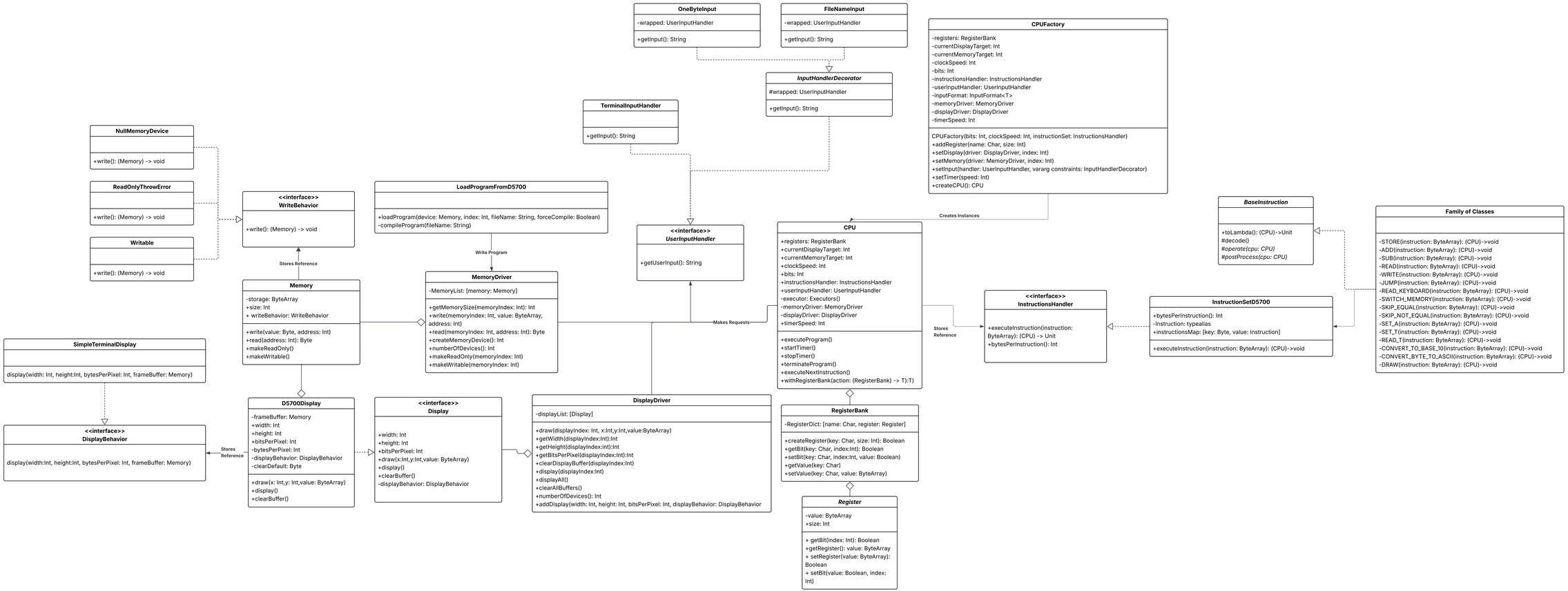
OneByteInput
-wrapped: UserInputHandler
+getInput(): String
FileNameInput
-wrapped: UserInputHandler
+getInput(): String
InputHandlerDecorator
#wrapped: UserInputHandler
+getInput(): String
CPUFactory
-registers: RegisterBank
-currentDisplayTarget: Int
-currentMemoryTarget: Int
-clockSpeed: Int
-bits: 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
TerminalInputHandler
+getInput(): String
NullMemoryDevice
+write(): (Memory) -> void
ReadOnlyThrowError
+write(): (Memory) -> void
Writable
+write(): (Memory) -> void
<<interface>>
WriteBehavior
+write(): (Memory) -> void
LoadProgramFromD5700
+loadProgram(device: Memory, index: Int, fileName: String, forceCompile: Boolean)
-compileProgram(fileName: String)
Write Program
Stores Reference
Memory
-storage: ByteArray
+size: Int
+ writeBehavior: WriteBehavior
+write(value: Byte, address: Int)
+read(address: Int): Byte
+makeReadOnly()
+makeWritable()
MemoryDriver
-MemoryList: [memory: Memory]
+getMemorySize(memoryIndex: Int): Int
+write(memoryIndex: Int, value: ByteArray, address: Int)
+read(memoryIndex: Int, address: Int): Byte
+createMemoryDevice(): Int
+numberOfDevices(): Int
+makeReadOnly(memoryIndex: Int)
+makeWritable(memoryIndex: Int)
<<interface>>
UserInputHandler
+getUserInput(): String
Makes Requests
CPU
+registers: RegisterBank
+currentDisplayTarget: Int
+currentMemoryTarget: Int
+clockSpeed: Int
+bits: Int
+instructionsHandler: InstructionsHandler
+userInputHandler: UserInputHandler
-executor: Executors()
-memoryDriver: MemoryDriver
-displayDriver: DisplayDriver
+timerSpeed: Int
+executeProgram()
+startTimer()
+stopTimer()
+terminateProgram()
+executeNextInstruction()
+withRegisterBank(action: (RegisterBank) -> T):T)
Creates Instances
Stores Reference
<<interface>>
InstructionsHandler
+executeInstruction(instruction: ByteArray): (CPU) -> Unit
+bytesPerInstruction(): Int
BaseInstruction
+toLambda(): (CPU)->Unit
#decode()
#operate(cpu: CPU)
#postProcess(cpu: CPU)
Family of Classes
-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
InstructionSetD5700
+bytesPerInstruction(): Int
-Instruction: typealias
+instructionsMap: [key: Byte, value: Instruction]
+executeInstruction(instruction: ByteArray): (CPU)->void
SimpleTerminalDisplay
display(width: Int, height:Int, bytesPerPixel: Int, frameBuffer: Memory)
<<interface>>
DisplayBehavior
display(width:Int, height:Int, bytesPerPixel: Int, frameBuffer: Memory)
Stores Reference
D5700Display
-frameBuffer: Memory
+width: Int
+height: Int
+bitsPerPixel: Int
-bytesPerPixel: Int
-displayBehavior: DisplayBehavior
-clearDefault: Byte
+draw(x: Int,y: Int,value: ByteArray)
+display()
+clearBuffer()
<<interface>>
Display
+width: Int
+height: Int
+bitsPerPixel: Int
+draw(x:Int,y:Int,value: ByteArray)
+display()
+clearBuffer()
-displayBehavior: DisplayBehavior
DisplayDriver
-displayList: [Display]
+draw(displayIndex: Int, x:Int,y:Int,value:ByteArray)
+getWidth(displayIndex:Int):Int
+getHeight(displayIndex:Int):Int
+getBitsPerPixel(displayIndex:Int):Int
+clearDisplayBuffer(displayIndex:Int)
+display(displayIndex:Int)
+displayAll()
+clearAllBuffers()
+numberOfDevices(): Int
+addDisplay(width: Int, height: Int, bitsPerPixel: Int, displayBehavior: DisplayBehavior
RegisterBank
-RegisterDict: [name: Char, register: Register]
+createRegister(key: Char, size: Int): Boolean
+getBit(key: Char, index:Int): Boolean
+setBit(key: Char, index:Int, value: Boolean)
+getValue(key: Char)
+setValue(key: Char, value: ByteArray)
Register
-value: ByteArray
+size: Int
+ getBit(index: Int): Boolean
+getRegister(): value: ByteArray
+ setRegister(value: ByteArray): Boolean
+ setBit(value: Boolean, index: Int)