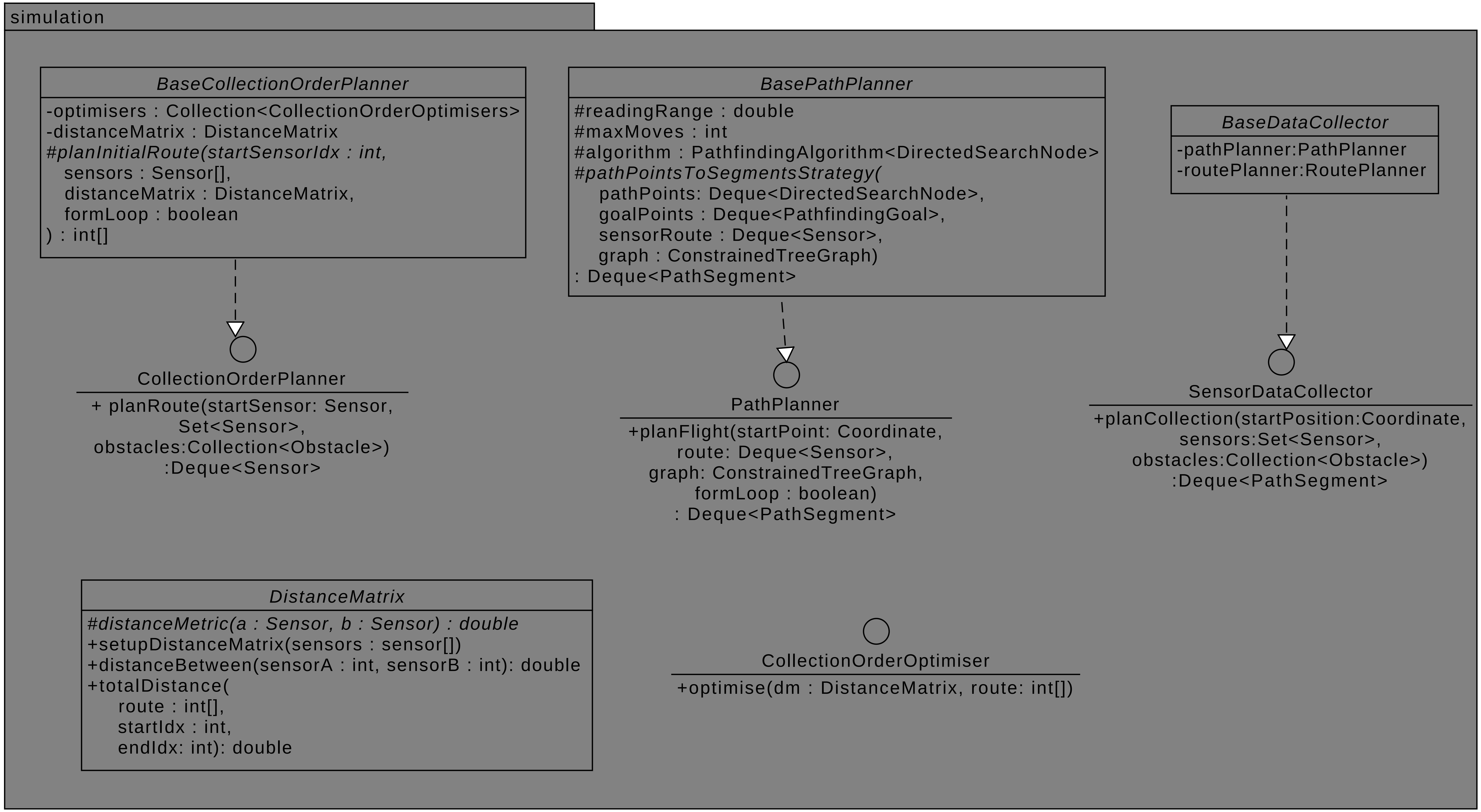
simulation
BaseCollectionOrderPlanner
-optimisers : Collection<CollectionOrderOptimisers>
-distanceMatrix : DistanceMatrix
#planInitialRoute(startSensorIdx : int,
sensors : Sensor[],
distanceMatrix : DistanceMatrix,
formLoop : boolean
) : int[]
BasePathPlanner
#readingRange : double
#maxMoves : int
#algorithm : PathfindingAlgorithm<DirectedSearchNode>
#pathPointsToSegmentsStrategy(
pathPoints: Deque<DirectedSearchNode>,
goalPoints : Deque<PathfindingGoal>,
sensorRoute : Deque<Sensor>,
graph : ConstrainedTreeGraph)
: Deque<PathSegment>
BaseDataCollector
-pathPlanner:PathPlanner
-routePlanner:RoutePlanner
CollectionOrderPlanner
+ planRoute(startSensor: Sensor,
Set<Sensor>,
obstacles:Collection<Obstacle>)
:Deque<Sensor>
PathPlanner
+planFlight(startPoint: Coordinate,
route: Deque<Sensor>,
graph: ConstrainedTreeGraph,
formLoop : boolean)
: Deque<PathSegment>
SensorDataCollector
+planCollection(startPosition:Coordinate,
sensors:Set<Sensor>,
obstacles:Collection<Obstacle>)
:Deque<PathSegment>
DistanceMatrix
#distanceMetric(a : Sensor, b : Sensor) : double
+setupDistanceMatrix(sensors : sensor[])
+distanceBetween(sensorA : int, sensorB : int): double
+totalDistance(
route : int[],
startIdx : int,
endIdx: int): double
CollectionOrderOptimiser
+optimise(dm : DistanceMatrix, route: int[])