

UVSimGUI

Attributes:

-master: Tk # The Tkinter root window -uvsim: UVSim # Instance of the UVSim class

Methods:

+ __init__(master: Tk) # Initializes the GUI interface with layout + add tab() # Adds new tab to the GUI (future extensibility) + change_color() # Opens color picker dialog to select a UI color scheme + update_gui_colors() # Applies selected color scheme to the GUI element + on_closing() # Handles window close event for GUI cleanup and app shutdown

Purpose: Provides a graphical interface for controlling UVSim execution Pre-conditions: Requires an initialized Tkinter root window Post-conditions: Displays program interface and allows user interaction with UVSim

UVSim

Attributes:

-memory: list # Memory storage for the program instructions and data -accumulator: int # Stores result of arithmetic operations -instruction_pointer: int # Tracks current instruction address

Methods:

+ __init__(memory: list) # Initializes memory and sets the accumulator and pointer to 0 + load_program(file_path: str) # Loads instructions from a file into memory

+ execute() # Executes the current instruction in memory and moves instruction pointer

- + read_input(address: int, value: int) # Stores user-provided input into the memory at the specified address
- + write_output(address: int) -> int # Returns the value stored at a memory address (for output purposes)
 - + load(address: int) # Loads value from memory at address into accumulator
 - + store(address: int) # Stores current accumulator value into memory at specified address
 - + add(address: int) # Adds value at memory address to accumulator
 - + subtract(address: int) # Subtracts memory value from accumulator
 - + divide(address: int) # Divides accumulator by value at memory address + multiply(address: int) # Multiplies accumulator by value at memory address
 - + branch(address: int) # Sets instruction pointer to specified address (unconditional jump)
 - + branchneg(address: int) # Jumps to address if accumulator is negative
 - + branchzero(address: int) # Jumps to address if accumulator is zero

Purpose: Implements a basic machine simulator Pre-conditions: Program must be properly loaded into memory and instructions must be formatted correctly Post-conditions: Executes instructions and updates memory and accumulator accordingly