<<interface>> ShipmentObserver

+ abstract update(shipment: Shipment)

<<interface>> Subject

+ abstract registerObserver(observer: ShipmentObserver)
+ abstract removeObserver(observer: ShipmentObserver)
+ abstract notifyObservers()

StandardShipment

BulkShipment

OvernightShipment

ExpressShipment

0..*

ui

Creates and uses these

abstract Shipment

+ id: String
  private set
+ status: String
  private set;
+ notes: String[]
  private set;
+ updateHistory: ShippingUpdate[]
  private set;
+ expectedDeliveryDate: Date
  private set;
+ currentLocation: String
- creationDate: Date
- observers: Observers[]

+ Shipment(id, status)
+ addNote(note: String)
+ addUpdate(status: String, date: Date)
- abstract validateDeliveryDate(): Boolean

ShippingUpdate

+ previousStatus: String
+ newStatus: String
+ date: Date

0..*

ShipmentFactory

- shipmentTypeMap: Map<String, ShipmentUpdateStrategy>

+ ceate(id: String, status: String, type: String)

TrackerViewHelper

+ shipmentID: State<String>
+ shipmentStatus: State<String>
+ shipmentLocation: State<String>
+ expectedShipmentDeliveryDate: State<String>
+ shipmentNotes: State<String[]>
+ shipmentUpdateHistory: State<String[]>

* all attributes have a private setter in this class

+ TrackerViewHelper()
+ trackShipmentID(id: String): Boolean
+ stopTracking()
+ update()

abstract ShipmentTracker

- shipments: Shipment[]

+ findShipment(id: String): Shipment?
+ addShipment(shipment: Shipment)
+ reset()
+ abstract run()

0..*

ShipmentUpdateStrategyFactory

- strategyMap: Map<String, ShipmentUpdateStrategy>

create(updateType: String): ShipmentUpdateStrategy

<<interface>> ShipmentUpdateStrategy

+ abstract update(id: String, updateType: String, dateOfUpdate: Date, otherInfo: String)

WebShipmentTracker

+ getInstance()
- ShipmentTracker()

StatusShipmentUpdateStrategy

CreatedShipmentUpdateStrategy

LocationShipmentUpdateStrategy

ExpectedDeliveryShipmentUpdateStrategy

NoteShipmentUpdateStrategy