Facade for entire program
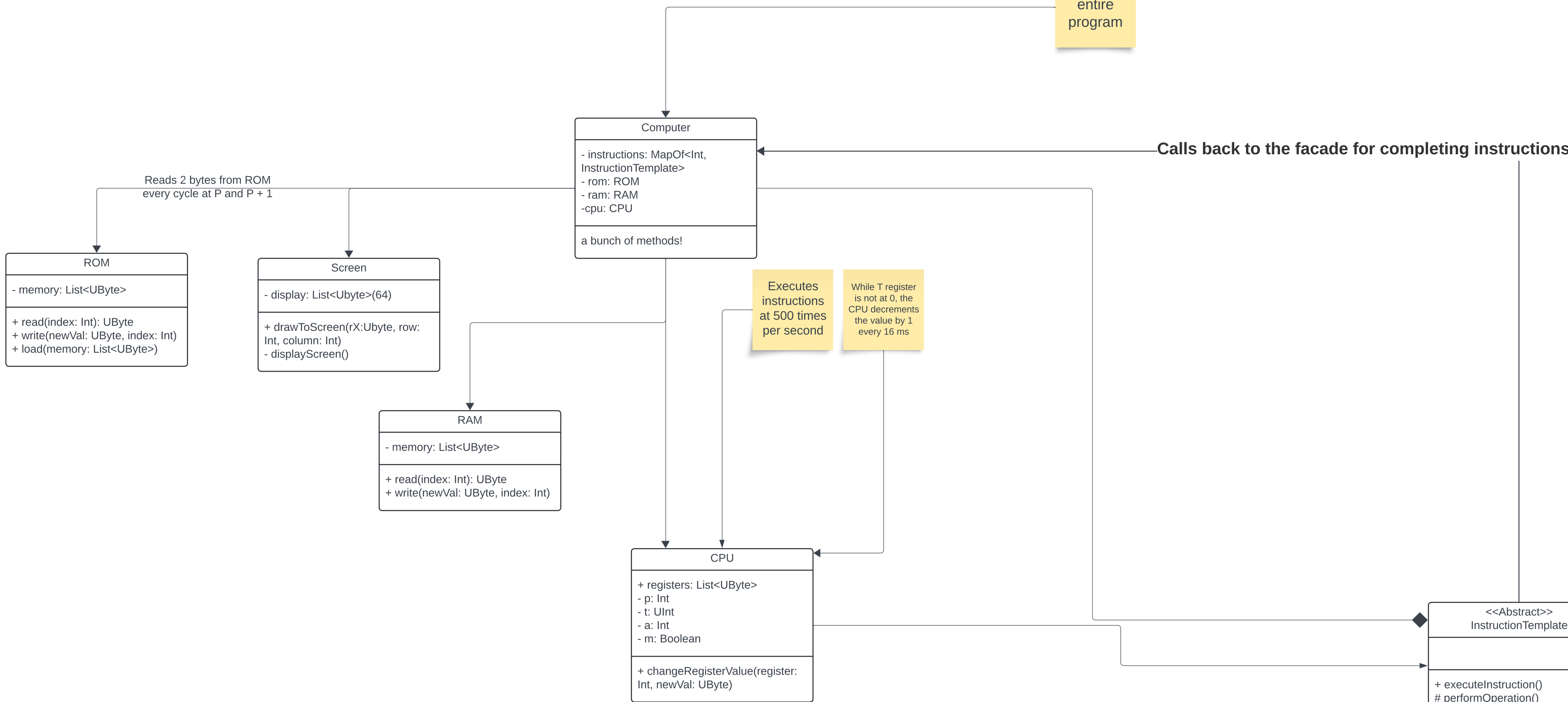
Computer
- instructions: MapOf<Int, InstructionTemplate>
- rom: ROM
- ram: RAM
-cpu: CPU

a bunch of methods!

Calls back to the facade for completing instructions

Reads 2 bytes from ROM every cycle at P and P + 1

ROM
- memory: List<UByte>

+ read(index: Int): UByte
+ write(newVal: UByte, index: Int)
+ load(memory: List<UByte>)

Screen
- display: List<Ubyte>(64)

+ drawToScreen(rX:Ubyte, row: Int, column: Int)
- displayScreen()

Executes instructions at 500 times per second

While T register is not at 0, the CPU decrements the value by 1 every 16 ms

RAM
- memory: List<UByte>

+ read(index: Int): UByte
+ write(newVal: UByte, index: Int)

CPU
+ registers: List<UByte>
- p: Int
- t: UInt
- a: Int
- m: Boolean

+ changeRegisterValue(register: Int, newVal: UByte)

<<Abstract>> InstructionTemplate

+ executeInstruction()
# performOperation()
# incrementCounter()

NOTE: The template automatically adds 2 to the program counter. If an instruction does not want to do this, it overrides the method and doesn't do anything.

Store

Add

Sub

Read

Write

Jump

ReadKeyboard

SwitchMemory

SkipEqual

SkipNotEqual

SetA

SetT

ReadT

ConvertToBase10

ConvertByteToAscii

Draw