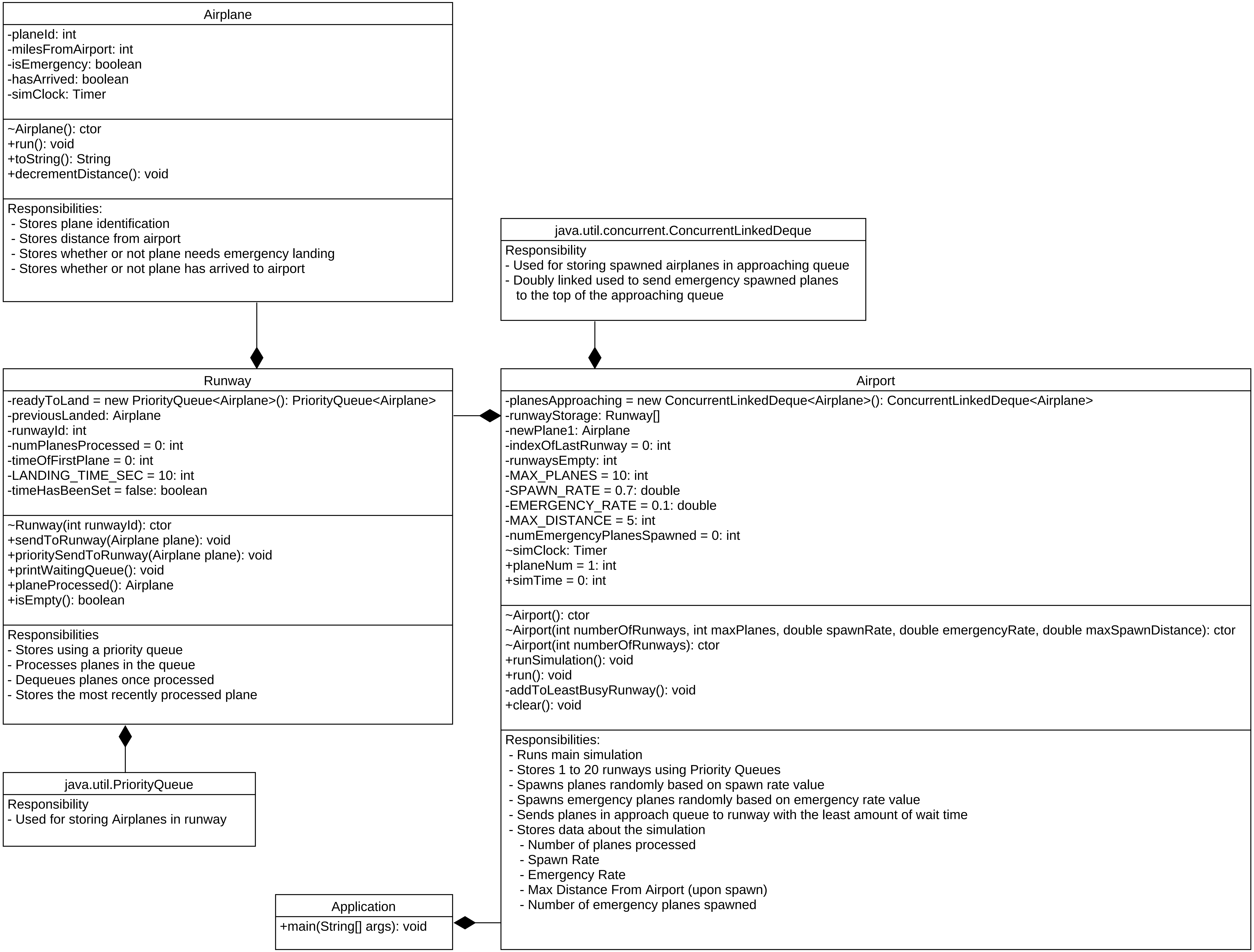

Airplane

-planeId: int
-milesFromAirport: int
-isEmergency: boolean
-hasArrived: boolean
-simClock: Timer

~Airplane(): ctor
+run(): void
+toString(): String
+decrementDistance(): void

Responsibilities:
- Stores plane identification
- Stores distance from airport
- Stores whether or not plane needs emergency landing
- Stores whether or not plane has arrived to airport

java.util.concurrent.ConcurrentLinkedDeque

Responsibility
- Used for storing spawned airplanes in approaching queue
- Doubly linked used to send emergency spawned planes to the top of the approaching queue

Runway

-readyToLand = new PriorityQueue<Airplane>(): PriorityQueue<Airplane>
-previousLanded: Airplane
-runwayId: int
-numPlanesProcessed = 0: int
-timeOfFirstPlane = 0: int
-LANDING_TIME_SEC = 10: int
-timeHasBeenSet = false: boolean

~Runway(int runwayId): ctor
+sendToRunway(Airplane plane): void
+prioritySendToRunway(Airplane plane): void
+printWaitingQueue(): void
+planeProcessed(): Airplane
+isEmpty(): boolean

Responsibilities
- Stores using a priority queue
- Processes planes in the queue
- Dequeues planes once processed
- Stores the most recently processed plane

Airport

-planesApproaching = new ConcurrentLinkedDeque<Airplane>(): ConcurrentLinkedDeque<Airplane>
-runwayStorage: Runway[]
-newPlane1: Airplane
-indexOfLastRunway = 0: int
-runwaysEmpty: int
-MAX_PLANES = 10: int
-SPAWN_RATE = 0.7: double
-EMERGENCY_RATE = 0.1: double
-MAX_DISTANCE = 5: int
-numEmergencyPlanesSpawned = 0: int
~simClock: Timer
+planeNum = 1: int
+simTime = 0: int

~Airport(): ctor
~Airport(int numberOfRunways, int maxPlanes, double spawnRate, double emergencyRate, double maxSpawnDistance): ctor
~Airport(int numberOfRunways): ctor
+runSimulation(): void
+run(): void
-addToLeastBusyRunway(): void
+clear(): void

Responsibilities:
- Runs main simulation
- Stores 1 to 20 runways using Priority Queues
- Spawns planes randomly based on spawn rate value
- Spawns emergency planes randomly based on emergency rate value
- Sends planes in approach queue to runway with the least amount of wait time
- Stores data about the simulation
  - Number of planes processed
  - Spawn Rate
  - Emergency Rate
  - Max Distance From Airport (upon spawn)
  - Number of emergency planes spawned

java.util.PriorityQueue

Responsibility
- Used for storing Airplanes in runway

Application

+main(String[] args): void