**Class Definition Document**

**1. UVSim Class**

The **UVSim** class acts as the main processing unit of the simulator, handling program execution, memory management, and user input processing.

**Attributes**

• **ui**: A reference to the GUI instance.

• **log** (list): Stores execution logs.

• **memSpace** (int): Tracks the next available memory location.

• **counter** (int): Tracks the current instruction being executed.

• **memory** (dict): A dictionary storing 100 memory locations, initialized with "0000".

• **accum** (Accumulator): An instance of the Accumulator class.

• **record** (bool): A flag for logging execution.

**Methods**

**Program Loading**

**fileInputToMemory(inputFile)**

**Purpose**: Reads a program from a file and loads it into memory.

**Inputs**:

• inputFile (str): The path to the input file.

**Returns**:

• bool: True if the file is successfully loaded, False if an error occurs.

**Program Execution**

**wordProcess(step=False)**

**Purpose**: Fetches, decodes, and executes an instruction.

**Inputs**:

• step (bool, optional): If True, pauses execution after each instruction for debugging.

**Returns**:

• None

**process\_input(input\_word)**

**Purpose**: Handles user input for **READ** instructions and stores it in memory.

**Inputs**:

• input\_word (str): User-provided input.

**Returns**:

• None

**stepProgram()**

**Purpose**: Executes a single instruction and updates the execution log.

**Inputs**:

• None

**Returns**:

• None

**Branching Instructions**

**branch(value)**

**Purpose**: Unconditionally jumps to the specified memory location.

**Inputs**:

• value (int): The memory address to jump to.

**Returns**:

• None

**branchneg(value)**

**Purpose**: Jumps to a memory location if the accumulator value is negative.

**Inputs**:

• value (int): The memory address to jump to.

**Returns**:

• None

**branchzero(value)**

**Purpose**: Jumps to a memory location if the accumulator value is zero.

**Inputs**:

• value (int): The memory address to jump to.

**Returns**:

• None

**Memory & Execution Log**

**inspectCurrent()**

**Purpose**: Returns the current execution state of the program.

**Inputs**:

• None

**Returns**:

• dict: A dictionary containing execution details.

**inspectMemory()**

**Purpose**: Prints all memory contents for inspection.

**Inputs**:

• None

**Returns**:

• None

**logDisplay()**

**Purpose**: Displays the execution log.

**Inputs**:

• None

**Returns**:

• None

**Saving & Quitting**

**getAccumulator()**

**Purpose**: Retrieves the current accumulator value.

**Inputs**:

• None

**Returns**:

• int: The current value of the accumulator.

**saveMemory()**

**Purpose**: Saves the current memory state and accumulator value to a file.

**Inputs**:

• None

**Returns**:

• None

**update\_console(message)**

**Purpose**: Updates the GUI console with a given message.

**Inputs**:

• message (str): The message to be displayed.

**Returns**:

• None

**quit()**

**Purpose**: Stops execution and closes the application.

**Inputs**:

• None

**Returns**:

• None

**2. Accumulator Class**

The **Accumulator** class manages arithmetic operations and interacts with memory.

**Methods**

**read(loc, input\_word)**

**Purpose**: Reads user input into memory at a specific location.

**Inputs**:

• loc (int): The memory address where input is stored.

• input\_word (str): The input value.

**Returns**:

• None

**write(loc, sign)**

**Purpose**: Outputs a value from memory to the console.

**Inputs**:

• loc (int): The memory address to read from.

• sign (int): Determines if the value should be displayed with a sign.

**Returns**:

• None

**load(loc, sign)**

**Purpose**: Loads a value from memory into the accumulator.

**Inputs**:

• loc (int): The memory address to load from.

• sign (int): Determines the sign of the loaded value.

**Returns**:

• None

**store(loc, sign)**

**Purpose**: Stores the accumulator’s value in memory.

**Inputs**:

• loc (int): The memory address where the value is stored.

• sign (int): Determines the sign of the stored value.

**Returns**:

• None

**add(loc, sign)**

**Purpose**: Adds a value from memory to the accumulator.

**Inputs**:

• loc (int): The memory address to add from.

• sign (int): Determines the sign of the addition.

**Returns**:

• None

**subtract(loc, sign)**

**Purpose**: Subtracts a memory value from the accumulator.

**Inputs**:

• loc (int): The memory address to subtract from.

• sign (int): Determines the sign of the subtraction.

**Returns**:

• None

**divide(loc, sign)**

**Purpose**: Divides the accumulator by a memory value.

**Inputs**:

• loc (int): The memory address of the divisor.

• sign (int): Determines the sign of the division.

**Returns**:

• None

**multiply(loc, sign)**

**Purpose**: Multiplies the accumulator by a memory value.

**Inputs**:

• loc (int): The memory address of the multiplier.

• sign (int): Determines the sign of the multiplication.

**Returns**:

• None

**3. UVSimUI Class**

The **UVSimUI** class provides a graphical interface for the simulator, allowing users to interact with UVSim visually.

**Components**

• **File Input**: Loads a program from a text file.

• **Console Output**: Displays execution logs and prompts.

• **Memory Table**: Shows all memory locations and their values.

• **Control Buttons**: Start, Step, Save, and Quit the simulation.

• **Accumulator Display**: Shows the current accumulator value.

**Methods**

**file\_handler(instance)**

**Purpose**: Handles file selection and program loading.

**execute\_handler(instance)**

**Purpose**: Runs the program continuously until completion.

**step\_handler(instance)**

**Purpose**: Steps through the program one instruction at a time.

**save\_handler(instance)**

**Purpose**: Saves the memory state to a file.

**quit\_handler(instance)**

**Purpose**: Exits the simulation.

**update\_accumulator(value)**

**Purpose**: Updates the accumulator display.

**refresh\_memory\_table()**

**Purpose**: Refreshes the memory view in the GUI.

**4. MyUVSimApp Class**

This class initializes the application and connects **UVSim** with **UVSimUI**.

**Methods**

**build()**

**Purpose**: Creates the UI and links it to the simulator.

**Returns**:

• None