**1. UVSim Class**

The UVSim class acts as the main **processing unit** of the simulator.

**Attributes**

* ui: A reference to the GUI instance.
* log: A list storing execution logs.
* memSpace: Tracks the next available memory location.
* counter: Tracks the current instruction being executed.
* memory: A dictionary storing 100 memory locations, initialized with "0000".
* accum: An instance of the Accumulator class.
* record: A flag for logging execution.

**Methods**

**Program Loading**

* fileInputToMemory(inputFile):
  + Validates and reads a file line-by-line.
  + Stores each instruction in memory.
  + Updates the console with success/failure messages.

**Program Execution**

* wordProcess(step=False):
  + Fetches an instruction from memory, decodes it, and executes the corresponding operation.
  + If step=True, execution pauses after each instruction for debugging.
* process\_input(input\_word):
  + Handles user input for READ instructions.
  + Stores input in memory and resumes execution.
* stepProgram():
  + Updates the console with the latest execution log.
  + Pauses execution for debugging.

**Branching Instructions**

* branch(value): Unconditionally jumps to the specified memory location.
* branchneg(value): Jumps if the accumulator value is negative.
* branchzero(value): Jumps if the accumulator value is zero.

**Memory & Execution Log**

* inspectCurrent(): Returns the program’s current execution state.
* inspectMemory(): Prints all memory contents.
* logDisplay(): Displays the execution log.

**Saving & Quitting**

* getAccumulator(): Returns the current accumulator value.
* saveMemory(): Saves the current memory state and accumulator value to a file.
* update\_console(message): Updates the GUI console.
* reset(): Resets program to default state.
* quit(): Stops the application and closes the window.

**2. Accumulator Class**

The Accumulator class manages arithmetic operations and interacts with memory.

**Key Methods**

* read(loc, input\_word): Reads user input into memory.
* write(loc, sign): Outputs a value from memory.
* load(loc, sign): Loads a value into the accumulator.
* store(loc, sign): Stores the accumulator’s value in memory.
* add(loc, sign): Adds a value from memory to the accumulator.
* subtract(loc, sign): Subtracts a memory value from the accumulator.
* divide(loc, sign): Divides the accumulator by a memory value.
* multiply(loc, sign): Multiplies the accumulator by a memory value.

**3. UVSimUI Class**

The UVSimUI class provides a graphical interface for the simulator.

**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\_chooser\_handler(instance, selection, touch): Preps file path from file picker selection
* file\_handler(instance): Handles file selection.
* execute\_handler(instance): Runs the program continuously.
* step\_handler(instance): Steps through the program one instruction at a time.
* save\_file\_handler(instance): Saves the memory state.
* quit\_handler(instance): Exits the simulation.
* update\_accumulator(value): Updates the accumulator display.
* refresh\_memory\_table(): Refreshes the memory view.
* make\_reset\_button(): Turns the select file button into an button to reset the program.
* reset\_handler(instance): Resets the GUI to default for additional use.

**4. MyUVSimApp Class**

This class initializes the application and connects UVSim with UVSimUI.

**Methods**

* build(): Creates the UI and links it to the simulator.
* Runs the application when executed.