purpose of the class, purpose of each function, description of any input parameters for the function, description of return value (if any), pre- and post-conditions for the function

Brandon Woodruff

GUI.PY

EXECUTEPROGRAM.PY

MAIN.PY

MEMORY.PY

UVSIM.PY

PROCESS_PROGRAM.PY

GUI class
Function: creates a simple GUI using tkinter
Input: UVsim class
Methods:
limit_code_lines():
purpose: limit the number of lines in the code block
input: event
open_file():
purpose: search directories and choose a txt file.
input: n/a
save_code_block():
purpose: save the text inputed into the textblock
input: n/a
run_code_block():
purpose: runs program with the selected file as the input
input: n/a
configure_color_scheme():
purpose:allows user to change GUI's colors
input: n/a
output():
purpose: display results of the program
input: output
read():
purpose: Read a word from the keyboard into memory.
input: value from keyboard
write():
purpose: write a word from memory to gui
input: n/a

Brandon Woodruff

**SIMPLEGUI CLASS**

-main: GUI window
-sim: object
-label_1: object
-label_2: object
-label_3: object
-file_menu: object
-menu_bar: object
-operations_text: object
-code_text: object
-_program: list
-widgets: object

constructor()

limit_code_lines()

open_file()

save_code_block()

run_code_block()

configure_color_scheme()

output()

read()

write()

**MAIN**

my_Sim = UVSim()

my_memory = Memory(100)

my_gui = SimpleGUI(my_sim, my_memory)

my_gui.main.mainloop()

Main:
Function: to run the code
Input: Program: n/a
Methods: n/a

Brandon Woodruff

**EXECUTE CLASS**

execute_program()

execute_program class
Function: to execute the program
Input: UVSim class
Methods:
execute():
purpose: To actually execute, To call the other classes as needed.
input: GUI

Brandon Woodruff

**MEMORY CLASS**

-self._registers: int

constructor()

load()

store()

load_program()

truncate()

len()

clear():

Memory class
Function: handle memory-related operations
Input: n/a
Methods:
load():
purpose: load a word from memory into the accumulator
input: operand
store():
purpose: store a word from the acculator into memory
input: operand, accumulator
load_program():
purpose: load a program into memory
input: program
truncate():
purpose: truncate value to four digits to avoid overflow when needed
input: value
len():
purpose: return length of registers list
input: n/a
clear():
purpose: clear memory
input: n/a

Brandon Woodruff

**UVSIM CLASS**

-_accumulator: int
-_pc: int
-_operand: int
-_op: int

Constructor()

add()

subtract()

divide()

multiply()

branch()

branch_neg()

branch_zero()

halt()

get_accumulator()

set_accumulator()

set_operand()

get_operand()

set_memory()

set_program()

get_program()

get_pc()

UVSim class
Function: UVSim is a simulation that can interpret a machine language called BasicML.
Input:n/a
Methods:
add():
purpose: to add two numbers
input: two values to add together
subtract():
purpose: to subtract two numbers
input: two values to subtract
divide():
purpose: to divide a number from another
input: a value, and a value to divide that number by
multiply():
purpose: to multiply two numbers
input: two values to multiply
branch():
purpose: to move to a specific location in memory
input: a value
branchNeg():
purpose: to move to a specific location if the number is negative.
input: accumulator value, operand value, and pc value
branchZero():
purpose: to move to a specific location if the number is negative
input:
halt:
purpose: to end the program immediately
input: n/a
get_accumulator():
purpose: gets accumulator
input: n/a
set_accumulator() :
purpose: sets accumulator
input: n/a
set_operand():
purpose: sets operand
input: n/a
get_operand():
purpose: gets operand
input: n/a
set_memory():
purpose: sets memory
input: n/a
set_program():
purpose: sets program
input: n/a
get_program():
purpose: gets program
input: n/a
get_pc():
purpose: gets pc
input: n/a

Brandon Woodruff

**PROCESS**

read_txt()

Process class
Function: This class reads a program from a data source.
It also loads the program into memory
Input: n/a
Methods:
read_txt():
purpose: read a program from a file and return it as a list of instructions and data values
input: file_path

Brandon Woodruff