Assignment 3 - Extend Shipment Tracker

TrackingServer — *Singleton Pattern*

- instance: TrackingServer
- shipments: MutableMap<String, Shipment>
- parser: ShipmentDataParser
- updater: ShipmentUpdater
- factory: ShipmentFactory

+ getInstance(): TrackingServer
+ createShipment(request: CreateShipmentRequest): CreateShipmentResponse
+ updateShipment(request: UpdateShipmentRequest): UpdateShipmentResponse
+ getShipment(id: String): GetShipmentResponse
+ trackShipment(id: String): Bool
+ stopTracking(id: String): Bool

TrackingClient

- serverConnection: ServerConnection

+ sendCreateRequest(data: String): Bool
+ sendUpdateRequest(data: String): Bool
+ parseUserInput(input: String) Request
+ displayResponse(response: ServerResponse)

Communicates with

ClientRequest

+ type: RequestType
+ data: String
+ timestamp: Long

CreateShipmentRequest

+ shipmentType: ShipmentType
+ shipmentId: String
+ expectedDeliveryDate: Long
+ initialLocation: String

UpdateShipmentRequest

+ shipmentId: String
+ updateType: String
+ timestamp: Long
+ location: String
+ notes: String
+ newDeliveryDate: Long

Server Response

+ success: Bool
+ message: String
+ shipmentData: Shipment
+ abnormality: String

<<interface>> ShipmentSubject

+ addObserver(observer: ShipmentObserver)
+ removeObserver(observer: ShipmentObserver)
+ notifyObservers(shipment: Shipment, update: ShippingUpdate)

implements

ShipmentUpdater

- processors: Map<String, UpdateProcessor>

+ processUpdate(updateType: String, shipment: Shipment, timestamp: Long): ShippingUpdate

Uses

ShippingUpdate

+ previousStatus: String
+ newStatus: String
+ timestamp: Long

<<interface>> UpdateProcessorStrategy

+ processUpdate(shipment: Shipment, timestamp: Long): ShippingUpdate

Strategy Pattern

<<interface>> UI — *Notify if shipment doesn't exist*

Creates and uses

Observer Pattern

Shipment

+ status: String
+ id: String
+ notes: String[]
+ updateHistory: ShippingUpdate[]
+ expectedDeliveryDateTimestamp: Long
+ currentLocation: String
+ shipmentType: ShipmentType
+ creationDate: Long
+ isAbnormal: Bool
+ abnormality: String

+ addNote(note: String)
+ addUpdate(update: hippingUpdate)
+ updateLocation(location: String)
+ validateDeliveryDate(): ValidationResult
+ markAbnormal(abnormality: String)
+ clearAbnormality()

Observes

0..*

TrackerViewHelper — *has responsibility of preparing data for the using interface to display*

+ activeTrackingShipments: String[]
+ shipmentId: State<String>
+ shipmentTotes: State<String[]>
+ shipmentUpdateHistory: State<String[]>
+ expectedShipmentDeliveryDate: State<String[]>
+ shipmentStatus: State<String>
+ abnormalShipments: Man<String, String>

+ trackShipment(id: String)
+ stopTracking()
+ displayAbnormalityWarning(shipmentId: String, abnormality: String)
+ clearAbnormalityWarning(shipmentId: String)

implements

<<interface>> ShipmentObserver

+ onShipmentUpdate(shipmentId: String, update: ShippingUpdate)
+ onAbnormalityDetected(shipmentId: String, abnormality: String)

StandardShipment

ExpressShipment

*abnormality: <= 3 day delivery

OvernightShipment

*abnormality: next day delivery

BulkShipment

*abnormality: > 3 day delivery

TrackingSimulator

- shipments: Shipment[]
- parser: ShipmentDataParser
- updater: ShipmentUpdater

+ findShipment(id: String): Shipment?
+ addShipment(shipment: Shipment)
+ runSimulation()
+ processFileUpdate(update: string)

1

ShipmentDataParser

+ readFile(filePath: String): String[]
+ parseUpdate(update: String): String[]

CreatedUpdateStrategy

ShippedUpdateStrategy

LocationUpdateStrategy

DeliveredUpdateStrategy

DelayedUpdateStrategy

LostUpdateStrategy

CanceledUpdateStrategy

NoteAddedUpdateStrategy

Factory Pattern

ShipmentFactory

+ createShipment(type: ShipmentType, id: String, expectedDate: Long, location: String, creationTime: Long): Shipment