<<Interface>> <<Enumeration>> SimulationManager AircraftStatus - runways: std::vector<Runway> **LANDING** - userInterface: UserInterface* Aircraft **TAKEOFF** - logger: Logger **CRASHED** id: int **IDLE** - landingQueue: Queue code: std::string IN_AIR - takeOffQueue: Queue fuelLevel: int - threads: std::vector<std::thread> airTime: int <<Enumeration>> - statistics: Statistics runway: Runway* RunwayStatus - timer: Timer craftStatus: AircraftStatus + runSimulation(int simulationTime) getRunway() const **FREE OCCUPIED** getId() const + waitThreadsCompletion() getStatus() const + generateAircraft() <<Enumeration>> getFuelLevel() const + generateTakeoffAircraft(Aircraft&) getAirTime() const RunwayOperation + processAircraft() getCode() const **LANDING** + updateRunwayStatus() setRunway(Runway*) **TAKEOFF** + updateStatistics() **NONE** refill() const + handleLanding() setFuelLevel(int) + handleTakeOff() setStatus(AircraftStatus) + assignRunwayForLanding() Queue setAirTime(int) + assignRunwayForTakeOff() toString() const - landingQueue: pr_queue<Aircraft> + logRunwayStatus(std::string&) takeOffQueue: std::queue<Aircraft> + logEvent(std::string&, bool) Directed Association - takeOffQueue: pr_queue<Aircraft> + getTakeOffQueueSize() const Runway + Queue() + displayStatistics() const + enqueueLanding(aircraft: Aircraft): + getRunways() const status: RunwayStatus void priority: bool + dequeueLanding(): Aircraft + getLandingQueueSize() const + displaySimulationStats() const planesLanded: int + dequeueTakeOff(): Aircraft + runwayStatusToString(RunwayStatus) planesTakeOff: int + isLandingQueueEmpty(): bool lastOperation: RunwayOperation + isTakeOffQueueEmpty(): bool lastChangeTime: std::chrono + landingQueueSize(): size_t getStatus() const **Statistics** getPlanesLanded() const + takeOffQueueSize(): size_t - crashedAircraftCount: int getPlanesTakeOff() const + isEmpty(): bool - totalLandingTime: seconds getLastOperation() const + dequeue(): Aircraft + enqueueTakeOff(aircraft: Aircraft): -memberName getLastChangeTime() const void - totalTakeoffTime: seconds getCode() const + getLandingQueue(): const priority_queue<Aircraft, - takeoffCount: int setStatus(RunwayStatus) vector<Aircraft>, CompareByFuel>& - totalLandedPlanes: int setPriority(bool) + getTakeOffQueue(): const priority queue<Aircraft, - totalTookOffPlanes: int incrementPlanesLanded() vector<Aircraft>, CompareByFuel>& - totalLandingWaitTime: sec incrementPlanesTakeOff() - totalFuelAvailable: int setLastOperation(RunwayOperation) - totalTakeoffWaitTime: sec hasLandingPriority() const UserInterface setLastChangeTime(std::chrono) + Statistics() - simulationManager: + aircraftCrashed(): void SimulationManager* + updateLandingWaitTime(seconds): **IdCodeGenerator** + UserInterface() void + UserInterface(SimulationManager&) + displayStatistics(): void - lastAssignedID: int + displaySimulationStatus() + updateTakeoffCount(): void # generateCode(const std::string&): std::string + displayLog(const std::string&) + updateTakeoffWaitTime(seconds): # generateID(): int + header() + getAverageLandingWaitTime(): double + IdCodeGenerator() + displayAircraftDetails(const Aircraft&) + getAverageTakeoffWaitTime(): double + ~IdCodeGenerator() + getCrashedAircraftCount(): int + getId(): int + getLandingCount(): int + getCode(): const std::string& Logger + getTakeoffCount(): int logFile: std::ofstream + getTotalLandingTime(): + Logger(const std::string&) Timer chrono::seconds + ~Logger() + getTotalFuelAvailable(): int - currentTime: int + logEvent(const std::string&) + getTotalLandedPlanes(): int + Timer() + getTotalTookOffPlanes(): int + getCurrentTime(): int + incrementLandedPlanes(): void + incrementTime(int): void crashHandling + incrementTookOffPlanes(): void + reset(): void + checkForCrash(Aircraft&, const + updateStatistics(runways): void std::vector<Runway>&, Statistics&) : + getTotalTakeoffTime(): void chrono::seconds