|  |  |  |  |
| --- | --- | --- | --- |
| Class | Methods | Scenario | Output |
| 1. Inventory | AddType(Bagel bagel1) | Add a bagel type | Returns true |
|  |  | Doesn’t add a bagel type | Returns false |
|  | removeBagel (Bagel bagel99) | Removes a bagel from the list | Returns true |
|  |  | Doesn’t remove bagel from list | Returns false |
|  | checkCapacity(List<Bagel> bagels) | Adding failed due to full capacity | Returns string “basket full!” |
|  |  | Adding success due to not full capacity | Returns string”Item added” |
|  | changeCapacity(List<Basket> basket) | Changes capacity of basket | Returns true |
|  |  | Didn’t change capacity of basket | Returns false |
|  | checkExists(Bagel bagelExist) | Removes an item that doesn’t exist | Returns string “Item doesn’t exist” |
|  |  | Removes item that exists | Returns “Item removed successfully” |
|  | sumItems(List<Bagel> bagels) | Gives customer the total cost of items | Returns sum of totalcost of items (double) |
|  |  | Fails to give total cost of items | Returns 0; |
|  | viewCost(Bagel bagelCost) | Shows cost of bagel | Returns double |
|  |  | Fails to show cost of bagel | Returns 0 |
|  | addFilling(Bagel bagel) | Adds a filling to bagel | Returns bagel’s filling |
|  |  | Doesn’t add filling | Returns “no filling” |
|  | costFilling(Bagel bagel) | Shows cost of filling | Returns cost of filling |
|  |  |  | Returns 0 |
|  | checkIfOrderinList(Bagel bagel) | Item in list | Returns true |
|  |  | Item not in list | Return false |