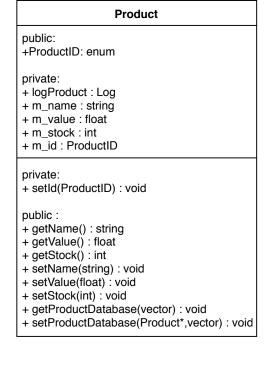
StateMachine VendingMachine private: + currentState : unsigned char + m transactionCash: float + m interface: Interface* + productDatabase : Product* + m_maxStates : const unsigned char + logVendingMachine : Log + m eventGenerated : bool + m pEventData : EventData* public: + cancelEvent(): void + cashIncrementEvent(float): void + productSelectionEvent(int): void + externalEvent(char,EventData*): void + internalEvent(char.EventData*) : void private: + ST_Idle(EventData*): void + ST Devolution(EventData*): void + ST_Validation(VendingMachineData*): void + ST_Transaction(VendingMachineData*): void + ST_Deployment(VendingMachineData*): void



Interface

public:

protected:

private:

protected:

private:

+ StateEngine(): void

+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

