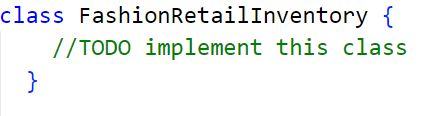
## Problem 2.Fashion Retail Management



In the dynamic world of fashion retail, efficiently managing inventory is pivotal for catering to the ever-evolving demands of customers who seek clothing, shoes, and accessories. To meet these demands, you need to design a **class** named "**FashionRetailInventory** with updated functionality tailored to this specialized environment.

**Functionality**

**Constructor**

Should have these **3** properties:

* **storehouse (string);**
* **location (string);**
* **productStock (empty array);**

**At the initialization** of the **FashionRetailInventory** **class**, the **constructor** accepts the **storehouse** and **location.**

### addProduct (productName, size, quantity, price)

This method adds a fashion product to the store's inventory. The method accepts four arguments:

* **productName (string);**
* **size (string);**
* **quantity (number);**
* **price (number);**
* If a product with the same name and size already exists in the **productStock**, add the quantity to the product and **return** the following message:

**`You added ${quantity} more pieces of product ${productName}** **size ${size}`**

* Otherwise, **add** the product with properties: **{productName, size, quantity, price}** to the **productStock** and **return** the following message:

**`The product ${productName}, size ${size} was successfully added to the inventory`**

### sendProduct (productName, size)

This method allows the sending of a product from the store's inventory. The method accepts two arguments:

* **productName (string)**;
* **size (string);**
* If the product with the given name and size is not present in the **productStock**, **throw** an **error** with the following message:

**`The product ${productName}, size ${size} is not in the inventory`**

* Otherwise, remove the product from the **productStock** and **return** the following message:

**`The product ${productName}, size ${size} was successfully removed from the inventory`**

### findProductsBySize(size)

The method accepts one arguments:

* **size (string);**
* Iterate through the **productStock** array and find all products that match the specified size and **return** a list of the products that match the given size in the format:

**`${product1}-${quantity} pieces, ${product2}-${quantity} pieces,…`**

* If no products matching the specified size are found in the **productStock**, **return:**

**`There are no products available in that size`**

### listProducts ()

* If there are no products in stock, **return**:

**`${storehouse} storehouse is empty`**

* Otherwise, on the first line **return**:

**`${storehouse} storehouse in ${location} available products:`**

* On the next lines, display information about each **product, sorted** in **alphabetical** order by their **productName** in the following format:

**`${productName}/Size:${size}/Quantity:${quantity}/Price:${price}$`**

### Examples

|  |
| --- |
| **Input 1** |
| const storeHouse = new FashionRetailInventory("East", "Milano");    console.log(storeHouse.addProduct("Shirt", "M", 10, 25.0));    console.log(storeHouse.addProduct("T-Shirt", "M", 10, 25.0));    console.log(storeHouse.addProduct("Sweather", "M", 10, 25.0));    console.log(storeHouse.addProduct("Sweather", "M", 10, 25.0)); |

|  |
| --- |
| **Output 1** |
| The product Shirt, size M was successfully added to the inventory  The product T-Shirt, size M was successfully added to the inventory  The product Sweather, size M was successfully added to the inventory  You added 10 more pieces of product Sweather size M |

|  |
| --- |
| **Input 2** |
| const storeHouse = new FashionRetailInventory("East", "Milano");    console.log(storeHouse.addProduct("Shirt", "M", 10, 25.0));    console.log(storeHouse.addProduct("T-Shirt", "M", 10, 25.0));    console.log(storeHouse.sendProduct("T-Shirt", "M"));    console.log(storeHouse.sendProduct("Sweather", "M")); |

|  |
| --- |
| **Output 2** |
| The product Shirt, size M was successfully added to the inventory  The product T-Shirt, size M was successfully added to the inventory  The product T-Shirt, size M was successfully removed from the inventory  Uncaught Error Error: The product Sweather, size M is not in the inventory |

|  |
| --- |
| **Input 3** |
| const storeHouse = new FashionRetailInventory("East", "Milano");    console.log(storeHouse.addProduct("Shirt", "M", 10, 25.0));    console.log(storeHouse.addProduct("T-Shirt", "M", 10, 25.0));    console.log(storeHouse.findProductsBySize("M"));    console.log(storeHouse.findProductsBySize("XL")); |

|  |
| --- |
| **Output 3** |
| The product Shirt, size M was successfully added to the inventory  The product T-Shirt, size M was successfully added to the inventory  Shirt-10 pieces, T-Shirt-10 pieces  There are no products available in that size |

|  |
| --- |
| **Input 4** |
| const storeHouse = new FashionRetailInventory("East", "Milano");    console.log(storeHouse.addProduct("Shirt", "M", 10, 25.0));    console.log(storeHouse.addProduct("T-Shirt", "M", 10, 25.0));    console.log(storeHouse.addProduct("Shirt", "L", 5, 30.0));    console.log(storeHouse.addProduct("Shoes", "9", 8, 50.0));    console.log(storeHouse.sendProduct("Shoes", "9", 8, 50.0));    console.log(storeHouse.listProducts()); |

|  |
| --- |
| **Output 4** |
| The product Shirt, size M was successfully added to the inventory  The product T-Shirt, size M was successfully added to the inventory  The product Shirt, size L was successfully added to the inventory  The product Shoes, size 9 was successfully added to the inventory  The product Shoes, size 9 was successfully removed from the inventory  East storehouse in Milano available products:  Shirt/Size:M/Quantity:10/Price:25$  Shirt/Size:L/Quantity:5/Price:30$  T-Shirt/Size:M/Quantity:10/Price:25$ |