Aquí tienes un informe sobre cómo funciona el código proporcionado, que implementa una calculadora basada en agentes utilizando la biblioteca Mesa y una interfaz gráfica con Tkinter.

Informe sobre el Funcionamiento de la Calculadora Basada en Agentes

1. Estructura General

El código está organizado en varias clases que representan diferentes componentes de la calculadora. Utiliza el patrón de diseño basado en agentes, donde cada operación matemática se maneja a través de un agente específico. La interfaz gráfica permite al usuario ingresar expresiones matemáticas y obtener resultados.

2. Clases Principales

2.1 Agentes de Operación

* **OperationAgent**: Clase base para todos los agentes de operación. Define un método handle que debe ser implementado por las subclases.
* **SumAgent**: Realiza la suma de dos operandos.
* **SubtractAgent**: Realiza la resta de dos operandos.
* **MultiplyAgent**: Realiza la multiplicación de dos operandos.
* **DivideAgent**: Realiza la división de dos operandos y maneja la división por cero.
* **PowerAgent**: Realiza la potenciación (elevar un número a otro).

Cada uno de estos agentes implementa el método handle, que toma una lista de operandos y devuelve el resultado correspondiente.

2.2 Agente de Entrada/Salida

* **InputOutputAgent**: Este agente gestiona la entrada del usuario y la salida del resultado. Utiliza una cola (deque) para almacenar las expresiones ingresadas y un método step para procesarlas.
  + **shunting\_yard**: Convierte una expresión infija (como 2 + 3 \* 4) en notación postfija (también conocida como notación polaca inversa), lo que facilita su evaluación respetando la precedencia de las operaciones.
  + **evaluate\_postfix**: Evalúa la expresión en notación postfija utilizando una pila, llamando a los agentes correspondientes para realizar las operaciones.

2.3 Modelo de Calculadora

* **CalculatorModel**: Coordina todos los agentes. Crea instancias de los agentes de operación y del agente de entrada/salida, añadiéndolos a un programador (BaseScheduler) que controla el flujo de ejecución.

3. Interfaz Gráfica

* **CalculatorApp**: Clase que define la interfaz gráfica utilizando Tkinter. Incluye:
  + Un campo de entrada para que el usuario ingrese expresiones matemáticas.
  + Un botón para calcular el resultado.
  + Una etiqueta para mostrar el resultado.

Cuando el usuario hace clic en el botón "Calcular", se llama al método calculate, que:

1. Extrae y tokeniza la expresión ingresada utilizando expresiones regulares para identificar números y operadores.
2. Envía la expresión al agente de entrada/salida.
3. Ejecuta el modelo para procesar la expresión.

4. Ejecución

Al ejecutar el script, se inicializa una ventana Tkinter donde el usuario puede ingresar expresiones matemáticas. El flujo general es:

1. El usuario escribe una expresión en el campo de entrada.
2. Al presionar "Calcular", se tokeniza la expresión y se envía al InputOutputAgent.
3. Este agente convierte la expresión a notación postfija y la evalúa utilizando los agentes correspondientes.
4. El resultado se muestra en la etiqueta designada en la interfaz gráfica.

Conclusión

Este código proporciona una calculadora funcional que utiliza un enfoque basado en agentes para manejar operaciones matemáticas, lo que permite una estructura modular y extensible. La interfaz gráfica facilita su uso, permitiendo a los usuarios realizar cálculos complejos mediante una representación intuitiva de las expresiones matemáticas.