Proyecto Calculadora

Documentación de Clases y Métodos

Elaborado por: Diego Chevez

Índice

[1. Introducción 1](#_Toc202370432)

[2. Number.java 1](#_Toc202370433)

[3. Operation.java 1](#_Toc202370434)

[4. Calculator.java 2](#_Toc202370435)

[5. Record.java 2](#_Toc202370436)

[6. Menu.java 3](#_Toc202370437)

[7. Main.java 3](#_Toc202370438)

[8. Conclusión 4](#_Toc202370439)

# 1. Introducción

Este documento presenta la documentación técnica, del proyecto de una calculadora de consola desarrollada en Java. La documentación incluye la descripción de cada clase, sus atributos y métodos, facilitando la comprensión, uso y desarrollo del mismo.

# 2. Number.java

La clase **Number** encapsula un número real utilizando el tipo de dato *double*, permitiendo el acceso controlado al valor mediante métodos *getter* y *setter*, promoviendo el encapsulamiento en el proyecto.

Atributos:

* **double value:** almacena el valor numérico.

Métodos:

* **Number(double value):** constructor que inicializa el valor.
* **double getValue():** devuelve el valor numérico almacenado.
* **void setValue(double value):** actualiza el valor numérico almacenado.

# 3. Operation.java

La enumeración **Operation** define los tipos de operaciones disponibles en la calculadora, evitando el uso de cadenas mágicas y mejorando la claridad en el código.

Valores enumerados:

* ADDITION
* SUBTRACTION
* MULTIPLICATION
* DIVISION
* POWER
* SQUARE\_ROOT

# 4. Calculator.java

La clase **Calculator** contiene métodos estáticos que realizan operaciones aritméticas utilizando instancias de la clase **Number** como parámetros. Gestiona las operaciones básicas, potencias y raíces cuadradas, incluyendo el manejo de excepciones para divisiones entre cero y raíces cuadradas de números negativos.

Métodos:

* **double add(Number a, Number b):** retorna la suma de dos números.
* **double subtract(Number a, Number b):** retorna la diferencia entre dos números.
* **double multiply(Number a, Number b):** retorna el producto de dos números.
* **double divide(Number a, Number b):** retorna el cociente de dos números; lanza *ArithmeticException* si hay división entre cero.
* **double power(Number base, Number exponent):** retorna la potencia de un número elevado a otro.
* **double squareRoot(Number number):** retorna la raíz cuadrada de un número; lanza *ArithmeticException* si el número es negativo.

# 5. Record.java

La clase **Record** gestiona el historial en memoria de las operaciones realizadas durante la ejecución de la calculadora, almacenando una lista de representaciones en cadena de cada operación para su revisión posterior.

Atributos:

* **List<String> operations:** almacena las operaciones registradas.

Métodos:

* **void addOperation(String operation):** agrega una operación al historial.
* **void showHistory():** muestra en consola el historial de operaciones registradas.

# 6. Menu.java

La clase **Menu** gestiona la interfaz de usuario a través de consola, mostrando el menú principal y submenús para la selección y ejecución de operaciones. Controla la entrada de datos del usuario, muestra resultados y permite una interacción clara y estructurada con la calculadora.

Atributos:

* **Scanner scanner:** gestiona la entrada de datos del usuario.
* **Record record:** gestiona el historial de operaciones realizadas.

Métodos:

* **void showMainMenu():** muestra el menú principal y gestiona la navegación de la aplicación.
* **void handleOperation(Operation op):** gestiona la ejecución de una operación seleccionada, permitiendo la repetición de operaciones del mismo tipo.
* **int getIntInput():** obtiene de forma segura un entero del usuario, manejando entradas inválidas.
* **double getDoubleInput():** obtiene de forma segura un número decimal del usuario, manejando entradas inválidas.
* **void clearConsole():** intenta limpiar la pantalla de la consola para una mejor experiencia de usuario.

# 7. Main.java

La clase **Main** sirve como punto de entrada de la aplicación. Inicializa las instancias de **Record** y **Menu**, y lanza el menú principal, iniciando la sesión interactiva de la calculadora.

Métodos:

* **main(String[] args):** inicializa y ejecuta la aplicación de la calculadora.

# 8. Conclusión

Esta documentación técnica facilita la comprensión de la estructura del proyecto, permitiendo un uso adecuado, una posible ampliación de funcionalidades y un mantenimiento ordenado del proyecto de calculadora en Java.