

Assumption: The strategy pattern can be implemented with regards to the type of algorithm used for shipments, such as creating and changing the state of it.

Shipment

+ status: String

+ id: String

+ notes: String[]

private set;

+ updateHistory: ShippingUpdate[]

ShippingUpdate

+ previousStatus: String + newStatus: String

+ timestamp: Long

+ expectedDeliveryDateTimeStamp:

+ currentLocation: String

+ addNote(note: String) + addUpdate(update: Update) - shipments: Shipment[]

+ findShipment(id: String): Shipment?

TrackingSimulator

+ addShipment(shipment:

Shipment) + runSimulator()

you will want addShipment to be public because it will make doing the strategy pattern a little easier

runSimulation reads the file and handles the updates to the corresponding shipment. You can probably just call it when the app launches.

Assumption: The observer pattern will be useful for notifying (from the TrackingSimulator) all of the needed areas that a shipment has changed