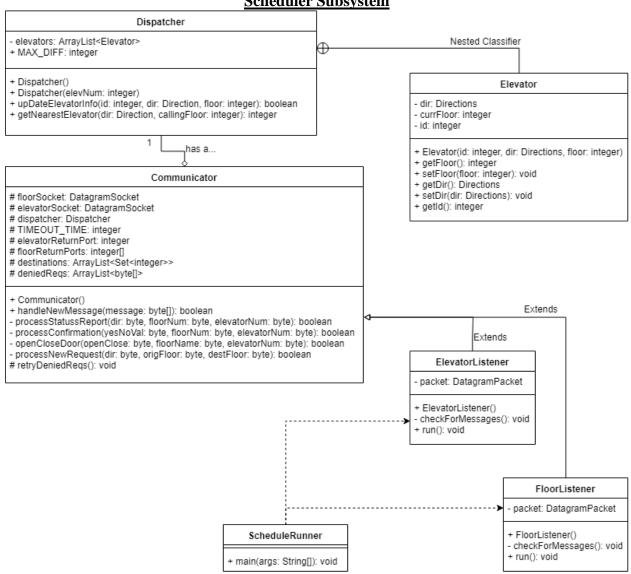
Scheduler Subsystem



Elevator Subsystem Elevator ElevatorMotor + MAX_FLOOR: int elv: Flevator 1 + MIN_FLOOR: int + MAX_SERVICE_QUEUE_CAPACITY: int currentElvState: ElevatorState - previousElvState: ElevatorState - elevNum: Integer has a. - isDoorOpen: boolean + ElevatorMotor(elevator: Elevator) - currFloorPosition: Integer + run(): void - floorDestionation: Integer - move(): void has. serviceScheduleQueue: PriorityBlockingQueue<Integer> serviceFloor(): void elevatorPassengerButtons: Set<Integer> - status: Direction - motor: ElevatorMotor - elevReciever: ElevatorReciever floorComparator: Comparator<Integer> <<enumeration>> Elevator State + Elevator(elvNumber: Integer, elvRecieve: ElevatorReciever) STANDBY + updateFloorToService(): boolean DOOR_CLOSE DOOR_OPEN + updateDirection(): void + canServiceCall(floorRequested: integer): boolean MOVE + moveUp(): void STOP + moveDown(): void + openDoors(): void + closeDoors(): void + removeFromPassengerButtons(floor: int): void + addToPassengerButtons(floor: int): void + addToServiceQueue(floor: int): void + add to Service Queue(): int + pollService Queue(): int + generateDoorOpenMsg(): byte[] + generateDoorCloseMsg(): byte[] + generateAcceptMsg(floorDest. int): byte[] has + generateDeclineMsg(floorDest: int): byte[] ~ generateStatusMsg(): byte[] ~ generateOpenMsg(): byte[] ~ generateCloseMsg(): byte[] + getElevNumber(): Integer + getElevatorReciever(): ElevatorReciever + getCurrFloorPosition(): Integer ElevatorReciever + getFloorDestionation(): Integer NUMBER_OF_ELEVATORS: int
 elevators: List<Elevators> + getServiceScheduleQueue(): PriorityBlockingQueue<Integer> + isPriorityQueueEmpty(): boolean - schedulerSocket: DatagramSocket + getElevatorPassengerButtons(): Set<Integer> ~ messagePort: int + getStatus(): Directions + ElevatorReciever() + recieverCommunicationLoop(): void + processSchedulerMsg(packet: DatagramPacket): void + addFloorToService(elevatorNumber: Integer, floorL Integer): void + sendMessage(msg: byte[]): void + sendResponse(msg: byte[], port: int): void + getElevators(): List<Elevator> + main(args: String[]): void + closeSocket(): void

Floor Subsystem

Floor

- level: int
- isDoorOpen: boolean
- isUpButtonPressed: boolean
- isDownButtonPressed: boolean
- isUpDirectionLampOn: boolean
- isDownDirectionLampOn: boolean
- floorDestinationButtonPressed: boolean
- + Floor(level: int)
- + addFloorButtonPressed(floor: int): void
- + removeFloorButtonPressed(floor: int): void
- + getFloorDestinationButtonPressed(): List<Integer>
- + openDoor(): void
- + closeDoor(): void
- + getLevel(): int
- + isDoorOpen(): boolean
- + setDoorOpen(isDoorOpen: boolean): void
- + isUpButtonPressed(): boolean
- + setUpButtonPressed(isUpButtonPressed: boolean): void
- + isDownButtonPressed(): boolean
- + setDownButtonPressed(isDownButtonPressed: boolean): void
- + isUpDirectionLampOn(): boolean
- + setUpDirectionLamp(isUpDirectionLampOn: boolean): void
- + isDownDirectionLampOn(): boolean
- + setDownDirectionLampOne(isDownDirectionLampOn: boolean): void
- + setLevel(level: int): void

has.

Floor System

- floors: List<Floors>
- systemFile: SystemFile
- que: Queue<Message>
- ~ datagramSocket: DatagramSocket
- STARTING_HOUR: int
- STARTING_MINUTE: int
- + FloorSystem()
- + main(args: String[]): void
- + startFloorSchedule(): void
- + schedulerCommunicationLoop(): void
- + getFloorObjectiveByLevel(floorLevel: int): Floor
- + getFloors(): List<Floor>
 printOutFloorInformation(floor: Floor, action: String): void
- + getQue(): Queue<Message>

Constants

- + ELEVATOR_PORT: int
- + FLOOR_PORT: int
- + MESSAGE_LENGTH: int
- + NUMBER_OF_ELEVATORS: int
- + LOWEST_FLOOR: int
- + HIGHEST_FLOOR: int + NUMBER_OF_FLOORS: int
- + ERROR: byte
- + NEW REQUEST FROM FLOOR: byte
- + REQUEST_RECEIVED: byte
- + OPEN_CLOSE_DOOR: byte
- + NEW_ELEVATOR_DESTINATION: byte
- + ELEVATOR_INFO_REQUEST: byte
- + CONFIRM_VOL_DESTINATION: byte + STATUS_REPORT: byte
- + YES: byte
- + NO: byte
- + OPEN: byte
- + CLOSE: byte
- + MANDATORY: byte
- + VOLUNTARY: byte
- + ELEVATOR_TRAVEL_SPEED_MS: byte
- Constants()

SystemFile

- + FILEPATH: String + FILENAME1: String - filename: String
- + SystemFile(filename: String)
- validateFloorRange(floor: int): boolean
- + readValidateAndCreatMessages(): void
- validateLine(lineInfo: String): boolean
- readFile(): ArrayList<String>
- + testReadFile(): ArrayList<String>
- + testValidateLine(lineInfo: String): boolean

Resources

<<enumeration>> Directions

UP DOWN STANDBY

- + isOpposite(dir1: Directions, dir2: Directions): boolean
- + getDirByInt(val: int): Directions
- + getIntByDir(dir: Directions): int

Message

- time: LocalTime
- startingFloor: int
- direction: Directions
- destinationFloor: int
- messageQueue: Queue<Message>
- ELEMENTS_IN_MESSAGE: int
- + Message(time: LocalTime, startingFloor: int, direction: Direction, destinationFloor: int)
- + getTime(): LocalTime
- + getStartingFloor(): int
- + getDirection(): Directions
- + getDestinationFloor(): int
- + getMessageQueue(): Queue<Message>
- + toString(): String