

Picker
- status: String
- currWork: PickingRequest
- checkResult: boolean
Picker(name: String)
pickingSingleFascia(pickLocation: String, sku: String): void

Loader
- status: String
- currWork: PickingRequest
- checkResult: boolean
Loader(name: String)
loadTruck(currTruck: Truck): void
checkFascia(): void

Sequencer
- status: String
- currWork: PickingRequest
- checkResult: boolean
Sequencer(name: String)
SequenceCheck(): void

Fascia
- colour: String
- model: String
- sku: String
- orientation: boolean
- location: String
Fascia(colour: String, model: String, sku: String, orientation: boolean)
getColour(): String
getModel(): String
toString(): String
getSku(): String
setLocation(): void
getLocation(): String
getOrientation(): boolean

BarcodeReader
- status: String
- currWork: PickingRequest
- checkResult: boolean
+ BarcodeReader(name: String)
+ setCurrWork(currPickingRequest: PickingRequest): void
+ isAvailable(): boolean
+ getName(): String
+ scanSku(): String
+ setCheckResult(result: boolean): void
+ getCheckResult(): boolean

WorkerFactory
- pickerList: ArrayList<BarcodeReader>
- sequencerList: ArrayList<BarcodeReader>
- loaderList: ArrayList<BarcodeReader>
WorkerFactory()
getWorkerType(): String, name: String): BarcodeReader
getPickedList(): ArrayList<BarcodeReader>
getSequencerList(): ArrayList<BarcodeReader>
getLoaderList(): ArrayList<BarcodeReader>
addLoader(loader: Loader): void
addSequencer(sequencer: Sequencer): void

PickingRequest
- id: int
- orderList: ArrayList<Order>
- frontSkus: ArrayList<String>
- backSkus: ArrayList<String>
- allSkus: ArrayList<String>
- fasciasList: ArrayList<Fascia>
- unsequencedFascias: ArrayList<String>
- locations: ArrayList<String>
PickingRequest(orders ArrayList<Order>, id: int)
getid(): int
getOrderList(): ArrayList<Order>
getAUSkus(): ArrayList<String>
getFrontSkus(): ArrayList<String>
getBackSkus(): ArrayList<String>
getFasciasList(): ArrayList<Fascia>
getUnsequencedFascia(): ArrayList<String>
addUnsequencedFascia(sku: String): void
renew(): void
getLocations(): ArrayList<String>
setLocation(location: ArrayList<String>)

SystemManage
- SERIALVERSIONID: long = 1L
- warehouse: String[]
- inputOrders: ArrayList<Order>
- inputPickingRequest: ArrayList<PickingRequest>
- loadedPickingRequest: ArrayList<PickingRequest>
- currTruck: Truck
- pickingList: ArrayList<PickingRequest>
- sequencingList: ArrayList<PickingRequest>
- sequencerFinishWork: ArrayList<Sequencer>
- systemHelper: SystemHelper
- workerFactory: WorkerFactory
Logger, Logger
consoleHandler: Handler
- orderTable: HashMap<String, ArrayList<String>
SystemManage()
createOrder(model: String, colour: String): void
createPickingRequest(): void
setWorkType(): String, name String): void
pickFascia(name: String): void
sendFasciaSequence(name: String): void
sequenceFascia(name: String): void
sequenceRescan(name: String): void
getNextLoadingFascia(): Sequencer
loadFascia(name: String): void
loaderRescan(name: String): void
replenishFascia(location: String): void
checkTruckStatus(): void
writeOrders(): void
writeMail(): void
getSystemHelper(): SystemHelper
getinputOrders(): ArrayList<Order>
getinputPickingRequest(): ArrayList<PickingRequest>
getLoadedPickingRequest(): ArrayList<PickingRequest>
setLoadedPickingRequest(pickingR: PickingRequest): void
getCurTruck(): Truck
getPickingList(): ArrayList<PickingRequest>
getSequencingList(): ArrayList<PickingRequest>
getSequencerFinishWork(): ArrayList<Sequencer>
getWorkerFactory(): WorkerFactory
setSequencerFinishWork(sequencer: Sequencer): void

Systemhelper
- traverse: String = "traverse_table.csv"
- translate: String = "translate.csv"
- initial: String = "initial.csv"
- warehouse: HashMap<String, Integer>
- locationMap: HashMap<String, String>
- orderTable: HashMap<String, ArrayList<String>>
- defaultAmount: int
- replaceAmount: int
Systemhelper(defaultAmount: int, replaceAmount: int)
initialize(): void
locationMap(): void
orderReader(): void
changeFasciaAmount(location: String, action: String): void
setTraverse(traverse: String): void
setTranslate(translate: String): void
setInitial(initial: String): void
getWarehouse(): HashMap<String, Integer>
getLocationMap(): HashMap<String, String>
getOrderTable(): HashMap<String, ArrayList<String>>
getDefaultAmount(): int
setDefaultAmount(DefaultAmount: int): void
setReplaceAmount(replaceAmount: int): void
setReplaceAmount(replaceAmount: int)

Truck
- volume: int=160
- contentTruck: ArrayList<ArrayList<String>>
- pickingRequests: ArrayList<PickingRequest>
- status: boolean
- lastReid: int
Truck()
loadFascia(currPickingRequest PickingRequest, frontFascia: ArrayList<String>, backFascia: ArrayList<String>): void
getContentTruck(): ArrayList<ArrayList<String>>
getStatus(): boolean
getLastReid(): int
getPickingContent(): ArrayList<PickingRequest>
leaveTruck(): void

Order
- colour: String
- model: String
- orderNum: String
- skus: ArrayList<String>
Order(colour: String, model: String, skus: ArrayList<String>, id: int)
getColour(): String
getModel(): String
toString(): String
getFrontSku(): String
getBackSku(): String
getOrderNum(): int