**DOCUMENTACIÓN DEL CÓDIGO DE LA CALCULADORA EN JAVA**

**PRESENTADO A LA PROFESORA:**

**LELI LILIANA DÍAZ IZQUIERDO**

**PRESENTADO POR EL ESTUDIANTES:**

**LUIS CARLOS GARAVITO AGAMEZ**

**SEGUNDO SEMETRES**

**INGENIERÍA SISTEMAS**

**CORPORACIÓN UNIVERSITARIA REMINGTON 2024-1**

Este código documenta una aplicación de calculadora simple implementada en Java.

**Descripción general:**

Características básicas:

* Realiza operaciones matemáticas básicas: suma, resta, multiplicación y división.
* Permite ingresar números decimales.
* Muestra el resultado de las operaciones en una pantalla.
* Borra el último dígito ingresado.

Funcionalidades adicionales:

* Calcula porcentajes.
* Realiza exponenciaciones.
* Calcula la raíz cuadrada.

Aspectos técnicos:

* Utiliza la biblioteca Swing de Java para la GUI.
* Maneja eventos de los botones y otros componentes de la interfaz.
* Realiza validaciones básicas de entrada de datos.

**Detalle de las clases:**

Importamos: javax.swing.JOptionPane

* Importamos la clase JOptionPane de la biblioteca javax.swing para la interacción con el usuario.

Clase: NewJFrame

* Esta clase representa la ventana principal de la aplicación de calculadora.
* Extiende la javax.swing.JFrameclase, que proporciona la funcionalidad básica para una ventana en una aplicación Swing.

**Explicación de las variables:**

Variable que la conforma:

* primerNumero:

En esta variable se almacena el primer número ingresado por el usuario.

* SegundoNumero:

En esta variable se almacena el segundo número ingresado por el usuario.

* operador:

En la variable operador almacenamos todas las operaciones seleccionadas (unos ejemplos de estas operaciones son: ^, /, \*, +, -, %, mod).

* resultado:

En la variable se almacena los resultados de los diferentes cálculos realizados.

* pantalla:

Esta variable es una referencia a un javax.swing.JTextAreaobjeto que muestra la salida de la calculadora.

**Detalle de los** **métodos:**

* initComponents():

El marco Java Swing llama a este método para inicializar los componentes gráficos (botones, área de texto, etc.) de la ventana.

* borraActionPerformed(ActionEvent evt):

Este método maneja el evento de clic en el botón "<=". Borra el área de texto (pantalla).

* // Other buttonActionPerformed methods:

Estos métodos manejan los eventos de clic en los botones de la calculadora (números, operadores, etc.). Actualizan la pantalla (pantalla) según la entrada del usuario y la operación seleccionada.

* igualActionPerformed(ActionEvent evt):

Este método maneja el evento de clic en el botón "=". Calcula el resultado en función de los números almacenados (primerNumero, segundoNumero), la operación seleccionada (operador) y muestra el resultado en la pantalla. También realiza una comprobación de errores de división por cero.

* porcientoActionPerformed(ActionEvent evt): Este método maneja el evento de clic en el botón "%". Calcula el número ingresado como porcentaje (divide por 100) y muestra el resultado.
* expoActionPerformed(ActionEvent evt): Este método maneja el evento de clic en el botón "^" (exponenciación). Solicita al usuario un exponente mediante un cuadro de diálogo, realiza el cálculo de la exponenciación y muestra el resultado. También realiza una comprobación de errores de bases negativas.
* RaizActionPerformed(ActionEvent evt): Este método maneja el evento de clic en el botón "√" (raíz cuadrada). Calcula la raíz cuadrada del número ingresado y muestra el resultado. Realiza una comprobación de errores de números negativos (la raíz cuadrada de un número negativo no es un número real).

En general, este código demuestra cómo crear una aplicación de calculadora sencilla con una interfaz gráfica de usuario utilizando Java Swing.

Notas adicionales:

* El código utiliza las funciones matemáticas integradas de Java, como Math.pow la exponenciación y Math.sqrtla raíz cuadrada.
* El código incluye manejo de errores básico para la división por cero y bases negativas en exponenciación.
* Puede mejorar aún más este código agregando más funcionalidades como manejar operaciones más complejas (por ejemplo, funciones trigonométricas), implementar la segunda pantalla, que pueda realizar varias operaciones en mismo proceso etc.

Reflexión sobre el trabajo en equipo y el proceso desarrollo:

* El trabajo en equipo estuvo bien nos ayudamos buscado la información y compartiendo las ideas.
* El proceso del desarrollo del código toco buscar en la documentación de java y videos de YouTube, en los códigos de personas que programaron una calculadora básica y editando cada parte para ver cuál el mejor para mejorar el código.