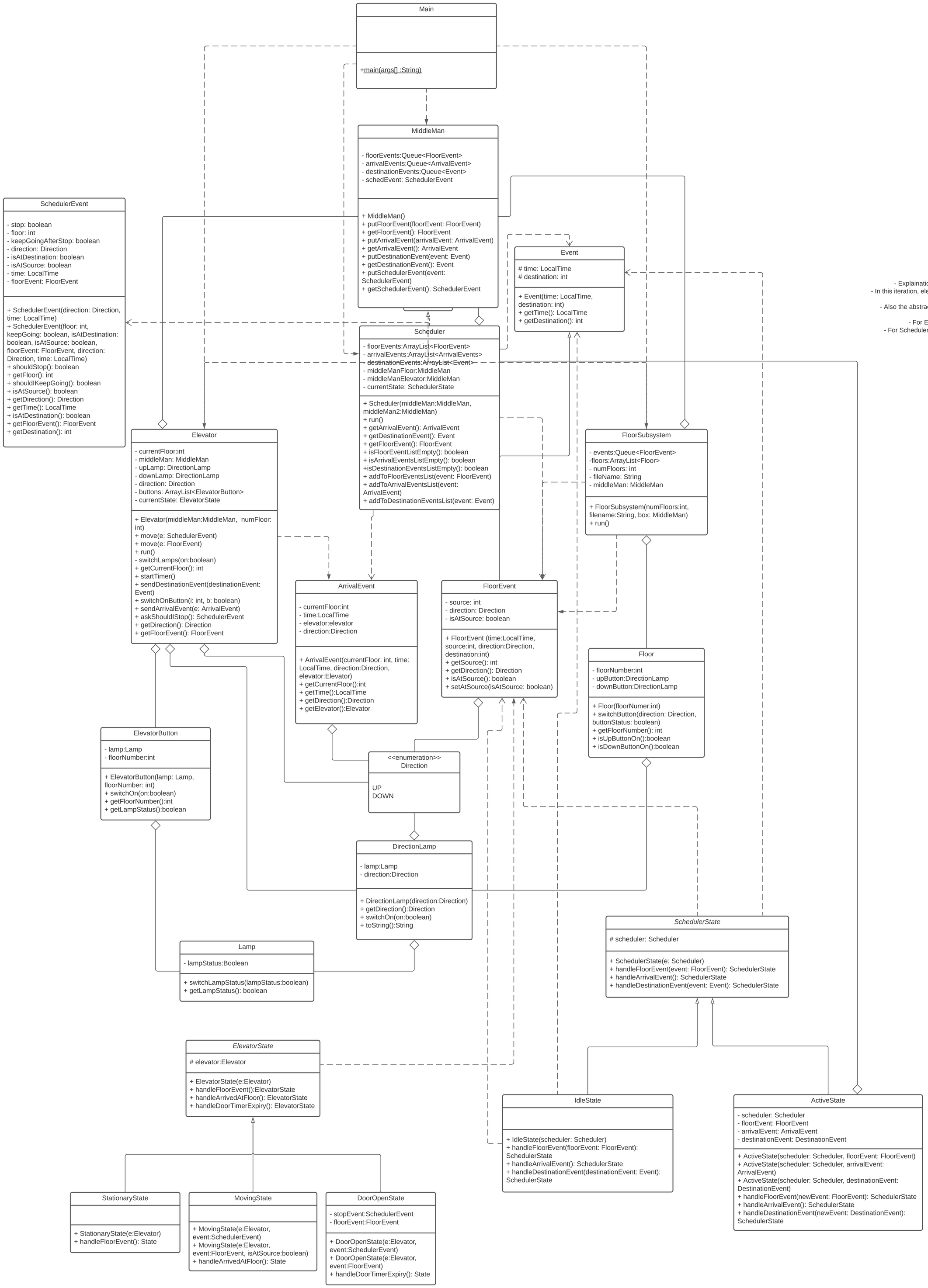
Main
+main(args[] :String)
MiddleMan
- floorEvents:Queue<FloorEvent>
- arrivalEvents:Queue<ArrivalEvent>
- destinationEvents:Queue<Event>
- schedEvent: SchedulerEvent
+ MiddleMan()
+ putFloorEvent(floorEvent: FloorEvent)
+ getFloorEvent(): FloorEvent
+ putArrivalEvent(arrivalEvent: ArrivalEvent)
+ getArrivalEvent(): ArrivalEvent
+ putDestinationEvent(event: Event)
+ getDestinationEvent(): Event
+ putSchedulerEvent(event: SchedulerEvent)
+ getSchedulerEvent(): SchedulerEvent
SchedulerEvent
- stop: boolean
- floor: int
- keepGoingAfterStop: boolean
- direction: Direction
- isAtDestination: boolean
- isAtSource: boolean
- time: LocalTime
- floorEvent: FloorEvent
+ SchedulerEvent(direction: Direction, time: LocalTime)
+ SchedulerEvent(floor: int, keepGoing: boolean, isAtDestination: boolean, isAtSource: boolean, floorEvent: FloorEvent, direction: Direction, time: LocalTime)
+ shouldStop(): boolean
+ getFloor(): int
+ shouldIKeepGoing(): boolean
+ isAtSource(): boolean
+ getDirection(): Direction
+ getTime(): LocalTime
+ isAtDestination(): boolean
+ getFloorEvent(): FloorEvent
+ getDestination(): int
Event
time: LocalTime
destination: int
+ Event(time: LocalTime, destination: int)
+ getTime(): LocalTime
+ getDestination(): int
Summary:
- Explaination of all the classes can be found in the README.md
- In this iteration, elevatorstate the abstract class and its 3 children were added to the UML.
- Also the abstract class SchedulerState and its 2 children ActiveState, and IdleState.
- For ElevatorState it has relationships with Floorevent.
- For SchedulerState it has relationships with FloorEvent, Scheduler, and ScheduleState.
Scheduler
- floorEvents:ArrayList<FloorEvent>
- arrivalEvents:ArrayList<ArrivalEvents>
- destinationEvents:ArrayList<Event>
- middleManFloor:MiddleMan
- middleManElevator:MiddleMan
- currentState: SchedulerState
+ Scheduler(middleMan:MiddleMan, middleMan2:MiddleMan)
+ run()
+ getArrivalEvent(): ArrivalEvent
+ getDestinationEvent(): Event
+ getFloorEvent(): FloorEvent
+ isFloorEventListEmpty(): boolean
+ isArrivalEventsListEmpty(): boolean
+isDestinationEventsListEmpty(): boolean
+ addToFloorEventsList(event: FloorEvent)
+ addToArrivalEventsList(event: ArrivalEvent)
+ addToDestinationEventsList(event: Event)
Elevator
- currentFloor:int
- middleMan: MiddleMan
- upLamp: DirectionLamp
- downLamp: DirectionLamp
- direction: Direction
- buttons: ArrayList<ElevatorButton>
- currentState: ElevatorState
+ Elevator(middleMan:MiddleMan, numFloor: int)
+ move(e: SchedulerEvent)
+ move(e: FloorEvent)
+ run()
- switchLamps(on:boolean)
+ getCurrentFloor(): int
+ startTimer()
+ sendDestinationEvent(destinationEvent: Event)
+ switchOnButton(i: int, b: boolean)
+ sendArrivalEvent(e: ArrivalEvent)
+ askShouldIStop(): SchedulerEvent
+ getDirection(): Direction
+ getFloorEvent(): FloorEvent
FloorSubsystem
- events:Queue<FloorEvent>
-floors:ArrayList<Floor>
- numFloors: int
- fileName: String
- middleMan: MiddleMan
+ FloorSubsystem(numFloors:int, filename:String, box: MiddleMan)
+ run()
ArrivalEvent
- currentFloor:int
- time:LocalTime
- elevator:elevator
- direction:Direction
+ ArrivalEvent(currentFloor: int, time: LocalTime, direction:Direction, elevator:Elevator)
+ getCurrentFloor():int
+ getTime():LocalTime
+ getDirection():Direction
+ getElevator():Elevator
FloorEvent
- source: int
- direction: Direction
- isAtSource: boolean
+ FloorEvent (time:LocalTime, source:int, direction:Direction, destination:int)
+ getSource(): int
+ getDirection(): Direction
+ isAtSource(): boolean
+ setAtSource(isAtSource: boolean)
Floor
- floorNumber:int
- upButton:DirectionLamp
- downButton:DirectionLamp
+ Floor(floorNumer:int)
+ switchButton(direction: Direction, buttonStatus: boolean)
+ getFloorNumber(): int
+ isUpButtonOn():boolean
+ isDownButtonOn():boolean
ElevatorButton
- lamp:Lamp
- floorNumber:int
+ ElevatorButton(lamp: Lamp, floorNumber: int)
+ switchOn(on:boolean)
+ getFloorNumber():int
+ getLampStatus():boolean
<<enumeration>>
Direction
UP
DOWN
DirectionLamp
- lamp:Lamp
- direction:Direction
+ DirectionLamp(direction:Direction)
+ getDirection():Direction
+ switchOn(on:boolean)
+ toString():String
Lamp
- lampStatus:Boolean
+ switchLampStatus(lampStatus:boolean)
+ getLampStatus(): boolean
SchedulerState
scheduler: Scheduler
+ SchedulerState(e: Scheduler)
+ handleFloorEvent(event: FloorEvent): SchedulerState
+ handleArrivalEvent(): SchedulerState
+ handleDestinationEvent(event: Event): SchedulerState
ElevatorState
elevator:Elevator
+ ElevatorState(e:Elevator)
+ handleFloorEvent():ElevatorState
+ handleArrivedAtFloor(): ElevatorState
+ handleDoorTimerExpiry(): ElevatorState
IdleState
+ IdleState(scheduler: Scheduler)
+ handleFloorEvent(floorEvent: FloorEvent): SchedulerState
+ handleArrivalEvent(): SchedulerState
+ handleDestinationEvent(destinationEvent: Event): SchedulerState
ActiveState
- scheduler: Scheduler
- floorEvent: FloorEvent
- arrivalEvent: ArrivalEvent
- destinationEvent: DestinationEvent
+ ActiveState(scheduler: Scheduler, floorEvent: FloorEvent)
+ ActiveState(scheduler: Scheduler, arrivalEvent: ArrivalEvent)
+ ActiveState(scheduler: Scheduler, destinationEvent: DestinationEvent)
+ handleFloorEvent(newEvent: FloorEvent): SchedulerState
+ handleArrivalEvent(): SchedulerState
+ handleDestinationEvent(newEvent: DestinationEvent): SchedulerState
StationaryState
+ StationaryState(e:Elevator)
+ handleFloorEvent(): State
MovingState
+ MovingState(e:Elevator, event:SchedulerEvent)
+ MovingState(e:Elevator, event:FloorEvent, isAtSource:boolean)
+ handleArrivedAtFloor(): State
DoorOpenState
- stopEvent:SchedulerEvent
- floorEvent:FloorEvent
+ DoorOpenState(e:Elevator, event:SchedulerEvent)
+ DoorOpenState(e:Elevator, event:FloorEvent)
+ handleDoorTimerExpiry(): State