**1.DominozPizzaDelivery**

Implement the functionality to read all the data from topping.txt and price.txt

Topping.txt contains -🡪 KeyId(1 for veg and 2 foe nonveg) , toppingid, topping name ,timetocook

Price.txt contains -🡪 toppingid, topping price

Implement the functionality to read all the data from dish.txt and location.txt

Method Name : **Map<key , Map<topping , price>> populateData(String toppingFile,String priceFile);**

Classes Provided: Bean class called **topping , price , order**

Note:Please do not change the method signature.

You can add more members to bean or client,as needed.

Marks: 15

**2.CalculateOrder**

Implement the functionality to read the data from the order.txt file

Order.txt contains Keyid(veg -1 or nonveg-2 ) , toppingid , pizza size , pizza topping

Method Name : **List<Order> calculateOrder(String orderFile, Map<key , Map<topping , price>> map);**

Classes Provided: Bean class called **topping , price , order**

Note:Please do not change the method signature.

You can add more members to bean or client,as needed.

Marks: 5

**3. choose an order**

Implement the functionality to randomly choose an order from the list returned in above method2

Method Name : **Order chooseOrder(list<order> order);**

Classes Provided: Bean class called **topping , price , order**

Note:Please do not change the method signature.

You can add more members to bean or client,as needed.

Marks: 10

**4.checkOrder**

Implement the check whether the type of order returned by method3 is available or not.

If found return it else throw NoSuchOrderFound exception

Method Name : **Order CheckOrder(Order order, Map<key , Map<topping , price>> map);**

Classes Provided: Bean class called **topping , price , order**

Note:Please do not change the method signature.

You can add more members to bean or client,as needed.

Marks: 5

**5. CalculateBill**

Implement the functionality to calculate bill of the order returned in method 4.

Bill will be acc to the pizza size and type of topping and offcourse the type of pizza.

If the pizza size is regular and timetocook is less then 30 mins then cost = topping price only

Else if timetocook is more then 30 mins , cost = topping price + time to took \* 1 rs ;

If pizza size is medium den cost = twice of topping price

And for large pizza it is thrice the topping price;

And timetocook constraint is same for medium and large pizza as it was for regular. That is if time > 30mins then add d time \*1rs in the cost.

Method Name : **void calculateBill(Order order, Map<key , Map<topping , price>> map);**

Classes Provided: Bean class called **topping , price , order**

Note:Please do not change the method signature.

You can add more members to bean or client,as needed.

Marks: 10