Names:

Talhah Waheed

Kamen Petkov

Class design table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Class name  Inventory | | |  | | | |  | |
| Attributes | | | | | | | | |
| Name | type | Static [YES/NO] | | Access modifier | Initial value if any | Range of possible values | | Notes / comments |
| ingredients | double[] | no | | private | All elements set to 300.0d initially | Each element can be a double value between 0.0d and 300.0d (both included) | | An array of doubles. Set only once in the constructor. |
| ingredientsNames | String[] | no | | private | Initialize each element to a different ingredient name | N/A | | An array of Strings. Set only once in the constructor. |
| ingredientCosts | double[] | no | | private | All elements set to a number between 0.0d and 1.0d | Each element can be a double value between 0.0d and 1.0d (both included) | | An array of doubles. Set only once in the constructor. |
| recipes | double[][] | no | | private | N/A | The ingredients should be set to a value between 0.0d and 2.0d. | | This is a double array that shows the amount of ingredients used for each type of taco. Set only once in the constructor. |
| shells | int | no | | private | 200 | Between 0 and 200. | | Decrement every time a new order is placed. |
|  |  |  | |  |  |  | |  |
| Methods (including constructors) | | | | | | | | |
| Name | Return type | Static [YES/NO] | | Access modifier |  | List of arguments and their types | | Notes / comments |
| Inventory() | N/A | no | | public |  | None | | Initializes all attributes. |
| getShells() | int | no | | public |  | None | | Returns the amount of shells. |
| getIngredientNames() | String[] | no | | public |  | None | | Returns the names of the ingredients. |
| getIngredientCosts() | double[] | no | | public |  | None | | Returns the cost of the ingredients. |
| getIngredients() | double[] | no | | public |  | None | | Returns the amount of ingredients. |
| getRecipes() | double[] | no | | public |  | int menuItemID | | Returns the amount of ingredients needed for the different types of tacos. |
| checkIfEnough() | boolean | no | | public |  | double[] ingredientsNeeded | | Checks if there are enough ingredients for the taco ordered. Returns true or false. |
| update() | void | no | | public |  | Order newOrder | | Updates the ingredients array and shells attribute using the newOrder object information. |
| showSummary() | void | no | | public |  | N/A | | Displays the current ingredients summary. |

Class / object interactions design table (A class / object interacts (calls a method of) with class/object B):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class / Object A | calls method (it can be constructor to create an object) | of class / object B with arguments (if any) | to | Notes: |
| Inventory | Constructor | Order newOrder | get the information that is set in order class: orderId, menuID, cost, price, ingredients, etc. | - |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Test case design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case name | Tested class | Tested method | Test input | Expected outcome |
| check amount of ingredients | Inventory | getIngredients() | Test for amounts from 0.0d to 300.0d.  Check what happens for values different from the one mentioned above. | If it’s between 0.0 and 300.0d the app should work properly.  If not: there should be a message that says that something is wrong. |
| check the ingredient cost | Inventory | getIngredientCost() | Test if it’s positive. | If it’s positive, the program is working properly.  If it returns a negative value, something is wrong and we should get a message that something is wrong. |
| check the ingredients used for each recipe | Inventory | getRecipes() | Test if the value is between 0.0d and 2.0d. | The program should work fine if it’s in this range. Otherwise, we should get a message that something is wrong. |
| check the amount of shells | Inventory | getShells() | Test if it’s more than 0. | If the amount of shells is < 0 there is a problem. Get a message that something is wrong. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Class name  Order | | |  | | | |  | |
| Attributes | | | | | | | | |
| Name | type | Static [YES/NO] | | Access modifier | Initial value if any | Range of possible values | | Notes / comments |
| orderIDCount | int | YES | | private | 0 | Between 0 and 199 (both included) | | This attribute will keep track of the number of tacos ordered. It is incremented every time a new order object is created. |
| orderID | int | NO | | private | - | Between 0 and 199 (both included) | | It is set once in the constructor. Based on the current orderIDCount value. |
| menuID | int | NO | | private | - | Possible values: 0, 1, 2, 3 | | Set once in the constructor. |
| cost | double | NO | | private | - | Positive based on the formula: cost = taco shell cost + ingredient 1 amount \* ingredient 1 cost + …. + ingredient 5 amount \* ingredient 5 cost. | | Set only once in the constructor. |
| price | double | NO | | private | - | Positive, based on the formula: price = 1.20 \* cost | | Set only once in the constructor. |
| ingredients | double[] | NO | | private | - | An array of the size 5 with values ranging from 0.00 (included) to 2.20 (included) | | Set only once in the constructor. |
| Methods (including constructors) | | | | | | | | |
| Name | Return type | Static [YES/NO] | | Access modifier |  | List of arguments and their types | | Notes / comments |
| Order(...) | N/A | NO | | public |  | int menuID, double[] ingredients | | Parameterized constructor. Sets up order ID as well. |
| getOrderID() | int | NO | | public |  | N/A | | Getter for the orderID attribute |
| getMenuID() | int | NO | | public |  | N/A | | Getter for the menuID |
| getCost() | double | NO | | public |  | N/A | | Getter for the cost attribute |
| getPrice() | double | NO | | public |  | N/A | | Getter for the price attribute |
| getRecipe | double[] | NO | | public |  | int menuItemID | | Getter for the amount of ingredients used for the specific taco that is ordered. |
|  |  |  |  |  |  |  |  |  |

Class / object interactions design table (A class / object interacts (calls a method of) with class/object B):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class / Object A | calls method (it can be constructor to create an object) | of class / object B with arguments (if any) | to | Notes: |
| Order | calls method getRecipe(menuItemID) from the | Inventory inventory | get +- 10 % of the amount of ingredients used for the specific taco that is ordered | - |
| Order | class method getIngredientCosts() from the | Inventory inventory | get the cost of each ingredient for the tacos. | - |
|  |  |  |  |  |
|  |  |  |  |  |

Test case design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case name | Tested class | Tested method | Test input | Expected outcome |
| check the orderIDCount and orderID values ( have to be > 0) | Order | getOrderID() | Values >=0  Values < 0 | If the value is < 0 there is a problem. |
| check the value for menuID (has to be between 0 and 3) | Order | getMenuID() | Values between 0 and 3.  Values different from the abovementioned. | If the value is different from the boundaries, there is a problem and the order can’t be proceeded. |
| check cost and price | Order | getCost()  getPrice() | Values >= 0.  Values < 0. | If the value of these attributes is < 0. There is a problem and the program is not working correctly. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Class name  OrderList | | |  | | | |  | |
| Attributes | | | | | | | | |
| Name | type | Static [YES/NO] | | Access modifier | Initial value if any | Range of possible values | | Notes / comments |
| orders | Order[] | NO | | private | Empty array of size 200. | N/A | | New orders (Order type references). |
| menuItemsSold | int[] | NO | | private | Empty array of size 4 - all elements initialized to 0 | Each element value can be an integer between 0 and 200 (both included) | | Used to keep track of how many orders for a particular menuItem were sold. |
| totalOrdersPlaced | int | NO | | private | 0 | Between 0 and 200 (both included) | | Incremented every time a new order is placed. |
| totalCost | double | NO | | private | 0.0d | Positive. Lower bound: 0.0d. Upper bound depends on the ingredient costs and order details. | | Updated every time a new order is placed. |
| totalProfit | double | NO | | private | 0.0d | Positive. Lower bound: 0.0d. Upper bound depends on the ingredient costs and order details | | Updated every time a new order is placed. |
| totalSales | double | NO | | private | 0.0d | Positive. Lower bound: 0.0d. Upper bound depends on the ingredient costs and order details | | Updated every time a new order is placed. |
| shellsLeft | int | NO | | private | 200 | Positive. Upper bound 200. Lower bound 0. | | Updated every time a new order is placed. |
| Methods (including constructors) | | | | | | | | |
| Name | Return type | Static [YES/NO] | | Access modifier |  | List of arguments and their types | | Notes / comments |
| OrderList() | N/A | NO | | public |  | N/A | | Used to set up the orders array and initialize other attributes. |
| add() | void | NO | | public |  | Order newOrder | | Adds new Order object reference to the array; updates all relevant statistics. |
| getTotalordersPlaced() | int | NO | | public |  | N/A | | Getter for the totalOrdersPlaced attribute |
| getTotalCost() | double | NO | | public |  | N/A | | Getter for the totalCost attribute |
| getTotalProfit() | double | NO | | public |  | N/A | | Getter for the totalProfit attribute |
| getTotalSales() | double | NO | | public |  | N/A | | Getter for the totalSales attribute |
| getOrdersByMenuItem() | int | NO | | public |  | int menuItemID | | Access the menuItemsSold array element (based on menuItemID argument) and return the amount of orders for this menuItem |
|  |  |  |  |  |  |  |  |  |

Class / object interactions design table (A class / object interacts (calls a method of) with class/object B):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class / Object A | calls method (it can be constructor to create an object) | of class / object B with arguments (if any) | to | Notes: |
| OrderList | getCost | Order newOrder | calculate the total cost in the add() method of OrderList class | - |
| OrderList | getPrice | Order newOrder | calculate the total price in the add() method of OrderList class | - |
| OrderList | getMenuID | Order newOrder | increment the menuItemsSold array | - |
|  |  |  |  |  |

Test case design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case name | Tested class | Tested method | Test input | Expected outcome |
| check if the shellsLeft is more than or equal to 0 | OrderList | getShellsLeft() | values >= 0  values < 0 | If the output is < 0, there is a problem. |
| check if the totalOrdersPlaced is > 0 | OrderList | getTotalOrdersPlaced() | values >= 0  values < 0 | If the output is < =0, there is a problem. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Class name  TacoTruck | | |  | | | |  | |
| Attributes | | | | | | | | |
| Name | type | Static [YES/NO] | | Access modifier | Initial value if any | Range of possible values | | Notes / comments |
| name | String | NO | | private | Truck O’Bell | N/A | | Set only once in the constructor. |
| inventory | Inventory | NO | | private | N/A | N/A | | Inventory object; Set only once in the constructor. |
| orders | OrderList | NO | | private | N/A | N/A | | OrderList object. Set only once in the constructor. |
| menuItems | String[] | NO | | private | “Bolts and Hammers”  “Fuel Taco”  “Broken Tooth”  “Screw Taco” | N/A | | 1D array of Strings. Set only once in the constructor. |
| anOrder | Order | NO | | public | N/A | N/A | | Order object; |
|  |  |  | |  |  |  | |  |
| Methods (including constructors) | | | | | | | | |
| Name | Return type | Static [YES/NO] | | Access modifier |  | List of arguments and their types | | Notes / comments |
| TacoTruck() | N/A | NO | | public |  | N/A | | Instantiate objects, initialize arrays, etc. |
| getRecipe() | double[] | NO | | private |  | int menuItemID | | Return the recipe 1D array given the menu item ID |
| getMenuItemName() | String | NO | | private |  | int menuItemID | | Return a String name given the menu item ID |
| run() | void | NO | | public |  | N/A | | Displays menus, get input from user, start manual/simulated ordering, etc. |
| manual() | void | NO | | private |  | N/A | | Manual order creation |
| simulate() | void | NO | | private |  | N/A | | Simulates creating multiple random orders |
| showSalesSummary() | void | NO | | public |  | N/A | | Displays the sales summary for orders |
|  |  |  |  |  |  |  |  |  |

Class / object interactions design table (A class / object interacts (calls a method of) with class/object B):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class / Object A | calls method (it can be constructor to create an object) | of class / object B with arguments (if any) | to | Notes: |
| TacoTruck | getShellsLeft() | OrderList orders | run the simulation if the shells left are more than 0. | - |
| TacoTruck | getIngredientNames() | Inventory inventory | get the names of the tacos and their ingredients for the manual order. | - |
| TacoTruck | checkIfEnough(getRecipe()) | Inventory inventory | checks if there are enough ingredients to make a taco | - |
| TacoTruck | getOrderID() | Order anOrder | gets the order ID that will show the number of the order. | - |
| TacoTruck | getOrdersByMenuItem() | OrderList orders | gets the menu item that was placed in the order and increments the number that shows the amount of time a specific taco was ordered |  |
| TacoTruck | getTotalCost() | OrderList orders | gets the total cost for the tacos made | - |
| TacoTruck | getTotalSales() | OrderList orders | gets the total sales for the tacos made | - |
| TacoTruck | getTotalProfit() | OrderList orders | gets the total profit for the tacos made | - |

Test case design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case name | Tested class | Tested method | Test input | Expected outcome |
| N/A | N/A | N/A | N/A | N/A |