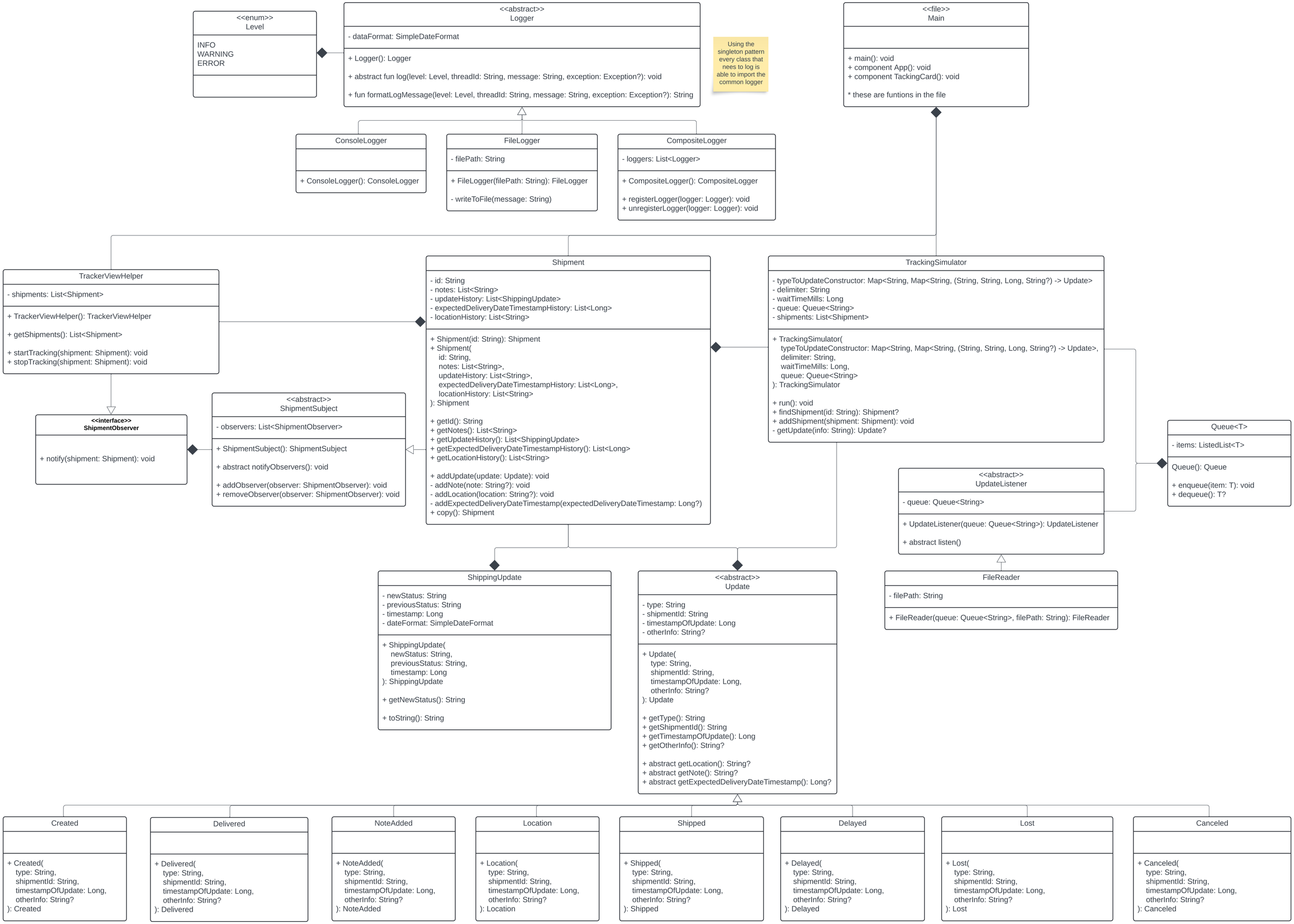
<<enum>>
Level
INFO
WARNING
ERROR
<<abstract>>
Logger
- dataFormat: SimpleDateFormat
+ Logger(): Logger
+ abstract fun log(level: Level, threadId: String, message: String, exception: Exception?): void
+ fun formatLogMessage(level: Level, threadId: String, message: String, exception: Exception?): String
Using the singleton pattern every class that nees to log is able to import the common logger
<<file>>
Main
+ main(): void
+ component App(): void
+ component TackingCard(): void
* these are funtions in the file
ConsoleLogger
+ ConsoleLogger(): ConsoleLogger
FileLogger
- filePath: String
+ FileLogger(filePath: String): FileLogger
- writeToFile(message: String)
CompositeLogger
- loggers: List<Logger>
+ CompositeLogger(): CompositeLogger
+ registerLogger(logger: Logger): void
+ unregisterLogger(logger: Logger): void
TrackerViewHelper
- shipments: List<Shipment>
+ TrackerViewHelper(): TrackerViewHelper
+ getShipments(): List<Shipment>
+ startTracking(shipment: Shipment): void
+ stopTracking(shipment: Shipment): void
Shipment
- id: String
- notes: List<String>
- updateHistory: List<ShippingUpdate>
- expectedDeliveryDateTimestampHistory: List<Long>
- locationHistory: List<String>
+ Shipment(id: String): Shipment
+ Shipment(
id: String,
notes: List<String>,
updateHistory: List<String>,
expectedDeliveryDateTimestampHistory: List<Long>,
locationHistory: List<String>
): Shipment
+ getId(): String
+ getNotes(): List<String>
+ getUpdateHistory(): List<ShippingUpdate>
+ getExpectedDeliveryDateTimestampHistory(): List<Long>
+ getLocationHistory(): List<String>
+ addUpdate(update: Update): void
- addNote(note: String?): void
- addLocation(location: String?): void
- addExpectedDeliveryDateTimestamp(expectedDeliveryDateTimestamp: Long?)
+ copy(): Shipment
TrackingSimulator
- typeToUpdateConstructor: Map<String, Map<String, (String, String, Long, String?) -> Update>
- delimiter: String
- waitTimeMills: Long
- queue: Queue<String>
- shipments: List<Shipment>
+ TrackingSimulator(
typeToUpdateConstructor: Map<String, Map<String, (String, String, Long, String?) -> Update>,
delimiter: String,
waitTimeMills: Long,
queue: Queue<String>
): TrackingSimulator
+ run(): void
+ findShipment(id: String): Shipment?
+ addShipment(shipment: Shipment): void
- getUpdate(info: String): Update?
<<interface>>
ShipmentObserver
+ notify(shipment: Shipment): void
<<abstract>>
ShipmentSubject
- observers: List<ShipmentObserver>
+ ShipmentSubject(): ShipmentSubject
+ abstract notifyObservers(): void
+ addObserver(observer: ShipmentObserver): void
+ removeObserver(observer: ShipmentObserver): void
Queue<T>
- items: ListedList<T>
Queue(): Queue
+ enqueue(item: T): void
+ dequeue(): T?
<<abstract>>
UpdateListener
- queue: Queue<String>
+ UpdateListener(queue: Queue<String>): UpdateListener
+ abstract listen()
FileReader
- filePath: String
+ FileReader(queue: Queue<String>, filePath: String): FileReader
ShippingUpdate
- newStatus: String
- previousStatus: String
- timestamp: Long
- dateFormat: SimpleDateFormat
+ ShippingUpdate(
newStatus: String,
previousStatus: String,
timestamp: Long
): ShippingUpdate
+ getNewStatus(): String
+ toString(): String
<<abstract>>
Update
- type: String
- shipmentId: String
- timestampOfUpdate: Long
- otherInfo: String?
+ Update(
type: String,
shipmentId: String,
timestampOfUpdate: Long,
otherInfo: String?
): Update
+ getType(): String
+ getShipmentId(): String
+ getTimestampOfUpdate(): Long
+ getOtherInfo(): String?
+ abstract getLocation(): String?
+ abstract getNote(): String?
+ abstract getExpectedDeliveryDateTimestamp(): Long?
Created
+ Created(
type: String,
shipmentId: String,
timestampOfUpdate: Long,
otherInfo: String?
): Created
Delivered
+ Delivered(
type: String,
shipmentId: String,
timestampOfUpdate: Long,
otherInfo: String?
): Delivered
NoteAdded
+ NoteAdded(
type: String,
shipmentId: String,
timestampOfUpdate: Long,
otherInfo: String?
): NoteAdded
Location
+ Location(
type: String,
shipmentId: String,
timestampOfUpdate: Long,
otherInfo: String?
): Location
Shipped
+ Shipped(
type: String,
shipmentId: String,
timestampOfUpdate: Long,
otherInfo: String?
): Shipped
Delayed
+ Delayed(
type: String,
shipmentId: String,
timestampOfUpdate: Long,
otherInfo: String?
): Delayed
Lost
+ Lost(
type: String,
shipmentId: String,
timestampOfUpdate: Long,
otherInfo: String?
): Lost
Canceled
+ Canceled(
type: String,
shipmentId: String,
timestampOfUpdate: Long,
otherInfo: String?
): Canceled