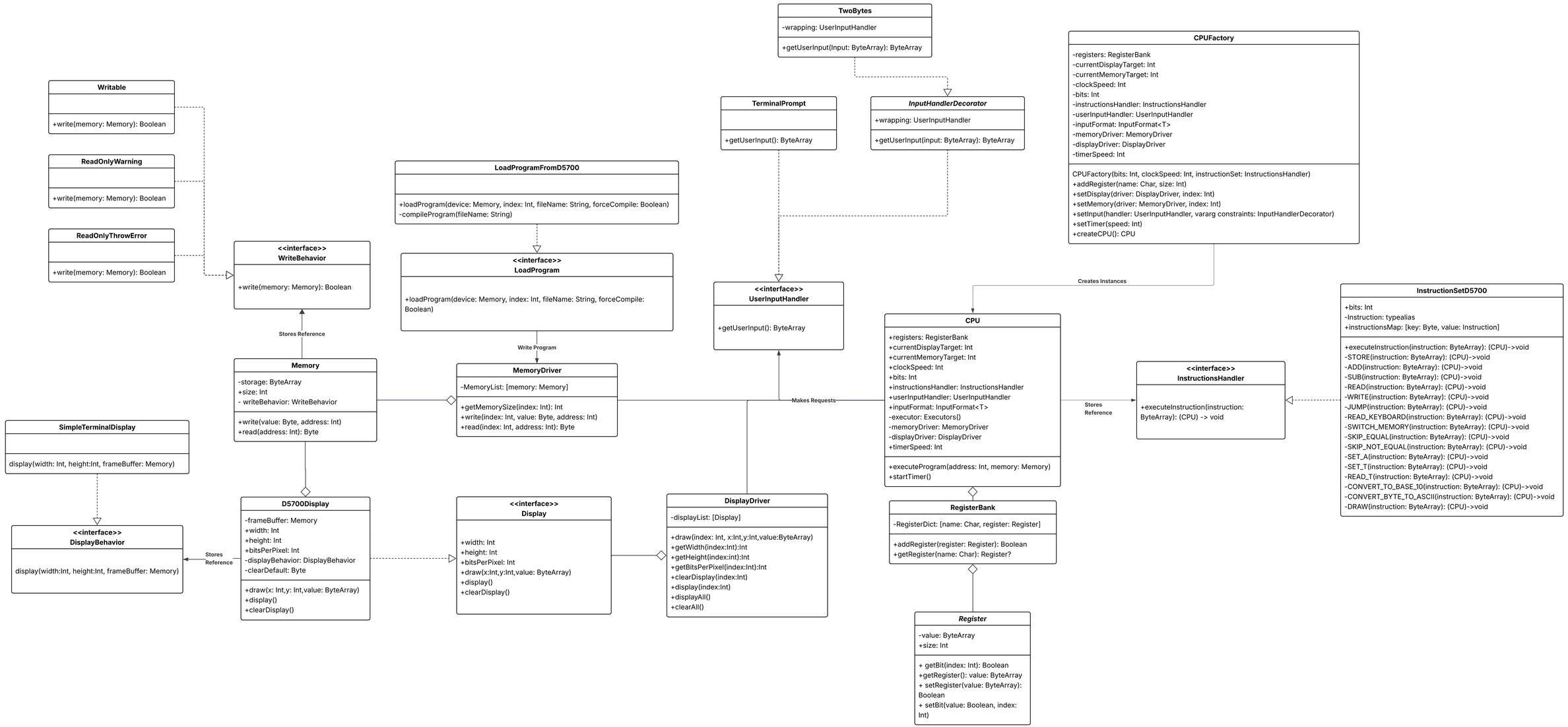
TwoBytes
-wrapping: UserInputHandler
+getUserInput(Input: ByteArray): ByteArray
CPUFactory
-registers: RegisterBank
-currentDisplayTarget: Int
-currentMemoryTarget: Int
-clockSpeed: Int
-bits: Int
-instructionsHandler: InstructionsHandler
-userInputHandler: UserInputHandler
-inputFormat: InputFormat<T>
-memoryDriver: MemoryDriver
-displayDriver: DisplayDriver
-timerSpeed: Int
CPUFactory(bits: Int, clockSpeed: Int, instructionSet: InstructionsHandler)
+addRegister(name: Char, size: Int)
+setDisplay(driver: DisplayDriver, index: Int)
+setMemory(driver: MemoryDriver, index: Int)
+setInput(handler: UserInputHandler, vararg constraints: InputHandlerDecorator)
+setTimer(speed: Int)
+createCPU(): CPU
Writable
+write(memory: Memory): Boolean
ReadOnlyWarning
+write(memory: Memory): Boolean
ReadOnlyThrowError
+write(memory: Memory): Boolean
TerminalPrompt
+getUserInput(): ByteArray
InputHandlerDecorator
+wrapping: UserInputHandler
+getUserInput(input: ByteArray): ByteArray
LoadProgramFromD5700
+loadProgram(device: Memory, index: Int, fileName: String, forceCompile: Boolean)
-compileProgram(fileName: String)
<<interface>>
WriteBehavior
+write(memory: Memory): Boolean
<<interface>>
LoadProgram
+loadProgram(device: Memory, index: Int, fileName: String, forceCompile: Boolean)
<<interface>>
UserInputHandler
+getUserInput(): ByteArray
Creates Instances
CPU
+registers: RegisterBank
+currentDisplayTarget: Int
+currentMemoryTarget: Int
+clockSpeed: Int
+bits: Int
+instructionsHandler: InstructionsHandler
+userInputHandler: UserInputHandler
+inputFormat: InputFormat<T>
-executor: Executors()
-memoryDriver: MemoryDriver
-displayDriver: DisplayDriver
+timerSpeed: Int
+executeProgram(address: Int, memory: Memory)
+startTimer()
InstructionSetD5700
+bits: Int
-Instruction: typealias
+instructionsMap: [key: Byte, value: Instruction]
+executeInstruction(instruction: ByteArray): (CPU)->void
-STORE(instruction: ByteArray): (CPU)->void
-ADD(instruction: ByteArray): (CPU)->void
-SUB(instruction: ByteArray): (CPU)->void
-READ(instruction: ByteArray): (CPU)->void
-WRITE(instruction: ByteArray): (CPU)->void
-JUMP(instruction: ByteArray): (CPU)->void
-READ_KEYBOARD(instruction: ByteArray): (CPU)->void
-SWITCH_MEMORY(instruction: ByteArray): (CPU)->void
-SKIP_EQUAL(instruction: ByteArray): (CPU)->void
-SKIP_NOT_EQUAL(instruction: ByteArray): (CPU)->void
-SET_A(instruction: ByteArray): (CPU)->void
-SET_T(instruction: ByteArray): (CPU)->void
-READ_T(instruction: ByteArray): (CPU)->void
-CONVERT_TO_BASE_10(instruction: ByteArray): (CPU)->void
-CONVERT_BYTE_TO_ASCII(instruction: ByteArray): (CPU)->void
-DRAW(instruction: ByteArray): (CPU)->void
Stores Reference
<<interface>>
InstructionsHandler
+executeInstruction(instruction: ByteArray): (CPU) -> void
Stores Reference
Memory
-storage: ByteArray
+size: Int
- writeBehavior: WriteBehavior
+write(value: Byte, address: Int)
+read(address: Int): Byte
Write Program
MemoryDriver
-MemoryList: [memory: Memory]
+getMemorySize(index: Int): Int
+write(index: Int, value: Byte, address: Int)
+read(index: Int, address: Int): Byte
Makes Requests
SimpleTerminalDisplay
display(width: Int, height:Int, frameBuffer: Memory)
<<interface>>
DisplayBehavior
display(width:Int, height:Int, frameBuffer: Memory)
Stores Reference
D5700Display
-frameBuffer: Memory
+width: Int
+height: Int
+bitsPerPixel: Int
-displayBehavior: DisplayBehavior
-clearDefault: Byte
+draw(x: Int,y: Int,value: ByteArray)
+display()
+clearDisplay()
<<interface>>
Display
+width: Int
+height: Int
+bitsPerPixel: Int
+draw(x:Int,y:Int,value: ByteArray)
+display()
+clearDisplay()
DisplayDriver
-displayList: [Display]
+draw(index: Int, x:Int,y:Int,value:ByteArray)
+getWidth(index:Int):Int
+getHeight(index:int):Int
+getBitsPerPixel(index:Int):Int
+clearDisplay(index:Int)
+display(index:Int)
+displayAll()
+clearAll()
RegisterBank
-RegisterDict: [name: Char, register: Register]
+addRegister(register: Register): Boolean
+getRegister(name: Char): Register?
Register
-value: ByteArray
+size: Int
+ getBit(index: Int): Boolean
+getRegister(): value: ByteArray
+ setRegister(value: ByteArray): Boolean
+ setBit(value: Boolean, index: Int)