NodeStatus Free <<interface>> <<interface>> <<interface>> **IStartStop ITransportingNode IChainNode** <<interface>> <<interface>> <<interface>> <<interface>> **IProcessingNode** IMda **ICheckInDispatcher** IDropOff +Start():void +NodeId:int +Length: int +Stop():void +OnStatusChangedTofFree: Action +Capacity: int +MovingSpeed: int +AddNextNode(IChainNode node):void +AddNextNode(IChainNode node):void +NodeStatus: Status +Workers:int +PassBaggage(baggage Baggage):void +Process(baggage Baggage): void '-----, Mda CheckInDispatcher DropOff +DropOff(): ctor +Mda(): ctor +CheckInDispatcher(): ctor ChainNode +SetCheckIns(List<ICheckInDesks>):void +DispatchBaggage():void +NextNode: prop +Node(): ctor <<interface>> <<interface>> IFlight **IBaggage** +BaggaeAmount: int +DispatchedBaggage:int +FlightNumber: string +Type: BaggageType +Flight: string <<interface>> TransportingNode **IConveyor** ProcessingNode +Log: List<BaggageEventLog> +TransporterId: int +Gate: string +AddLog(logCreationtime: timespan, +DipartcherTime: Timespan -conveyorBelt: List<Baggage> #currentBagggage: IBaggage -lastBaggage: Baggage #allNextNodes: List<IChainNode> description): void -timer: Timer +ProcessingNode():ctor +TransportingNode(length: int): ctor +AddNextNode(IChainNode node) +Add():void +CanAdd(): Baggage Flight +Baggage(): ctor +Baggage(): ctor ChechInDesk Conveyor MainSecurity Conveyor():void MainSecurity():void CheckInDesk():void BaggageLog +Description: string +LogCreated: Timespan <<interface>> <<interface>> **ICheckInDesk IMainSecurity** +BaggageLog(): ctor <<enum>> BaggageType Small Medium Large