

CALCULADORA OPERATIVA

Integrantes:

Juan Pablo Herrera Jaramillo

Santiago Castañeda Giraldo

Profesora:

Diana Margot Margot Lopez Herrera

Materia:

Técnicas de Programación y Laboratorio

Universidad de Antioquia

Facultad de Ingeniería

Medellín

2024-2

Documentación del Proyecto:

1. Descripción General:

Este proyecto implementa una calculadora que puede realizar operaciones tanto aritméticas como de bits, permitiendo realizar operaciones como suma, resta, multiplicación, división, potenciación, raíz cuadrada, y operaciones lógicas bit a bit.

Este sistema incluye las respectivas validaciones de cada operación para que el funcionamiento sea óptimo.

2. Estructura del Proyecto:

El proyecto consta de las siguientes clases:

- **Calculadora:** Contiene el método principal para ejecutar la calculadora.
- **Menú:** Presenta las opciones al usuario y maneja la interacción entre el usuario y los operadores.
- **OperadoresArit:** Implementa las operaciones aritméticas.
- **OperadoresBit:** Implementa las operaciones bit a bit.
- **Validaciones:** Se encarga de validar las entradas del usuario y manejar los posibles errores de entrada.

3. Clases:

3.1. Clase Calculadora

- **Descripción:** Esta es la clase principal que ejecuta el programa. Utiliza `BufferedReader` para capturar la entrada del usuario y delega las operaciones a la clase `Menu`.
- **Excepciones:** Maneja dos tipos de excepciones:
 - `IOException`: Ocurre en caso de errores de entrada y salida.
 - `Exception`: Maneja cualquier otro error inesperado que pueda ocurrir durante la ejecución.

3.2. Clase Menu

- **Descripción:** Esta clase se encarga de la interfaz de usuario. Presenta las opciones disponibles para realizar operaciones aritméticas o de bits, y llama a los métodos correspondientes según la elección del usuario.

- Este método muestra un menú con opciones y captura la elección del usuario, llamando a las operaciones aritméticas o de bits según corresponda.

- **Operaciones Disponibles:**
 - **Operadores Aritméticos:**
 - Suma
 - Resta
 - Multiplicación
 - División
 - Potenciación
 - Raíz Cuadrada
 - **Operadores de Bits:**
 - Suma
 - Resta
 - Multiplicación
 - División
 - AND
 - OR
 - XOR
 - Desplazamiento a la derecha
 - Desplazamiento a la izquierda

3.3. Clase OperadoresArit

- **Descripción:** Esta clase implementa las operaciones aritméticas básicas como suma, resta, multiplicación, división, potenciación y raíz cuadrada.

- **Atributos:**
 - `private double resultado`: Almacena el resultado de las operaciones.

- **Métodos:**
 - **sumar(double a, double b)**: Suma dos números.
 - **restar(double a, double b)**: Resta dos números.
 - **multiplicar(double a, double b)**: Multiplica dos números.
 - **dividir(double a, double b)**: Divide dos números.
 - **potenciar(double base, double exponente)**: Calcula la potencia de un número.
 - **raizCuadrada(double a)**: Calcula la raíz cuadrada de un número.

3.4. Clase OperadoresBit

- **Descripción:** Implementa operaciones bit a bit entre dos números, como AND, OR, XOR, desplazamientos y operaciones aritméticas simples (suma, resta, multiplicación, división).
- **Atributos:**
 - `private int resultado`: Almacena el resultado de las operaciones.
- **Métodos:**
 - **sumar(int a, int b)**: Suma dos números.
 - **restar(int a, int b)**: Resta dos números.
 - **multiplicar(int a, int b)**: Multiplica dos números.
 - **dividir(int a, int b)**: Divide dos números.
 - **and(int a, int b)**: Realiza una operación AND entre dos números.
 - **or(int a, int b)**: Realiza una operación OR entre dos números.
 - **xort(int a, int b)**: Realiza una operación XOR entre dos números.
 - **desplazarDer(int a, int b)**: Desplaza los bits de un número hacia la derecha.
 - **desplazarIzq(int a, int b)**: Desplaza los bits de un número hacia la izquierda.

3.5. Clase Validaciones

- **Descripción:** Se encarga de capturar las entradas del usuario, validarlas y manejar las posibles excepciones que surgen cuando se ingresan datos no válidos.
- **Métodos:**
 - **tecladoCadena(BufferedReader bln, String label)**: Captura una cadena de texto.
 - **tecladoNumero(BufferedReader bln, String label)**: Captura un número y valida que sea un valor numérico.
 - **pedirDivisor(BufferedReader bln, String label)**: Captura un divisor, validando que no sea cero.
 - **pedirRadicando(BufferedReader bln, String label)**: Captura un número y valida que no sea negativo para el cálculo de raíces.

- **validarEntero(BufferedReader bln, String label):** Captura un número entero y valida que el valor ingresado sea un número entero.

4. Conclusión:

Este proyecto de calculadora es una implementación que combina operadores aritméticos y de bits, siempre teniendo en cuenta la validación de los datos permitiendo que el programa sea fácil de entender y usar para el usuario que ejecute el programa.