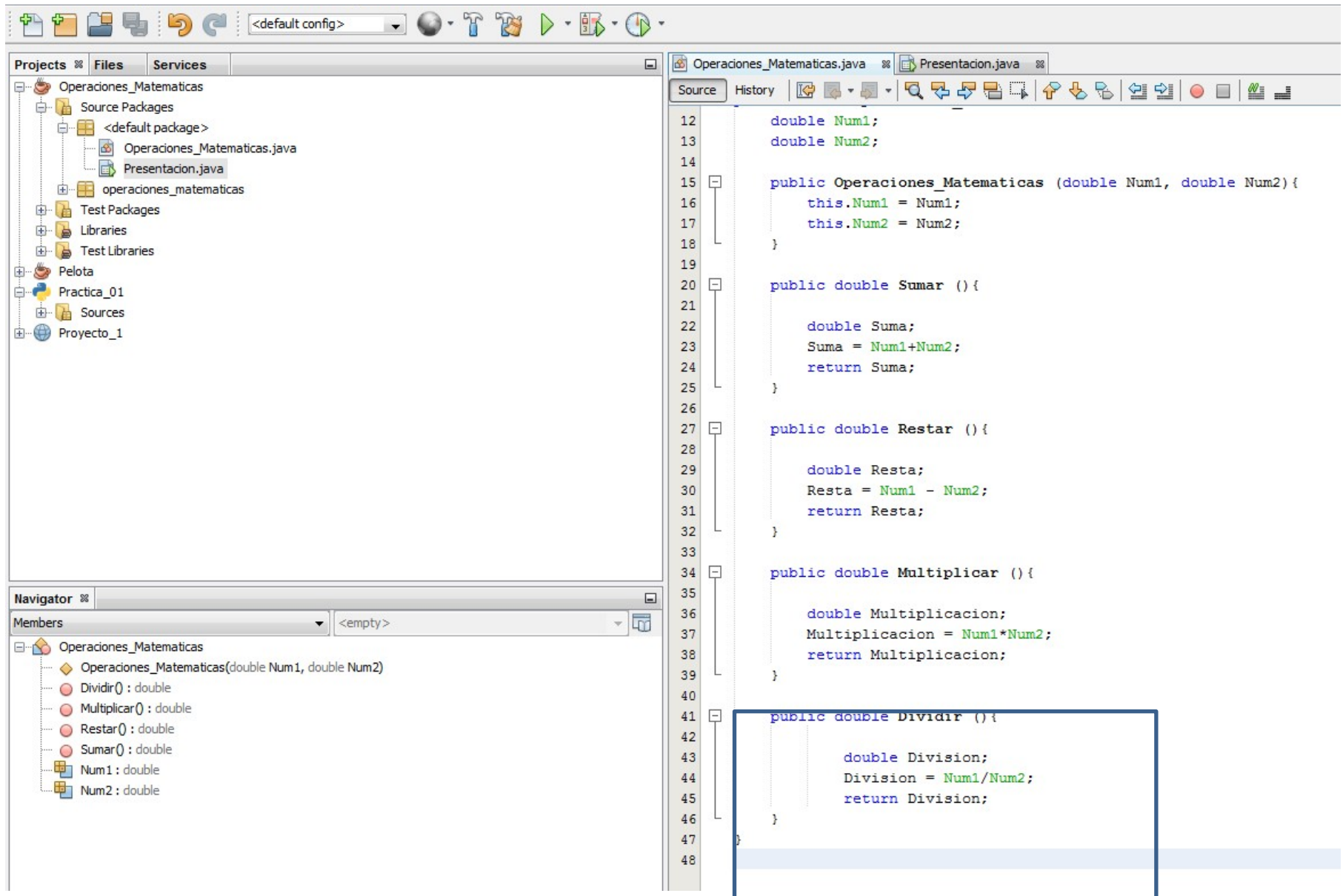
The screenshot displays an IDE with the following components:

- Projects Panel:** Shows a project named 'Operaciones_Matematicas' with source packages and test packages.
- Source Editor:** Displays the code for 'Operaciones_Matematicas.java' and 'Presentacion.java'. The code includes methods for performing mathematical operations and a main method to initialize the GUI.
- Navigator Panel:** Shows the 'btnDividirActionPerformed' method in the 'Presentacion' class.

The code in the source editor is as follows:

```
167 Double Suma = OP.Sumar();
168 txtSuma.setText(String.valueOf(Suma));
169 }
170
171 private void btnMultiplicarActionPerformed(java.awt.event.ActionEvent evt) {
172     // TODO add your handling code here:
173     Double producto = OP.Multiplicar();
174     txtMultiplicacion.setText(String.valueOf(producto));
175 }
176
177 private void btnRestarActionPerformed(java.awt.event.ActionEvent evt) {
178     // TODO add your handling code here:
179     Double Resta = OP.Restar();
180     txtResta.setText(String.valueOf(Resta));
181 }
182
183 private void btnDividirActionPerformed(java.awt.event.ActionEvent evt) {
184     // TODO add your handling code here:
185     Double Division = OP.Dividir();
186     if (Division != Double.MAX_VALUE) {
187         txtDivision.setText(String.valueOf(Division));
188     } else {
189         txtDivision.setText("Error, Divisor en 0");
190     }
191 }
192
193 }
194
195 private void btnOKActionPerformed(java.awt.event.ActionEvent evt) {
196     // TODO add your handling code here:
197     OP = new Operaciones_Matematicas(Double.parseDouble(txtNum1.getText()), Double.parseDouble(txtNum2.getText()));
198 }
199
200 /**
201  * @param args the command line arguments
202  */
203 public static void main(String args[]) {
204     /* Set the Nimbus look and feel */
205     Look and feel setting code (optional)
206
207     /* Create and display the form */
208     java.awt.EventQueue.invokeLater(new Runnable() {
209         public void run() {
210             new Presentacion().setVisible(true);
211         }
212     });
213 }
```

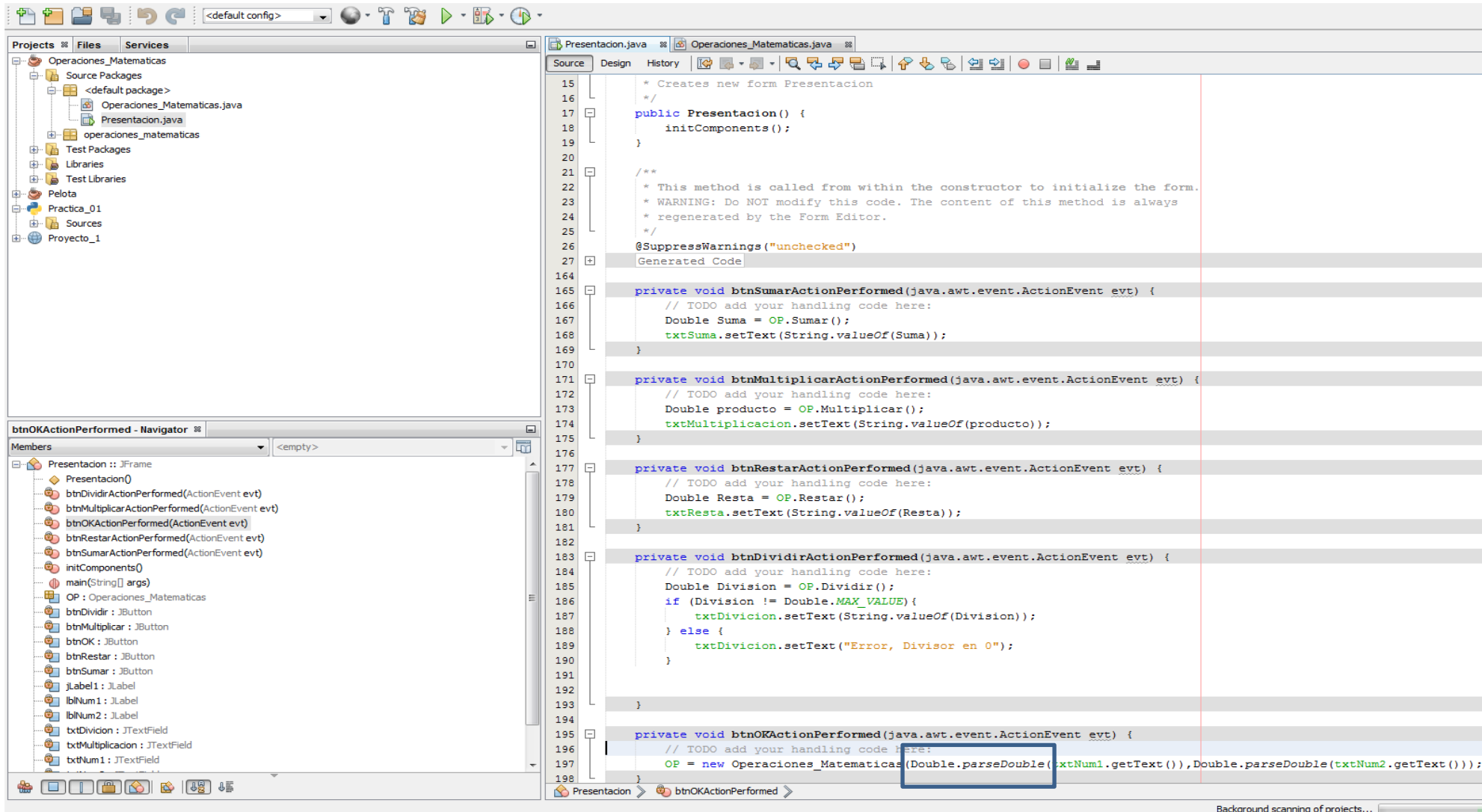
1. El primer error se encuentra en la programación del botón OK, que es el encargado de convertir los números ingresados por el usuario de tipo carácter a tipo numérico. El error consiste en la forma de conversión a tipo numérico, ya que los números ingresados por el cliente de acuerdo a la operación Dividir debe ser convertidos los números a tipos FLOTANTE y no de tipo ENTERO



2. El segundo error se encuentra en la base principal de desarrollo de las funciones de la calculadora. El error consiste en el método Dividir ya que no previene la posibilidad de que el cliente pueda ingresar un cero (0) en el denominador en el cual matemáticamente es un error

SOLUCION DE LOS ERRORES

1. Para el primer error se cambió la opción de conversión de `parseInt` a `parseDouble` para que acepte valores de tipo entero y de tipo coma flotante, así cualquier número que digite el usuario pueda ser procesado por el método Dividir() y dé el resultado exacto de la operación



2. Para el segundo error se ingreso el algoritmo de programacion IF, ELSE y su función es la de preguntar si el denominador es diferente de cero (0), en el caso que sea verdadero, muestre un valor alto controlado Incoherente en el resultado en el cual indique que es indeterminada la operación matematicamente

The screenshot displays an IDE with the following components:

- Projects View:** Shows a project named 'Operaciones_Matematicas' containing source packages, test packages, and libraries. The 'Sources' folder is expanded, showing 'Operaciones_Matematicas.java' and 'Presentacion.java'.
- Operaciones_Matematicas - Navigator:** Lists the members of the 'Operaciones_Matematicas' class, including the constructor 'Operaciones_Matematicas(double Num1, double Num2)' and methods 'Dividir()', 'Multiplicar()', 'Restar()', 'Sumar()', 'Num1', and 'Num2'.
- Source Editor:** Displays the code for 'Operaciones_Matematicas.java'. The code includes a class declaration with two double variables, a constructor, and four methods: 'Sumar()', 'Restar()', 'Multiplicar()', and 'Dividir()'. The 'Dividir()' method uses an if-else statement to handle division by zero, returning 'Double.MAX_VALUE' if the denominator is zero.

```
7  /**
8  *
9  * @author wilder.gomez
10 */
11 public class Operaciones_Matematicas {
12     double Num1;
13     double Num2;
14
15     public Operaciones_Matematicas (double Num1, double Num2) {
16         this.Num1 = Num1;
17         this.Num2 = Num2;
18     }
19
20     public double Sumar () {
21
22         double Suma;
23         Suma = Num1+Num2;
24         return Suma;
25     }
26
27     public double Restar () {
28
29         double Resta;
30         Resta = Num1 - Num2;
31         return Resta;
32     }
33
34     public double Multiplicar () {
35
36         double Multiplicacion;
37         Multiplicacion = Num1*Num2;
38         return Multiplicacion;
39     }
40
41     public double Dividir () {
42
43         if (Num2!=0) {
44             double Division;
45             Division = Num1/Num2;
46             return Division;
47         } else {
48             return Double.MAX_VALUE;
49         }
50     }
51 }
52
```


- Finalizada las pruebas CAJA BLANCA, la Calculadora funciona correctamente con todos sus funciones incluyendo todo tipo de valores numéricos como se refleja en la figura

