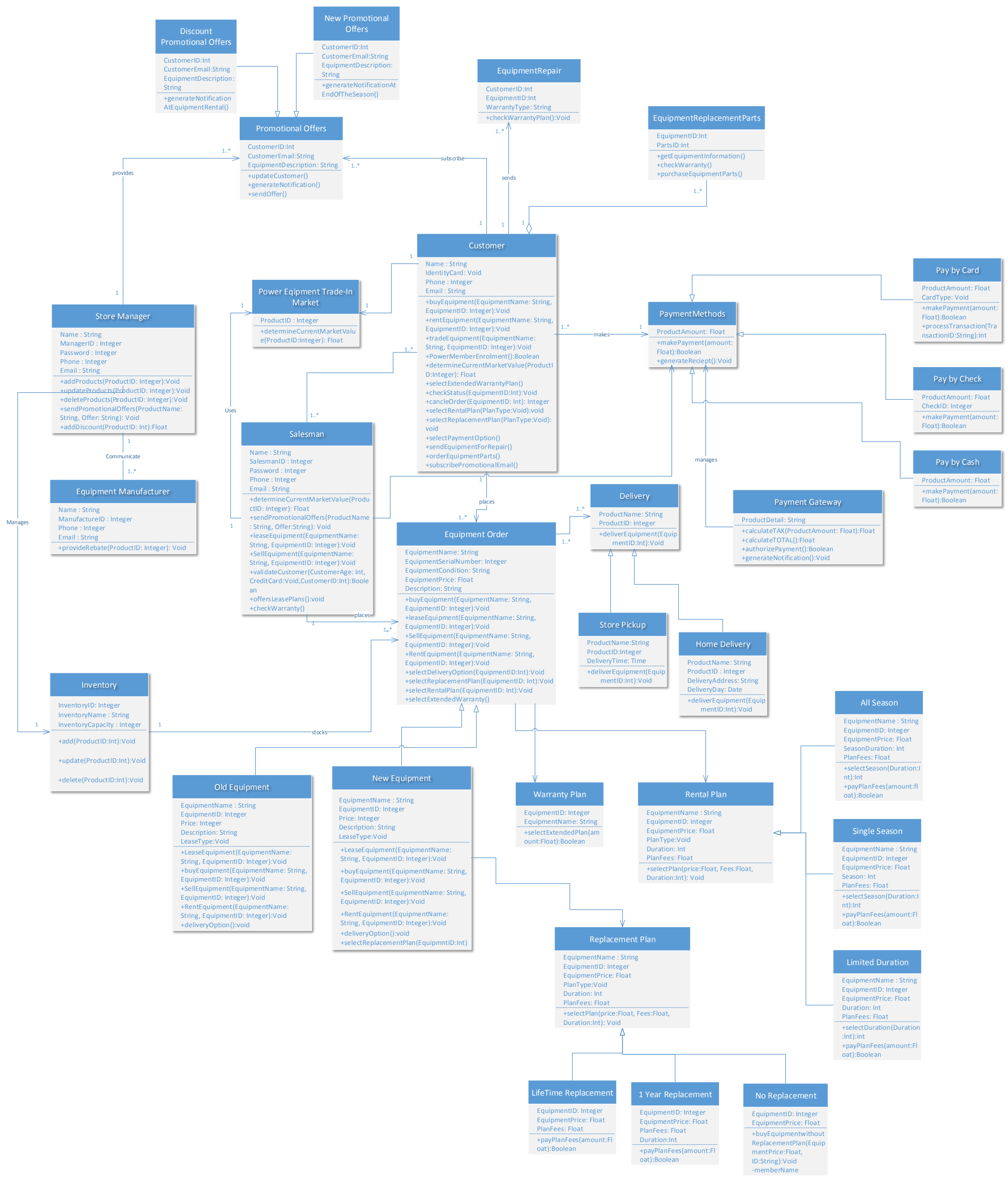
NOTE: I have not expand any class in Design Class diagram, For all classes which I have used in 3 design patterns, I have expand them in design pattern only.

1. Complete List of classes in Design model:

- Customer
- Store Manager
- Salesman
- Equipment Manufacturer
- Equipment Order
  - Old Equipment
  - New Equipment
- Inventory
- Power Equipment Trade-In Market
- Delivery
  - Store Pickup
  - Home Delivery
- Payment Methods
  - Pay by Cash
  - Pay by Check
  - Pay by Card
- Payment Gateway
- Warranty Plan
- Rental Plan
  - All Season
  - Single Season
  - Limited Duration
- Replacement Plan
  - Life Time Replacement
  - 1 Year Replacement
  - No Replacement
- Repair Equipment
- Promotional Offers
  - Discount Promotional Offers
  - New Promotional Offers

2. Complete UML Design Model (No Class expanded for pattern)


Discount Promotional Offers
CustomerID:Int
CustomerEmail:String
EquipmentDescription: String
+generateNotificationAtEquipmentRental()
New Promotional Offers
CustomerID:Int
CustomerEmail:String
EquipmentDescription: String
+generateNotificationAtEndOfTheSeason()
EquipmentRepair
CustomerID:Int
EquipmentID:Int
WarrantyType: String
+checkWarrantyPlan():Void
EquipmentReplacementParts
EquipmentID:Int
PartsID:Int
+getEquipmentInformation()
+checkWarranty()
+purchaseEquipmentParts()
Promotional Offers
CustomerID:Int
CustomerEmail:String
EquipmentDescription: String
+updateCustomer()
+generateNotification()
+sendOffer()
provides
subscribe
sends
Customer
Name : String
IdentityCard: Void
Phone : Integer
Email : String
+buyEquipment(EquipmentName: String, EquipmentID: Integer):Void
+rentEquipment(EquipmentName: String, EquipmentID: Integer):Void
+tradeEquipment(EquipmentName: String, EquipmentID: Integer):Void
+PowerMemberEnrolment():Boolean
+determineCurrentMarketValue(ProductID:Integer): Float
+selectExtendedWarrantyPlan()
+checkStatus(EquipmentID:Int):Void
+cancleOrder(EquipmentID: Int): Integer
+selectRentalPlan(PlanType:Void):void
+selectReplacementPlan(PlanType:Void): void
+selectPaymentOption()
+sendEquipmentForRepair()
+orderEquipmentParts()
+subscribePromotionalEmail()
Power Eqipment Trade-In Market
ProductID : Integer
+determineCurrentMarketValue(ProductID:Integer): Float
Store Manager
Name : String
ManagerID : Integer
Password : Integer
Phone : Integer
Email : String
+addProducts(ProductID: Integer):Void
+updateProducts(ProductID: Integer):Void
+deleteProducts(ProductID: Integer):Void
+sendPromotionalOffers(ProductName: String, Offer: String): Void
+addDiscount(ProductID: Int):Float
Communicate
Equipment Manufacturer
Name : String
ManufactureID : Integer
Phone : Integer
Email : String
+provideRebate(ProductID: Integer): Void
Manages
Uses
Salesman
Name : String
SalesmanID : Integer
Password : Integer
Phone : Integer
Email : String
+determineCurrentValue(ProductID: Integer): Float
+sendPromotionalOffers(ProductName : String, Offer:String): Void
+leaseEquipment(EquipmentName: String, EquipmentID: Integer):Void
+SellEquipment(EquipmentName: String, EquipmentID: Integer):Void
+validateCustomer(CustomerAge: Int, CreditCard:Void,CustomerID:Int):Boolean
+offersLeasePlans():void
+checkWarranty()
places
PaymentMethods
ProductAmount: Float
+makePayment(amount: Float):Boolean
+generateReciept():Void
makes
manages
Pay by Card
ProductAmount: Float
CardType: Void
+makePayment(amount: Float):Boolean
+processTransaction(TransactionID:String):Int
Pay by Check
ProductAmount: Float
CheckID: Integer
+makePayment(amount: Float):Boolean
Pay by Cash
ProductAmount: Float
+makePayment(amount: Float):Boolean
Payment Gateway
ProductDetail: String
+calculateTAX(ProductAmount: Float):Float
+calculateTOTAL():Float
+authorizePayment():Boolean
+generateNotification():Void
Delivery
ProductName: String
ProductID: Integer
+deliverEquipment(EquipmentID:Int):Void
Equipment Order
EquipmentName: String
EquipmentSerialNumber: Integer
EquipmentCondition: String
EquipmentPrice: Float
Description: String
+buyEquipment(EquipmentName: String, EquipmentID: Integer):Void
+leaseEquipment(EquipmentName: String, EquipmentID: Integer):Void
+SellEquipment(EquipmentName: String, EquipmentID: Integer):Void
+RentEquipment(EquipmentName: String, EquipmentID: Integer):Void
+selectDeliveryOption(EquipmentID:Int):Void
+selectReplacementPlan(EquipmentID: Int):Void
+selectRentalPlan(EquipmentID: Int):Void
+selectExtendedWarranty()
Store Pickup
ProductName:String
ProductID:Integer
DeliveryTime: Time
+deliverEquipment(EquipmentID:Int):Void
Home Delivery
ProductName: String
ProductID : Integer
DeliveryAddress: String
DeliveryDay: Date
+deliverEquipment(EquipmentID:Int):Void
Inventory
InventoryID: Integer
InventoryName : String
InventoryCapacity : Integer
+add(ProductID:Int):Void
+update(ProductID:Int):Void
+delete(ProductID:Int):Void
stocks
Old Equipment
EquipmentName : String
EquipmentID: Integer
Price: Integer
Description: String
LeaseType:Void
+LeaseEquipment(EquipmentName: String, EquipmentID: Integer):Void
+buyEquipment(EquipmentName: String, EquipmentID: Integer):Void
+SellEquipment(EquipmentName: String, EquipmentID: Integer):Void
+RentEquipment(EquipmentName: String, EquipmentID: Integer):Void
+deliveryOption():void
New Equipment
EquipmentName : String
EquipmentID: Integer
Price: Integer
Description: String
LeaseType:Void
+LeaseEquipment(EquipmentName: String, EquipmentID: Integer):Void
+buyEquipment(EquipmentName: String, EquipmentID: Integer):Void
+SellEquipment(EquipmentName: String, EquipmentID: Integer):Void
+RentEquipment(EquipmentName: String, EquipmentID: Integer):Void
+deliveryOption():void
+selectReplacementPlan(EquipmntID:Int)
Warranty Plan
EquipmentID: Integer
EquipmentName: String
+selectExtendedPlan(amount:Float):Boolean
Rental Plan
EquipmentName : String
EquipmentID: Integer
EquipmentPrice: Float
PlanType:Void
Duration: Int
PlanFees: Float
+selectPlan(price:Float, Fees:Float, Duration:Int): Void
All Season
EquipmentName : String
EquipmentID: Integer
EquipmentPrice: Float
SeasonDuration: Int
PlanFees: Float
+selectSeason(Duration:Int):Int
+payPlanFees(amount:Float):Boolean
Single Season
EquipmentName : String
EquipmentID: Integer
EquipmentPrice: Float
Season: Int
PlanFees: Float
+selectSeason(Duration:Int):Int
+payPlanFees(amount:Float):Boolean
Limited Duration
EquipmentName : String
EquipmentID: Integer
EquipmentPrice: Float
Duration: Int
PlanFees: Float
+selectDuration(Duration:Int):int
+payPlanFees(amount:Float):Boolean
Replacement Plan
EquipmentName : String
EquipmentID: Integer
EquipmentPrice: Float
PlanType:Void
Duration: Int
PlanFees: Float
+selectPlan(price:Float, Fees:Float, Duration:Int): Void
LifeTime Replacement
EquipmentID: Integer
EquipmentPrice: Float
PlanFees: Float
+payPlanFees(amount:Float):Boolean
1 Year Replacement
EquipmentID: Integer
EquipmentPrice: Float
PlanFees: Float
Duration:Int
+payPlanFees(amount:Float):Boolean
No Replacement
EquipmentID: Integer
EquipmentPrice: Float
+buyEquipmentwithoutReplacementPlan(EquipmentPrice:Float, ID:String):Void
-memberName

3. List of design patterns used.

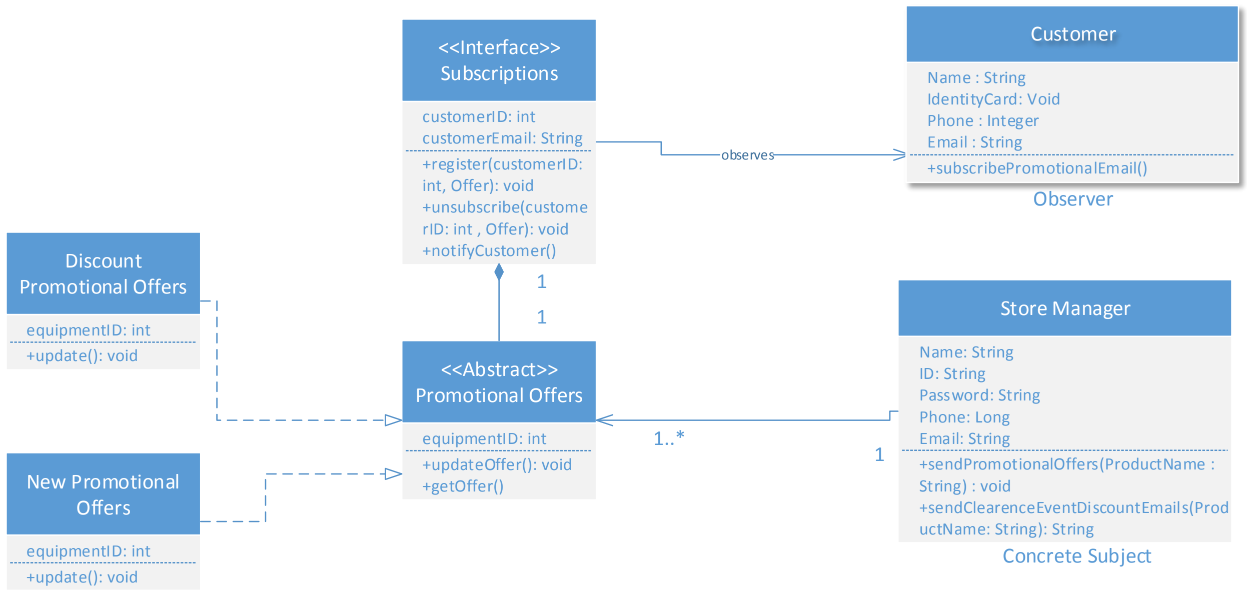
NOTE: Here I have expanded all relevant classes related to each design patterns.

Patterns Used:

1. Observer Pattern: Used on E-mail Subscription module
2. Strategy Pattern: Used on Replacement Plan module
3. Factory Pattern: Used on Payment module

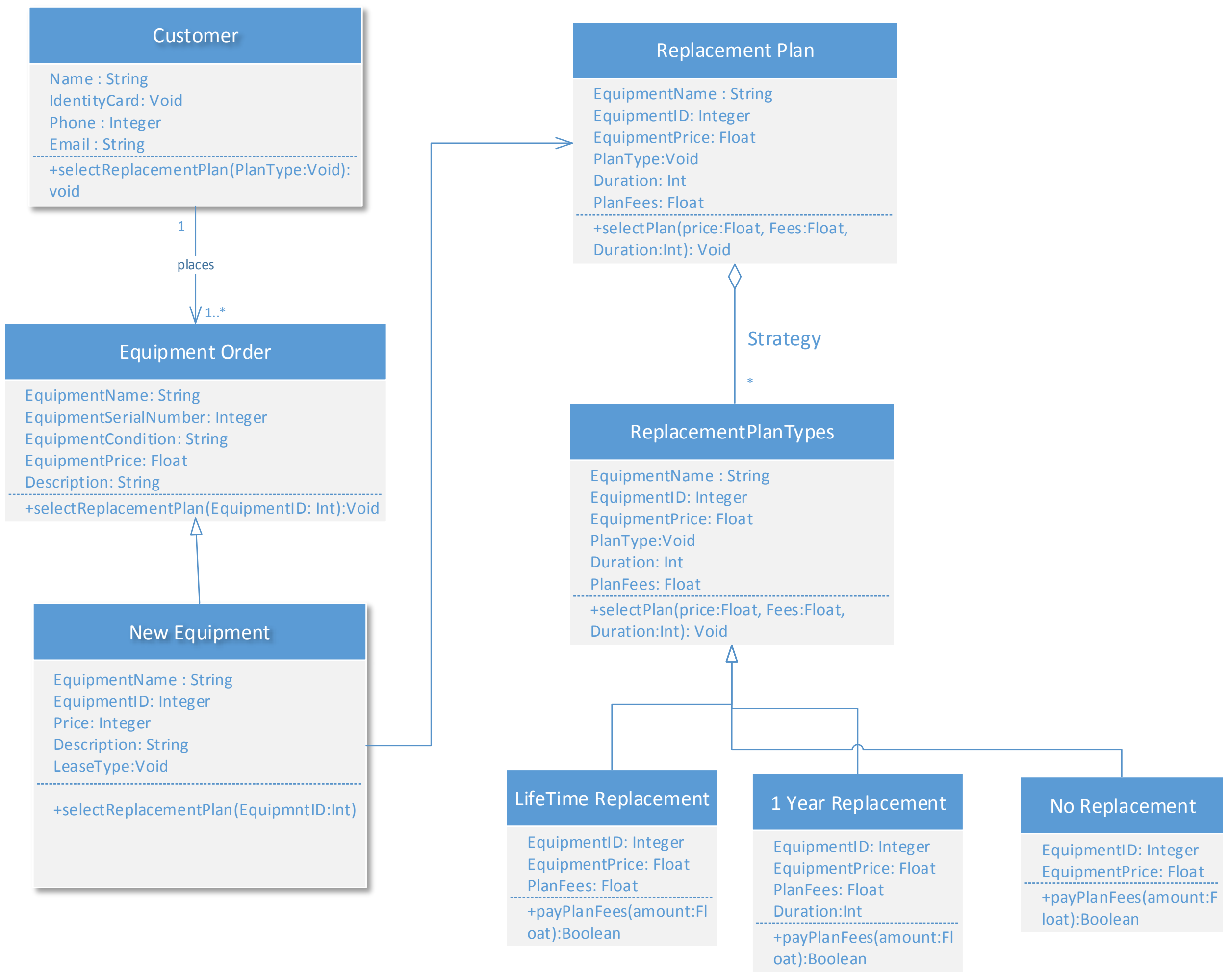
4. Documentation on design pattern usage.

- Observer Pattern:



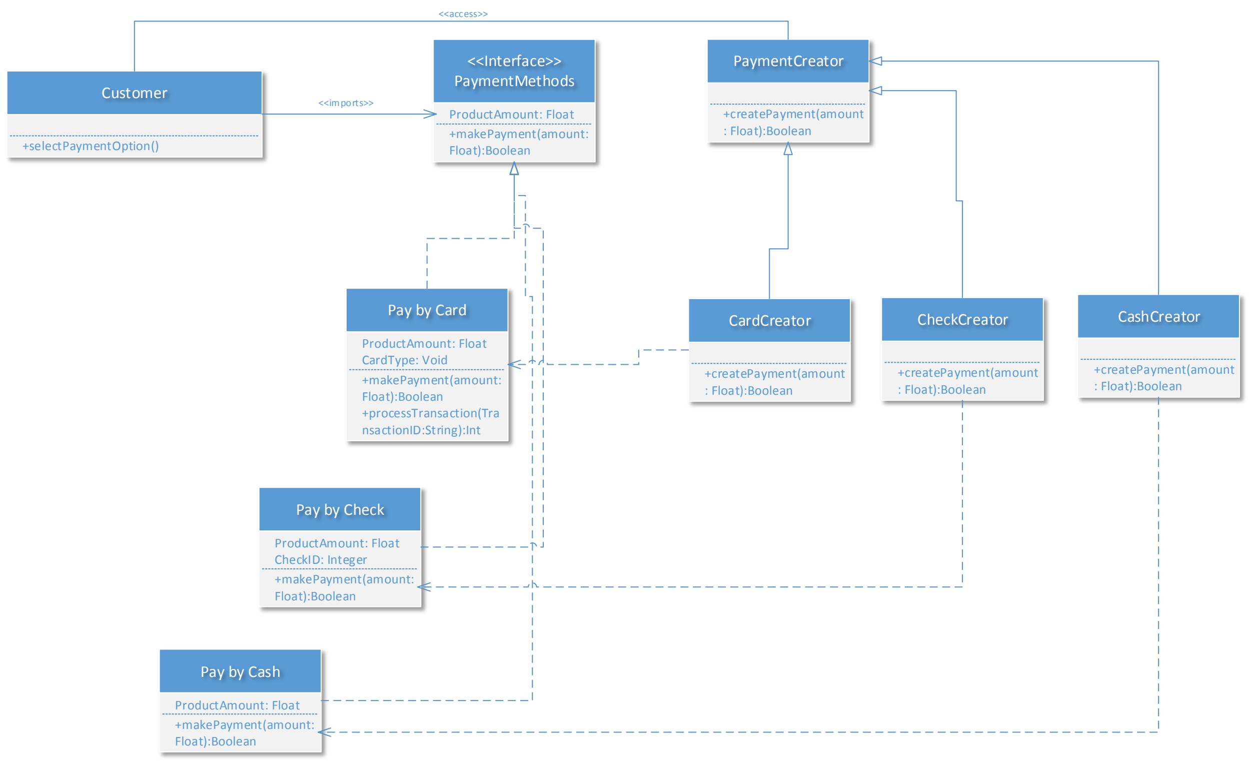
Here, Customer is the Observer, who use the Interface "Subscriptions" to get subscribe for promotional offers, Store manager generate promotional offers on different occasion, for ex. At the end of every season store manager sends 20% discount for clearance. Every time for storage space limitation threshold violate, storemanager sends 25% promotional offer to customer. Subscription Interface provide the customer to Register, unsubscribe from the email subscription. Promotional class provide updated offer to Interface.

- Strategy Pattern:



Here, any replacement plan can be selected from the three available plan which suited best to customer's needs while buying a NEW equipment. Strategy pattern is utilized on the above classes to show the proper strategy to choose appropriate plan for appropriate equipment & condition selected.

- Factory Pattern:


<<access>>
Customer
+selectPaymentOption()
<<imports>>
<<Interface>> PaymentMethods
ProductAmount: Float
+makePayment(amount: Float):Boolean
PaymentCreator
+createPayment(amount : Float):Boolean
Pay by Card
ProductAmount: Float
CardType: Void
+makePayment(amount: Float):Boolean
+processTransaction(TransactionID:String):Int
CardCreator
+createPayment(amount : Float):Boolean
CheckCreator
+createPayment(amount : Float):Boolean
CashCreator
+createPayment(amount : Float):Boolean
Pay by Check
ProductAmount: Float
CheckID: Integer
+makePayment(amount: Float):Boolean
Pay by Cash
ProductAmount: Float
+makePayment(amount: Float):Boolean

Here, Factory Pattern is implemented on Payment module class where we have various option to pay for the services we have used or items we have bought. All Payment methods have creator and they are connected to each of them. Payment Method is the Interface which provide various interface of payment type. Customer use this interface to select appropriate payment method suited best to his situation.