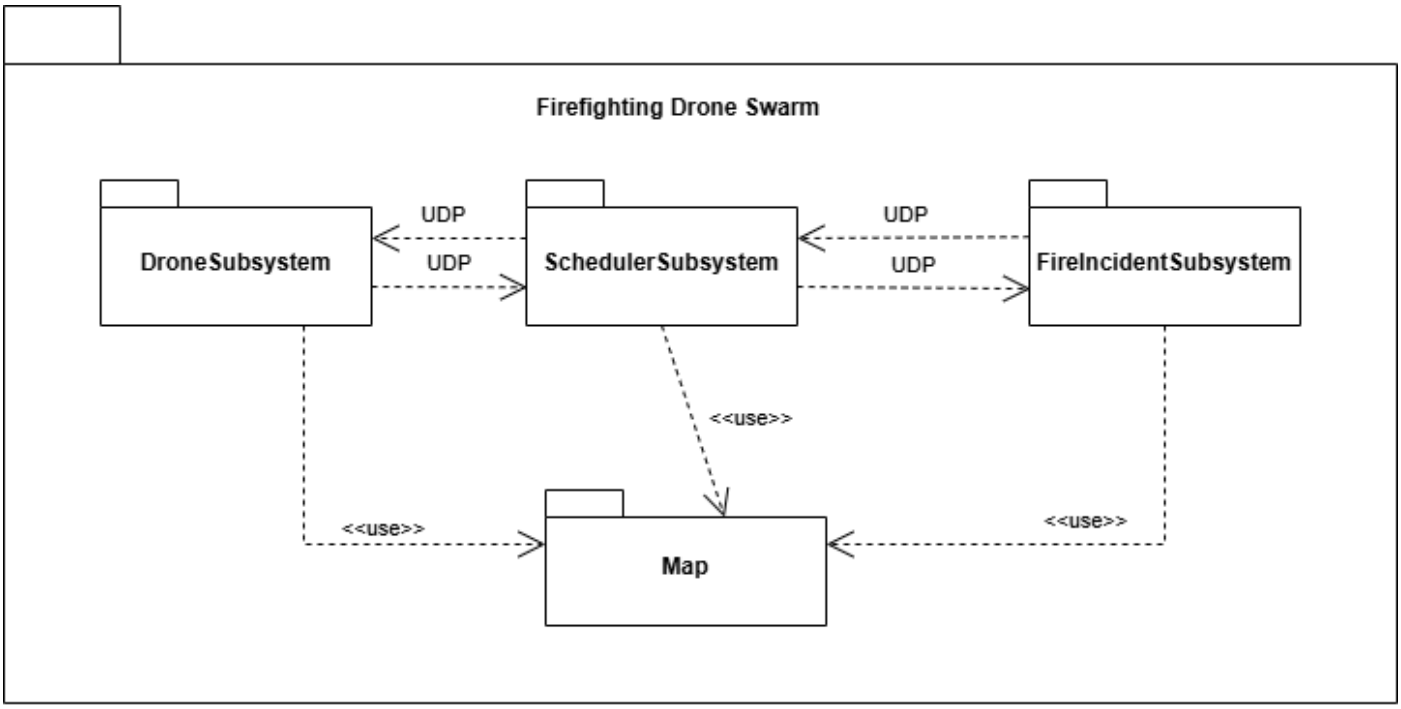
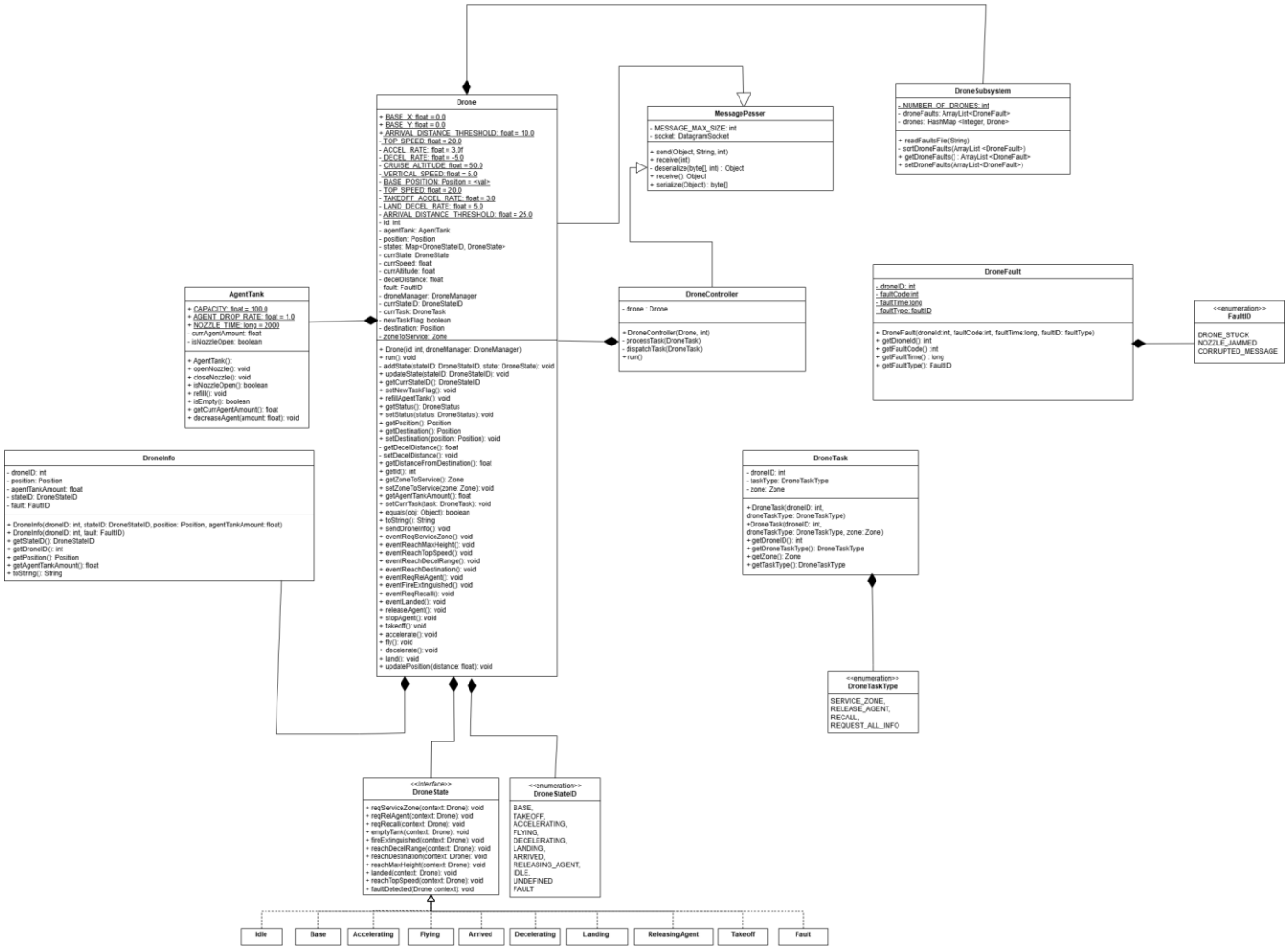
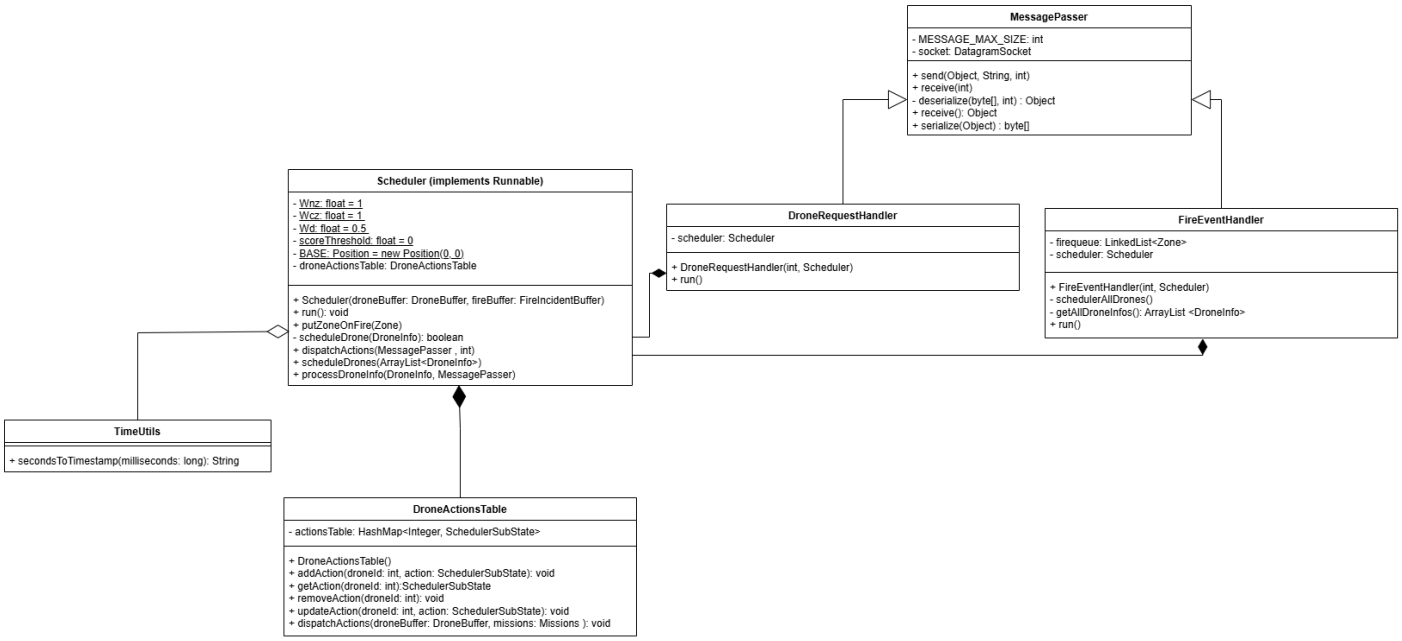
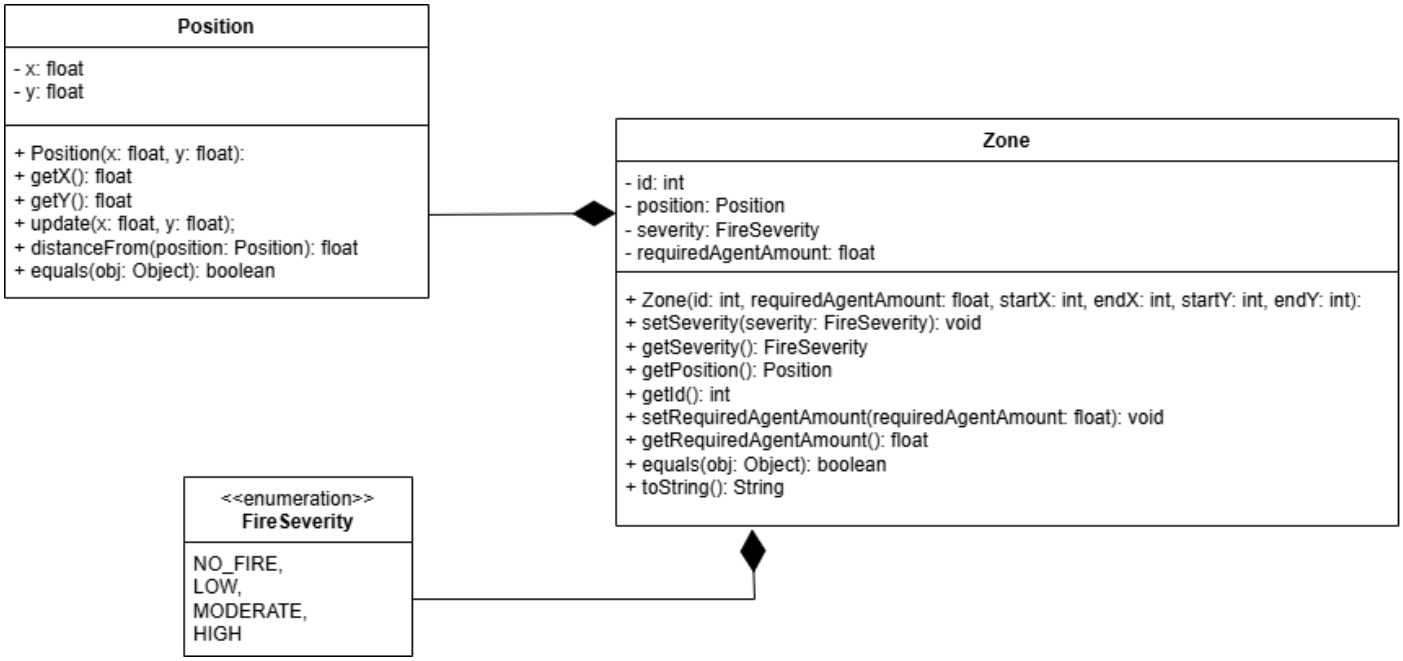

Firefighting Drone Swarm
DroneSubsystem
ScheduleSubsystem
FireIncidentSubsystem
Map
UDP
UDP
UDP
UDP
<<use>>
<<use>>
<<use>>


Drone
+ BASE_X: float = 0.0
+ BASE_Y: float = 0.0
+ ARRIVAL_DISTANCE_THRESHOLD: float = 10.0
- TOP_SPEED: float = 20.0
- ACCEL_RATE: float = 3.0f
- DECEL_RATE: float = -5.0
- CRUISE_ALTITUDE: float = 50.0
- VERTICAL_SPEED: float = 5.0
- BASE_POSITION: Position = <null>
- TOP_SPEED: float = 20.0
- TAKEOFF_ACCEL_RATE: float = 3.0
- LAND_DECEL_RATE: float = 5.0
- ARRIVAL_DISTANCE_THRESHOLD: float = 25.0
- id: int
- agentTank: AgentTank
- position: Position
- states: Map<DroneStateID, DroneState>
- currState: DroneState
- currSpeed: float
- currAltitude: float
- decelDistance: float
- fault: FaultID
- droneManager: DroneManager
- currStateID: DroneStateID
- currTask: DroneTask
- newTaskFlag: boolean
- destination: Position
- zoneToService: Zone
+ Drone(id: int, droneManager: DroneManager)
+ run(): void
- addState(stateID: DroneStateID, state: DroneState): void
+ updateState(stateID: DroneStateID): void
+ getCurrStateID(): DroneStateID
+ setNewTaskFlag(): void
+ refillAgentTank(): void
+ getStatus(): DroneStatus
+ setStatus(status: DroneStatus): void
+ getPosition(): Position
+ getDestination(): Position
+ setDestination(position: Position): void
+ getDecelDistance(): float
+ setDecelDistance(): void
+ getDistanceFromDestination(): float
+ getId(): int
+ getZoneToService(): Zone
+ setZoneToService(zone: Zone): void
+ getAgentTankAmount(): float
+ setCurrTask(task: DroneTask): void
+ equals(obj: Object): boolean
+ toString(): String
+ sendDroneInfo(): void
+ eventReqServiceZone(): void
+ eventReachMaxHeight(): void
+ eventReachTopSpeed(): void
+ eventReachDecelRange(): void
+ eventReachDestination(): void
+ eventReqRelAgent(): void
+ eventFireExtinguished(): void
+ eventReqRecall(): void
+ eventLanded(): void
+ releaseAgent(): void
+ stopAgent(): void
+ takeoff(): void
+ accelerate(): void
+ fly(): void
+ decelerate(): void
+ land(): void
+ updatePosition(distance: float): void
MessagePasser
- MESSAGE_MAX_SIZE: int
- socket: DatagramSocket
+ send(Object, String, int)
+ receive(int)
- deserialize(byte[], int) : Object
+ receive(): Object
+ serialize(Object) : byte[]
DroneSubsystem
- NUMBER_OF_DRONES: int
- droneFaults: ArrayList<DroneFault>
- drones: HashMap <Integer, Drone>
+ readFaultsFile(String)
- sortDroneFaults(ArrayList <DroneFault>)
+ getDroneFaults() : ArrayList <DroneFault>
+ setDroneFaults(ArrayList<DroneFault>)
AgentTank
+ CAPACITY: float = 100.0
+ AGENT_DROP_RATE: float = 1.0
+ NOZZLE_TIME: long = 2000
- currAgentAmount: float
- isNozzleOpen: boolean
+ AgentTank():
+ openNozzle(): void
+ closeNozzle(): void
+ isNozzleOpen(): boolean
+ refill(): void
+ isEmpty(): boolean
+ getCurrAgentAmount(): float
+ decreaseAgent(amount: float): void
DroneController
- drone : Drone
+ DroneController(Drone, int)
- processTask(DroneTask)
- dispatchTask(DroneTask)
- run()
DroneFault
- droneID: int
- faultCode: int
- faultTime:long
- faultType: faultID
+ DroneFault(droneId:int, faultCode:int, faultTime:long, faultID: faultType)
+ getDroneId(): int
+ getFaultCode(): int
+ getFaultTime() : long
+ getFaultType(): FaultID
<<enumeration>>
FaultID
DRONE_STUCK
NOZZLE_JAMMED
CORRUPTED_MESSAGE
DroneInfo
- droneID: int
- position: Position
- agentTankAmount: float
- stateID: DroneStateID
- fault: FaultID
+ DroneInfo(droneID: int, stateID: DroneStateID, position: Position, agentTankAmount: float)
+ DroneInfo(droneID: int, fault: FaultID)
+ getStateID(): DroneStateID
+ getDroneID(): int
+ getPosition(): Position
+ getAgentTankAmount(): float
+ toString(): String
DroneTask
- droneID: int
- taskType: DroneTaskType
- zone: Zone
+ DroneTask(droneID: int, droneTaskType: DroneTaskType)
+DroneTask(droneID: int, droneTaskType: DroneTaskType, zone: Zone)
+ getDroneID(): int
+ getDroneTaskType(): DroneTaskType
+ getZone(): Zone
+ getTaskType(): DroneTaskType
<<enumeration>>
DroneTaskType
SERVICE_ZONE,
RELEASE_AGENT,
RECALL,
REQUEST_ALL_INFO
<<Interface>>
DroneState
+ reqServiceZone(context: Drone): void
+ reqRelAgent(context: Drone): void
+ reqRecall(context: Drone): void
+ emptyTank(context: Drone): void
+ fireExtinguished(context: Drone): void
+ reachDecelRange(context: Drone): void
+ reachDestination(context: Drone): void
+ reachMaxHeight(context: Drone): void
+ landed(context: Drone): void
+ reachTopSpeed(context: Drone): void
+ faultDetected(Drone context): void
<<enumeration>>
DroneStateID
BASE,
TAKEOFF,
ACCELERATING,
FLYING,
DECELERATING,
LANDING,
ARRIVED,
RELEASING_AGENT,
IDLE,
UNDEFINED
FAULT
Idle
Base
Accelerating
Flying
Arrived
Decelerating
Landing
ReleasingAgent
Takeoff
Fault


MessagePasser
- MESSAGE_MAX_SIZE: int
- socket: DatagramSocket
+ send(Object, String, int)
+ receive(int)
- deserialize(byte[], int) : Object
+ receive(): Object
+ serialize(Object) : byte[]
Scheduler (implements Runnable)
- Wnz: float = 1
- Wcz: float = 1
- Wd: float = 0.5
- scoreThreshold: float = 0
- BASE: Position = new Position(0, 0)
- droneActionsTable: DroneActionsTable
+ Scheduler(droneBuffer: DroneBuffer, fireBuffer: FireIncidentBuffer)
+ run(): void
+ putZoneOnFire(Zone)
- scheduleDrone(DroneInfo): boolean
+ dispatchActions(MessagePasser , int)
+ scheduleDrones(ArrayList<DroneInfo>)
+ processDroneInfo(DroneInfo, MessagePasser)
DroneRequestHandler
- scheduler: Scheduler
+ DroneRequestHandler(int, Scheduler)
+ run()
FireEventHandler
- firequeue: LinkedList<Zone>
- scheduler: Scheduler
+ FireEventHandler(int, Scheduler)
- schedulerAllDrones()
- getAllDroneInfos(): ArrayList <DroneInfo>
+ run()
TimeUtils
+ secondsToTimestamp(milliseconds: long): String
DroneActionsTable
- actionsTable: HashMap<Integer, SchedulerSubState>
+ DroneActionsTable()
+ addAction(droneId: int, action: SchedulerSubState): void
+ getAction(droneId: int):SchedulerSubState
+ removeAction(droneId: int): void
+ updateAction(droneId: int, action: SchedulerSubState): void
+ dispatchActions(droneBuffer: DroneBuffer, missions: Missions): void


Position
- x: float
- y: float
+ Position(x: float, y: float):
+ getX(): float
+ getY(): float
+ update(x: float, y: float);
+ distanceFrom(position: Position): float
+ equals(obj: Object): boolean
Zone
- id: int
- position: Position
- severity: FireSeverity
- requiredAgentAmount: float
+ Zone(id: int, requiredAgentAmount: float, startX: int, endX: int, startY: int, endY: int):
+ setSeverity(severity: FireSeverity): void
+ getSeverity(): FireSeverity
+ getPosition(): Position
+ getId(): int
+ setRequiredAgentAmount(requiredAgentAmount: float): void
+ getRequiredAgentAmount(): float
+ equals(obj: Object): boolean
+ toString(): String
<<enumeration>>
FireSeverity
NO_FIRE,
LOW,
MODERATE,
HIGH