Food Delivery Design

<https://github.com/gmershad/FoodDeliveryApp>

Design Patterns involved:

1) Builder Design Pattern (For adding food item and ordering)

2) Interpreter Design Pattern (User can Search Different restaurant)

3) Iterator Pattern (User Sees Menu)

4) Observer Pattern(Track an order in Real Time)

5) Command Pattern(Order or Cancellation of Foo)

We can consider few more conditions-

Restaurant has a Menu (Menu class)

Menu has items (Item class)

User can order (Order class)

Order can have many selectedItems(SelectItem class)

User has to pay the BIll (Bill class)

There also be a Payment class(That must be thread safe)

Total - 8 classes

Restaurant

User

Menu

Item

Order

SelectItem

Bill

Payment

Lets see classes one by one -

Restuarant

-----------

- name : String

- location : string

- conatct : string

-----------

+ isOpen() : boolean

+ getName() : String

+ getLoaction() : String

+ getContact() : String

+ setContact()

+ displayMenu()

+ updateMenu)\_

+ generateBill()

User

-----------

- uid : String

- contact : String

-----------

+ setContact()

+ getContact()

+ getUid()

+ searchRestauranr()

+ selectRestuarant()

+ newOrder()

+ cancelOrder()

+ trackOrder()

+ payBill()

Menu

---------

- menu : List<Item>

---------

+ displayMenu()

+ updateMenu()

+ addItem()

+ deleteItem()

Item

----------

- itemId : String

- price : float

- name : String

----------

+ setItemId(String)

+ setName(String)

+ setPrice(float)

+ getItemId()

+ getName()

+ getPrice()

Order

-----------

- orderId : String

-----------

+ getStatus() // To track the order

+ updateStatus() // Restaurant admin can update it

+ getOrderId()

+ selectItem()

+ removeItem()

+ updateItem()

SelectItem

-----------

- itemId

- quantity : int

-----------

+ getItem()

+ setItem()

+ getQuantity()

+ setQuantity()

Bill

------------

- billNum

- orderId

------------

+ getBillNum()

+ getTotalAmount()

+ displayBill()

Payment

------------

- pid : String

- pType : String

- billNum : String

-------------

+ successfulyPaid() : boolean