



## Floor

- level: int
- isDoorOpen: boolean
- isUpButtonPressed: boolean
- isDownButtonPressed: boolean
- isUpDirectionLampOn: boolean
- isDownDirectionLampOn: boolean
- floorDestinationButtonPressed: boolean
- + Floor(level: int)
- + addFloorButtonPressed(floor: int): void
- + removeFloorButtonPressed(illoor: 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

#### Temp

+ main(args: String[])

# Floor System

- floors: List<Floors>

has

- 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
- + MANDATÓRY: 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

## <<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