

Data Structure

For our exam we were expected to make a system where we handle the adding, removal and viewing of stock items that have a product ID, name, weight (quantity) and expiration date. Since the items have to be added in and removed dynamically, without knowing an exact number of items that will be in the list at a given time, the best data structure to manage the items would be an ArrayList.

When an ArrayList hits its capacity, it doubles in size and allows for continuous adding of items. ArrayLists also allows us to sort items, iterate through the items and fetch and store data as we need going as far as to add or remove items at a targeted specific index. The ItemBatch class implements a Comparable<ItemBatch> so that we are able to compare the expiration data and take the stock from the closer expiry.

Products are added first and then a batch of that product can be added with its own specific quantity in weight and an expiry date. In this way even if we have the same item with two expiry dates, when we consume the product, the one with the closer date is exhausted before moving to the next product. Since products are bundled together, the system allows you to add a product as 'product name - product ID' and achieve the same goal as having a way to set the ID separately.

Pseudocode

Freezer Class

VAR:

FoodItems: ArrayList

Public Freezer() //Constructor for the freezer class

Food Class

VAR:

Batches: ArrayList

name: String

totalWeight: double

weight: double

consumed: double
index: int

Public Food(name: string) //Constructor for food object

PROMPT add batch to Batches and sort

INPUT batch

LOOP for batch //in order to get total weight

 SET totalWeight = batches weight

ENDLOOP

OUTPUT totalWeight

SET consumed = 0 //To allow consuming

SET index = 0

LOOP while consumed < weight

 GET batches index

IF batch weight > weight

 SET batch weight -= weight - consumed

IF batch weight < 0

 Remove item from batch

ELSE IF batch weight < weight

 Display error message

ELSE SET consumed += batch weight

 Remove item from batch

Batch Class

VAR

batchID: String

foodItem: Food

expiryDate: int

weight: double

Batch(weight: double, expiry: int) //Batch constructor

Test Cases

#	Test Case	Expected Result	Actual Result	Status
1	Add one product	Adds product into list	Product is added into list	Works as intended
2	Add multiple products	Adds all products into list	Products are added into list	Works as intended
3	Add product with field empty	Error message should display asking to enter food	Error message displays asking to enter food	Works as intended
4	View products added	Should be able to see all products currently in list	Able to see all products currently in list	Works as intended
5	Add a new batch into a product	Should be able to add a batch with expiry and weight	Able to add a batch with expiry and weight	Works as intended

6	Add new batch with no weight	Should trigger an error message asking for it	Error message displays	Works as intended
7	Add new batch with no expiry	Should trigger an error message asking for it	Error message displays	Works as intended
8	Consume a product where consumed weight is less than or equal to available weight	Should be able to consume and see that the consumed weight is deducted from the total	Able to consume and see that the consumed weight is deducted from the total	Works as intended
9	Consume a product where consumed weight is more than the available weight	Should not be able to consume the product and an error should display on CLI	Not able to consume the product and an error displays on CLI	Works as intended
10	Delete a food	Food should delete from the table	Food deletes from the table	Works as intended