







MessagePasser

- MESSAGE_MAX_SIZE: int socket: DatagramSocket
- + send(Object, String, int) + receive(int)
- deserialize(byte[], int) : Object
- + receive(): Object + serialize(Object) : byte[]

FireIncidentSubsystem

- + AGENT_AMOUNT: HashMap<FireSeverity_Integer> fireBuffer: FireIncidentBuffer clearZones: HashMap<Integer, Zone>

- fireZones: HashMap<Integer, Zone>
- events: ArrayList<SimEvent>
- + FireIncidentSubsystem(fireBuffer: FireIncidentBuffer)
- manualReqDrone(zone: Zone, eventTime: long, eventType: String): void trackFire(zone: Zone, eventTime: long): void

- track-ire(zone: Zone, eventr line: long): + readSimEventFile(eventFile: File): void + readSimZoneFile(zoneFile: File): void + run(): void + simStart(): void

- hasActiveFiresOrUpcomingEvents(eventIndex: int): boolean
 isEventReadyToProcess(eventIndex: int, eventIndexTime: long. currentTime: long): boolean
- sendEvent(event: SimEvent): void isOnFire(zone: Zone): boolean
- sortEventsByTime(events: ArrayList<SimEvent>): void timeToMillis(time: String): long

- uniter townins (time); long getCurrentTime(): long + getClearZones(): HashMap<Integer, Zone> + getEvents(): ArrayList<SimEvent> + getFireZones(): HashMap<Integer, Zone>

FireIncidentBuffer

- acknowledgmentMessages: ArrayList<Zone>
- eventMessages: ArrayList<Zone:
- + FireIncidentBuffer()
- + popAcknowledgementMessage(): Zone + popEventMessage(): Zone

- + addEventMessage(event: Zone): void + addAcknowledgementMessage(zoneMessage: Zone): void
- + newEvent(): boolean + newAcknowledgement(): boolean



SimEvent

- zoneld: int
- eventType: String severity: FireSeverity
- + SimEvent(time: long, zoneId: int, eventType: String, severity: String)
- + getTime(): long
- + getZoneId(): int
- + getEventType(): String
- + getSeverity(): FireSeverity + toString(): String