Operations Class

- Purpose of class: The Operations class encapsulates the functionality for simulating various operations on a computer emulator.

load_file

- Purpose of function: Reads program instructions from a file and loads them into memory.
- Input: Filename of the program file.
- Return Value: None.
- Pre-conditions: The file must exist and be accessible.
- Post-conditions: Program instructions are loaded into memory.

execute

- Purpose of function: Executes the loaded program instructions.
- Input: None.
- Return Value: None.
- Pre-conditions: Program instructions must be loaded into memory.
- Post-conditions: Program execution completes or halts, output is displayed.

stop

- Purpose of function: Stops the execution of the program.
- Input: None.
- Return Value: None.
- Pre-conditions: Program execution must be ongoing.
- Post-conditions: Program execution stops.

restart

- Purpose of function: Resets the program state to its initial state.
- Input: None.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Program state is reset, ready for a new execution or loading of program.

read_input

- Purpose of function: Reads input from the user and stores it in the specified memory address.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Input value is stored in memory at the specified address.

write_output

- Purpose of function: Writes output from the memory address to the console.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Output value is displayed on the console.

load_accumulator

- Purpose of function: Loads the value from the specified memory address into the accumulator.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Value from memory is loaded into the accumulator.

store_accumulator

- Purpose of function: Stores the value of the accumulator into the specified memory address.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Value of the accumulator is stored in memory at the specified address.

add

- Purpose of function: Adds the value from the specified memory address to the accumulator.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Value from memory is added to the accumulator.

subtract

- Purpose of function: Subtracts the value from the specified memory address from the accumulator.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Value from memory is subtracted from the accumulator.

divide

- Purpose of function: Divides the value in the accumulator by the value at the specified memory address.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: Value at the specified memory address must not be zero.
- Post-conditions: Value in the accumulator is divided by the value from memory.

multiply

- Purpose of function: Multiplies the value in the accumulator by the value at the specified memory address.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Value in the accumulator is multiplied by the value from memory.

branch

- Purpose of function: Changes the instruction counter to the specified address.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Instruction counter is set to the specified address.

branch_neg

- Purpose of function: Changes the instruction counter to the specified address if the accumulator is negative.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Instruction counter is set to the specified address if the accumulator is negative.

branch_zero

- Purpose of function: Changes the instruction counter to the specified address if the accumulator is zero.
- Input: Memory address.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Instruction counter is set to the specified address if the accumulator is zero.

halt

- Purpose of function: Halts the execution of the program.
- Input: None.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: Program execution is halted.

select_file

- Purpose of function: This function allows the user to select a file using a file dialog, loads the selected file into the emulator's memory, executes the program instructions, and displays a message box indicating successful execution or an error message if an exception occurs.
- Input: None.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: If successful, the program instructions are loaded into memory and executed; otherwise, an error message is displayed.

stop_program

- Purpose of function: This function stops the execution of the program running on the emulator.
- Input: None.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: The execution of the program is halted.

restart_program

- Purpose of function: This function restarts the execution of the program by resetting the emulator's state and loading a new program file.
- Input: None.
- Return Value: None.
- Pre-conditions: None.
- Post-conditions: The emulator's state is reset, and a new program can be loaded and executed.