StateMachine

protected:
+ currentState : unsigned char

private:
+ m_maxStates : const unsigned char
+ m_eventGenerated : bool
+ m_pEventData : EventData*

protected:
+ externalEvent(char,EventData*) : void
+ internalEvent(char,EventData*) : void

private:
+ StateEngine() : void

VendingMachine

private:
+ m_transactionCash : float
+ m_interface : Interface*
+ productDatabase : Product*
+ logVendingMachine : Log

public:
+ cancelEvent() : void
+ cashIncrementEvent(float) : void
+ productSelectionEvent(int) : void

private:
+ ST_Idle(EventData*) : void
+ ST_Devolution(EventData*) : void
+ ST_Validation(VendingMachineData*) : void
+ ST_Transaction(VendingMachineData*) : void
+ ST_Deployment(VendingMachineData*) : void

Product

public:
+ProductID: enum

private:
+ logProduct : Log
+ m_name : string
+ m_value : float
+ m_stock : int
+ m_id : ProductID

private:
+ setId(ProductID) : void

public :
+ getName() : string
+ getValue() : float
+ getStock() : int
+ setName(string) : void
+ setValue(float) : void
+ setStock(int) : void
+ getProductDatabase(vector) : void
+ setProductDatabase(Product*,vector) : void

Interface

public:
+getSystemInput(SystemData*) : virtual void
+setSystemOutput(SystemData*) : virtual void

+getUserInput(UserData*) : virtual void
+setUserOutput(UserData*) : virtual void

+printAdvertising(AdvertisingData*) : virtual void
+insertAdvertising(AdvertisingData*) : virtual void

Log

public:
+ Level : enum

private:
m_logLevel : Level
m_scope : string

public:
+ setLevel(Level) : void
+ setScope(string) : void
+ error(string) : void
+ warn(string) : void
+ info(string) : void
+ debug(string) : void

Node

private:
+ m_nodeData : int
+ m_nextNode : Node*

public:
+ getNodeData() : int
+ getNextNode() : Node*
+ setNodeData(int) : void
+ setNextNode(Node*) : void

Queue

private:
+ m_head : Node

public:
+ pop() : int
+ push(int) : void

DebugInterface

+ field: type

+ method(type): type

EventData

+ field: type

+ method(type): type