«enumeration»
ELEVATOR_BUTTON

UP
DOWN
INSIDE
START_MOVE
ARRIVAL

C Thread

«exception»
InterruptedException

«interface»
Runnable

C Main
- main(args:String[]): void

«interface»
ElevatorState

entry(context:Elevator): void
floorRequest(context:Elevator): void
arrivedAtFloor(context:Elevator): void
destinationRequest(context:Elevator): void
arrivedAtDestination(context:Elevator): void
doorsClosed(context:Elevator): void
displayState(): void
toString(): String

C ElevatorEvent

- time: String {readOnly}
- floor: int {readOnly}
- button: ELEVATOR_BUTTON {readOnly}
- car_button: int{readOnly}

- ElevatorEvent(time: String, floor: int, button: ELEVATOR_BUTTON, car_button: int)
- toString(): String
- getTime(): String
- getFloor(): int
- getButton(): ELEVATOR_BUTTON
- getCar_button(): int

C Floor

upLampOn: boolean
downLampOn: boolean
elevatorArrived: boolean
- eventQueue:EventQueue {readOnly}

- Floor(name: String, eventQueue: EventQueue)
- processFile(file: File): void
- processInput(input: String): void
- run(): void

C Elevator

- TIME_PER_FLOOR: long {readOnly}
- DOOR_OPERATION_TIME: long {readOnly}
- currentFloor: int
- elevatorId: int {readOnly}
- eventQueue: EventQueue {readOnly}
- currentEvent: ElevatorEvent
- doorsOpen: boolean
- states: Map<String, ElevatorState> {readOnly}
- currentState: ElevatorState

- Elevator(eventQueue: EventQueue, elevatorId: int)
- run(): void
- getElevatorId(): int
- floorRequested(): void
- arrivedAtFloor(): void
- destinationRequest(): void
- arrivedAtDestination(): void
- doorsClosed(): void
- setCurrentState(nextState: String): void
- getCurrentEvent(): ElevatorEvent
- processEvent(event: ElevatorEvent): void
- moveToFloor(floor: int) throws InterruptedException : void
- notifySchedulerOfArrival(): void
- openDoors() throws InterruptedException: void
- closeDoors() throws InterruptedException : void

C Scheduler

- schedulerState: Enum
- state: SchedulerState
- eventQueue:EventQueue {readOnly}
- floorRequestToBeProcessed: ElevatorEvent
- processedRequest:ElevatorEvent

- Scheduler(eventQueue: EventQueue)
- run():void
- readFloorRequest(): void
- processFloorRequest(): void
- sendElevatorRequest(): void

C Idle

C MovingToFloor

C MovingToDestination

C Loading

C Unloading

C EventQueue

- floorRequest:ArrayDeque<ElevatorEvent> {readOnly}
- elevatorRequest: ArrayDeque<ElevatorEvent> {readOnly}
- processedEvents: int
- maxEvents: int
- maxQueue: int {readOnly}

- EventQueue()
- mutex(): void
- synchronized getFloorRequest(): ElevatorEvent
- synchronized setFloorRequest(event: ElevatorEvent): void
- synchronized setElevatorRequest(event: ElevatorEvent): void
- synchronized getElevatorRequest(): ElevatorEvent
- synchronized elevatorArrived(): void