

<<enumeration>> Direction
UP DOWN STOPPED

<<enumeration>> SystemRequestType
REGISTER_ELEVATOR_CONTROLLER NEW_PRIMARY_REQUEST IS_STOP_REQUIRED PROCESS_REQUESTS_AT_CURRENT_F PROCESS_COMPLETED_REQUESTS SET_FLOOR_DIRECTION_LAMPS SET_FLOOR_LAMPS ADD_NEW_REQUEST STATUS_REQUEST ELEVATOR_SHUTDOWN_REQUEST

<<enumeration>> RequestStatus
PENDING PASSENGER_PICKED_UP

<<enumeration>> FaultType
NO_FAULT DOOR_FAULT FLOOR_TIMER_FAULT

Constants
+ <u>NUMBER_OF_FLOORS</u> : int + <u>NUMBER_OF_ELEVATORS</u> : int + <u>SCHEDULER_PORT</u> : int + <u>SCHEDULER_PORT_2</u> : int + <u>FLOOR_CONTROLLER_PORT</u> : int + <u>BASE_MOVE_TIME</u> : int + <u>INCREMENTAL_MOVE_TIME</u> : int + <u>LOADING_TIME</u> : int

>

FLOOR

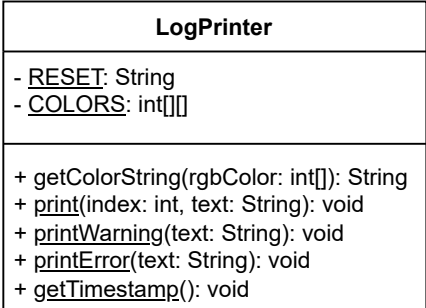
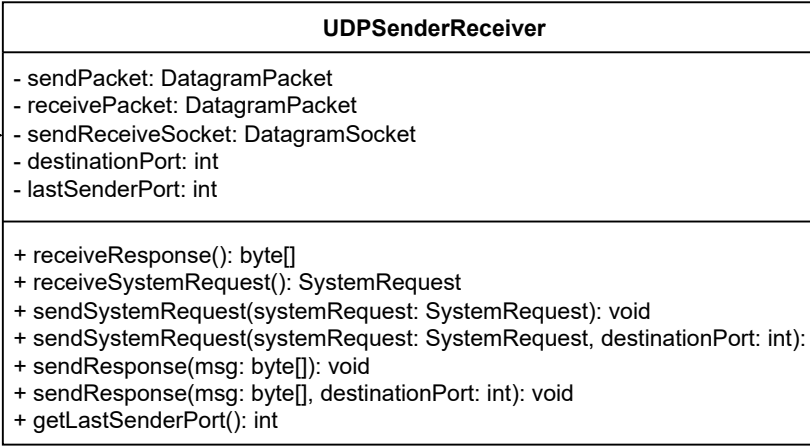
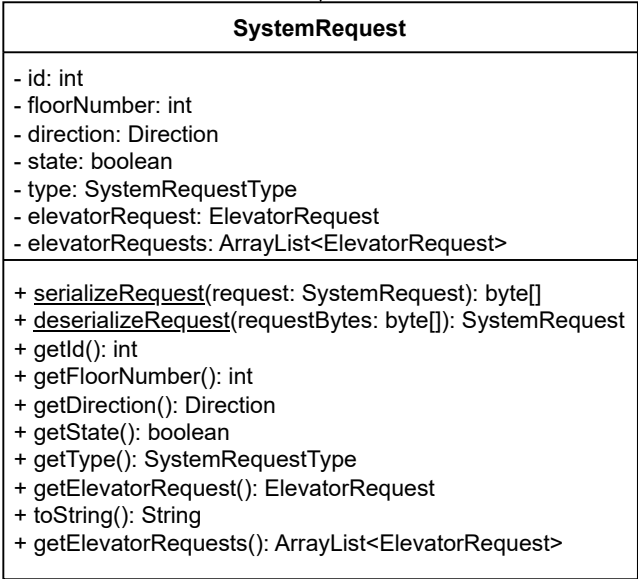
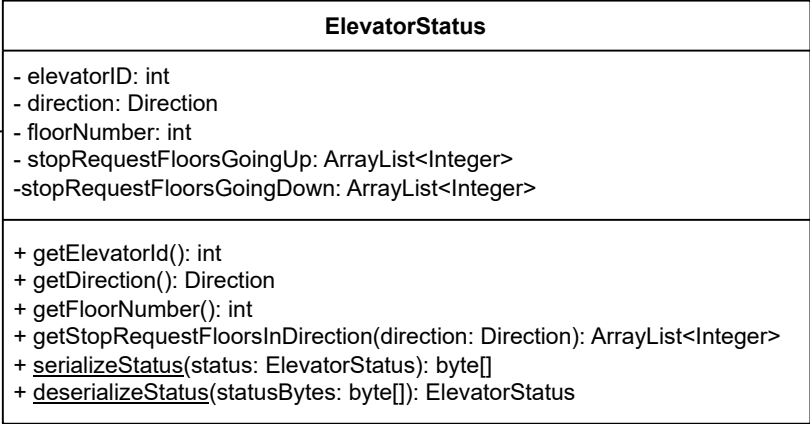
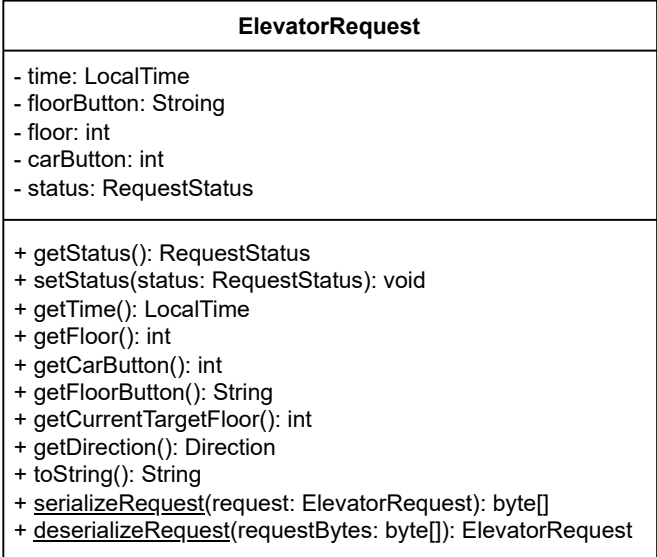
>

ED_UP

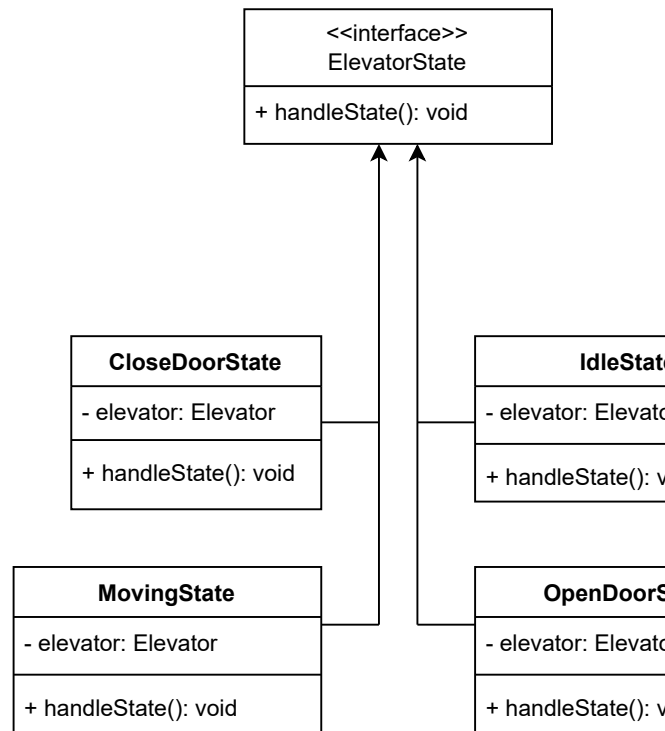
>>

AULT

t



void



e
or
void

State
or
void

Elevator
<div>- subsystem: ElevatorSubsystem - primaryRequest: ElevatorRequest - currentState: ElevatorState - direction: Direction - floorNumber: int - motorRunning: boolean - doorOpen: boolean - elevatorId: int - senderReceiver: UDPSenderReceiver - time: double - deadline: double</div>
<div>+ getNextFloorNumber(): int + setFloorNumberToNextFloor(): void + getCurrentState(): ElevatorState + setCurrentState(currentState: ElevatorState): void + setDoorOpen(doorOpen: boolean): void + getElevatorId(): int + getDirection(): Direction + getFloorNumber(): int + isMotorRunning(): boolean + setMotorRunning(motorRunning: boolean): void + isDoorOpen(): boolean + getSubsystem(): ElevatorSubsystem + setPrimaryRequest(primaryRequest: ElevatorRequest): void + setDirection(direction: Direction): void + getPrimaryRequest(): ElevatorRequest + getSenderReceiver(): UDPSenderReceiver + run(): void + getTime(): double + setTime(var: double): void + getDeadline(): double + setDeadline(var: double): void + synchDeadline(): void + hasTransientFault(): void + hasHardFault(): void</div>

ElevatorController
<div>- senderReceiver: UDPSenderReceiver - elevator: Elevator - subsystem: ElevatorSubsystem</div>
<div>+ addNewRequest(request: ElevatorRequest): void + processCompletedRequests(floorNumber: int, direction: Direction): void + processRequestsAtCurrentFloor(floorNumber: int, direction: Direction): void + receiveNewPrimaryRequest(): ElevatorRequest + isStopRequiredForFloor(nextFloorNumber: int, direction: Direction): boolean - getElevatorStatus(): ElevatorStatus + run(): void</div>

ElevatorSubsystem
<div>- downRequests: ArrayList<ElevatorRequest> - upRequests: ArrayList<ElevatorRequest> - hasWaitingRequests: boolean + elevatorLamps: ArrayList<Boolean> - <u>elevatorId</u>: int</div>
<div>+ getStopRequestFloorsInDirection(direction: Direction): ArrayList<Integer> + addNewRequest(request: ElevatorRequest): void + hasWaitingRequests(): boolean + setElevatorLamps(floorNumber: int, lampState: boolean): void + processCompletedRequests(floorNumber: int, direction: Direction): void + processRequestsAtCurrentFloor(floorNumber: int, direction: Direction): void + receiveNewPrimaryRequest(): ElevatorRequest + isStopRequiredForFloor(nextFloorNumber: int, direction: Direction): boolean + hasFault(faultType: FaultType, floorNumber: int): boolean + getWaitingRequests(): ArrayList<ElevatorRequest></div>



oid ean

> oid ean

FloorController
<div>- floorSubsystem: FloorSubsystem</div> <div>- receiver: UDPSenderReceiver</div>
<div>+ run(): void</div>

Floor
<div>- floorNumber: int</div> <div>- downRequests: ArrayList<ElevatorRequest></div> <div>- upRequests: ArrayList<ElevatorRequest></div> <div>- floorLampUp: boolean</div> <div>- floorLampDown: boolean</div> <div>- directionLamps: boolean[][]</div> <div>- isFirstFloor: boolean</div> <div>- isLastFloor: boolean</div>
<div>+ addRequest(event: ElevatorRequest): void</div> <div>+ checkForRequests(): ElevatorRequest</div> <div>+ setDirectionLamp(elevatorId: int, direction: Direction, state: boolean): void</div> <div>+ setFloorLamp(direction: Direction, state: boolean): void</div> <div>+ getUpRequests(): Collection<Object></div> <div>+ getDownRequests(): Collection<Object></div> <div>+ getDirectionLamps(): boolean[][]</div> <div>+ getFloorLampUp(): boolean</div> <div>+ getFloorLampDown(): boolean</div>



FloorSubsystem
<div>- sender: UDPSenderReceiver</div> <div>- numberOfFloors: int</div> <div>- floors: ArrayList<Floor></div>
<div>- readInputFile(filename: String): ArrayList<ElevatorRequest></div> <div>+ setDirectionLamp(elevatorId: int, floorNumber: int, direction: Direction, state: boolean): void</div> <div>+ setFloorLamp(floorNumber: int, direction: Direction, state: boolean): void</div> <div>+ run(): void</div>

Scheduler
<ul style="list-style-type: none">- floorSubsystems: FloorSubsystem- elevatorPorts: int[]- senderReceiver1: UDPSenderReceiver- senderReceiver2: UDPSenderReceiver
<ul style="list-style-type: none">+ setFloorSubsystem(floorSubsystems: FloorSubsystem): void+ addElevator(elevatorId: int, elevatorPort: int): void+ run(): void+ selectBestElevatorNumber(elevatorRequest: ElevatorRequest): int