1. Add a new product

In:

Id? : String  
 description? : String

Out:

NONE

Pre:

//id? Is new  
 not exists p in prdouctList | p.id = id?

Post:

//add new product  
 let p = new Product(id?, description?)|  
 p.id = id?  
 p.description = description?  
 add p to ProductList

1. Add a new customer with one delivery address

In:

customerId? : String  
 name? : String  
 addressId? : String  
 lineOne? : String  
 lineTwo? : String  
 contactPerson? : String  
 contactPhone? : String

Out:

NONE

Pre:

//customerId? Is unique  
not exists c in customerList | c.id = customerId?  
//addressId? Is unique  
not exists a in addressList | a.id = addressId?

Post:

//create new address  
 let a = new Address(addressId?, lineOne?, lineTwo?, contactPerson?, contactPhone?)|  
 a.id = addressId?  
 a.address = address?  
 a.contactPerson = contactPerson?  
 a.contactPhone = contactPhone?  
 add a to addressList  
 //create a new customer  
 let c = new Customer(customerId?, name?, a)|  
 c.id = customerId?  
 c.name = name?  
 addresses = new HashSet<Address>()  
 order = new HashSet<Order>()  
 address.add(a)  
 add c to customerList

1. Add a delivery address to an existing customer

In:

customerId? : String  
 addressId? : String  
 address? : String  
 contactPerson? : String  
 contactPhone? : String

Out:

NONE

Pre:

//addressId? Is new  
not exists a in addressList | a.addressId = addressId?  
//customerId? exists  
exists c in CustomerList|  
 c.id = customerId?

Post:

Let customer = element in customerList | customer.id = customerId?  
  
 let a = new Address(addressId?, address?, contactPerson?, contactPhone?)|  
 a.Id = addressId?  
 a.address = address?  
 a.contactPerson = contactPerson?  
 a.contactPhone = contactPhone?  
  
 add a to customer.addresses  
 add a to addressList

1. Add a standing order to an existing customer

In:

orderId? String  
customerId? : String  
addressId? : String  
productId? : String  
price? : Real  
quantities? : List<Integer>  
startDate? : Integer  
endDate? : Integer  
status? : Status

Out:

NONE

Pre:

//orderId? Is new  
not exists o in orderList | o.orderId = orderId?  
//customerId? exists  
exists c in CustomerList|  
 c.id = customerId?  
//addressId? Exists  
exists a in addressList|  
 a.id = addressId?  
//productid? Exists  
exists p in productList|  
 p.id = productid?  
//start date is before end date  
startDate? < endDate?  
//quantities length is equal to seven  
quantities.size() = 7

Post:

Let customer = element in customerList | customer.id = customerId?  
 let address = element in addressList | address.id = addressId?  
 let product = element in productList | product.id = productid?  
 let order = new Order(orderId?, customer, address, product, price?, quantities?, startDate?, endDate?, status?)|  
 order.id = orderId?  
 order.customer = customer  
 order.address = address  
 order.product = product  
 order.price = price?  
 order.quantities = quantities?  
 order.startDate = startDate?  
 order.endDate = endDate?  
 order.status = status?  
 add order to orderList  
 add order to customer.orders

1. List standing orders for a date, sorted by customer id, within a name, by the customer id

In:

date? : Integer

Out:

order!

Pre:

NONE!

Post:

For each order in orderList  
 then  
 order.getStartDate < date AND  
 order.getEndDate > date  
 {  
 add order to order!  
 }  
 order! = sortByCustomerName(order!)  
 order! = sortByCustomerId(order!)

1. Add a delivery

In:

deliveryId? : String  
 customerId? : String   
 addressId? : String  
 date? : Integer  
 orderId? : String  
 quantity? : Integer

Out:

NONE

Pre:

//deliveryId? Is new  
not exists d in deliveryList | d.Id = deliveryId?  
//customerId? exists  
exists c in CustomerList|  
 c.id = customerId?  
//addressId? Exists  
exists a in addressList|  
 a.id = addressId?

Post:

Let customer = element in customerList | customer.id = customerId?  
let address = element in addressList | address.id = addressId?  
let order = element in orderList | order.id = orderId?  
let deliveryItem = new DeliveryItem(order, quantity?,order.getQuantity(date?%7)-quantity)  
Let delivery = new Delivery(deliveryId?, customer, address, date?, deliveryItem)|  
 delivery.id = deliveryId?  
 delivery.customer = customer  
 delivery.address = address  
 delivery.date = date?  
 delivery.dayOfWeek = (date?%7)  
 delivery.deliveryItems = new HeshSet()  
 deliveryItems.add(deliveryItem)

add delivery to deliveryList