+ StateLogic <<interface>> logicObserver state : States - transitions : ArrayList<StateTransition> - events : HashMap<States, ArrayList<AbstractStateTransitionListener>> + updateTotal(int) : void + getState () : States + inState (state : States) : boolean + gotoState (next : States) : void + registerListener (state : States, listener : AbstractStateTransitionListener) : void + registerTransition (initial : States, end : States) : void + stateLogic - registerAll () : void <u>States</u> + valueOf(String) : States + values() (arrayList : States) + AddBagsLogic + approvedBagging : boolean + bagWeightLimit : Mass + addBagsLogic + endAddBags () : void + startAddBags () : void + CardPaymentLogic + dataRead : boolean + bank : cardIssuer + signature : String + getCardPaymentType (type : String) : PaymentMethods + isDataRead() : boolean + setDataRead(read : boolean) : void + approveTransaction (debitCardNumber : String, chargeAmount : double) : boolean + cardLogic + CardPaymentLogic(logic : CentralStationLogic, bank : CardIssuer) + CartLogic - balanceOwed : BigDecimal - cart : Map<Product, Integer> + updateBalance (balance : BigDecimal) : void + modifyBalance (amount : BigDecimal) : void + getBalanceOwed () : BigDecimal + calculateTotalCost () : BigDecimal + getCart () : Map<Product, Integer> + addBarcodedProductToCart (barcode : Barcode) : void + removeProductFromCart (product : BarcodedProduct) : void + addProductToCart (product : BarcodedProduct) : void + CurrencyLogic denominations : List<BigDecimal> + calculateChange (overpay : BigDecimal, availableCash : Map<BigDecimal, Integer>, includeUnavailable : boolean) : Map<BigDecimal, Integer> + banknoteCurrencyLogic/coinCurrencyLogic + getDenominationsAsList () : List<BigDecimal> + CentralStationLogic + CurrencyLogic(denominations : BigDecimal[]) + CurrencyLogic (denominations : List<BigDecimal>) + MembershipLogic - membershipCard : Card - accountName : String membershipNumber : String + getMembershipCard() : Card + createMembershipCard(cardType : String, cardNumber : String, cardholder : String, cvv : String, pin : String, isTapEnabled : boolean, hasChip : boolean) : void + setAccountame (accountName : String) : void getAccountName () : String + isMembershipNumberValid (membershipNumber : String) : boolean + membershipLogic + setMembershipNumber (cardNumber : String) : void + getMembershipNumber () : String + RemoveItemLogic + removeItemLogic removeBarcodedItem (product : BarcodedProduct) : void + WeightLogic bagMassTotal : Mass scaleOperational : boolean - sensitivity : Mass - actualWeight : Mass - expectedWeight : Mass + overrideDiscrepancy () : void + handleWeightDiscrepancy() : void + checkWeightDiscrepancy () : boolean + skipBaggingRequest (barcode : Barcode) : void + weightLogic + updateActualWeight (mass : Mass) : void removeExpectedWeight (barcode : Barcode) : void + addExpectedWeight (barcode : Barcode) : void + getExpectedWeight () : Mass + getTotalBagMass () : Mass + updateTotalBagMass (m : Mass) : void + getActualWeight () : Mass + AttendantLogic - inBaggingDiscrepency : boolean logic : CentralStationLogic - listeners : Set<AttendantFrameListener> + attendantCalled (guiLogic : GUILogic) : void + printDuplicateReceipt () : void + attendantLogic + grantApprovalSkipBagging (barcode : Barcode) : void + requestApprovalSkipBagging (barcode : Barcode) : void + getBaggingDiscrepency () : boolean + setBaggingDiscrepency (b : boolean) : void + baggingDiscrepencyDetected () : void + approveBaggingArea () : void disableCustomerStation() : void + enableCustomerStation() : void # confirmCall(guiLogic : GUILogic) : void

+ added(GUILogic, pk: PLUCodedProduct): void
+ scanned(GUILogic, pk: BarcodedProduct): void
+ ownBags(GUILogic, ownBags: boolean): void
+ attendantCalled(GUILogic): void
+ payOption(GUILogic, payOption: int): void
+ memberLogin(GUILogic, id: String): void

<<interface>> GUIListener