# KONGARA SAI 192365025

PROJECT 7 (PART-1)

# Step 1: Modify the main method

* Create a static method called displayInventory that takes the products array as a parameter.

# Move the code that displays the array from the main method to the displayInventory method.

* Replace the display code in the main method with a method call to displayInventory.

# Step 2: Create a static method to add to inventory

* Create a static method called addTolnventory that takes the products array and a Scanner object as parameters.

# Move the code that adds values to the array from the main method to the addTolnventory method.

* Move the local variables required

(e.g. tempNumber, tempName, tempQty, tempPrice) from the main method to the top of the addTolnventory method.

# Add a method call in main to the addTolnventory method. Step 3: Create a method to get the number of products

* Create a method called getNumProducts that takes a Scanner object as a parameter and returns an integer value.

# Move the code that gets the maximum number of products from the user into this method.

* Add a method call in main to the getNumProducts method and store the returned value in the maxsize variable.

# Step 4: Create methods in the Product class

* Create two new methods in

the Product class: addTolnventory and deductFromlnventory.

# The addTolnventory method should accept a parameter (quantity) that holds the number of items to add.

* The deductFromlnventory method should accept a parameter (quantity) that holds the number of items to deduct.

# Step 5: Modify the ProductTester class

* Create a method called getMenuOption that takes a Scanner object as a parameter and returns an integer value.

# Create a menu system that displays options and returns the menu choice entered by the user.

* Create a method called getProductNumber that takes the products array and a Scanner object as parameters and returns an integer value.
* Create methods called addlnventory and deductlnventory that take the products array and a Scanner object as parameters.

*CODE:*

public class Product {

private String name; private int quantity; private double price; private int itemNumber;

public Product(String name, int quantity, double price, int itemNumber) { this.name = name;

this.quantity = quantity; this.price = price; this.itemNumber = itemNumber;

}

public String getName() { return name;

}

public int getQuantity() { return quantity;

}

public double getPrice() { return price;

}

public int getItemNumber() { return itemNumber;

}

public void addTolnventory(int quantity) { this.quantity += quantity;

}

public void deductFromlnventory(int quantity) { if (quantity <= this.quantity) {

this.quantity -= quantity;

} else {

System.out.println("Not enough stock to deduct.");

}

}

}

import java.util.InputMismatchException; import java.util.Scanner;

public class ProductTester {

public static void main(String[] args) { Scanner scanner = new Scanner(System.in);

int maxsize = getNumProducts(scanner); Product[] products = new Product[maxsize];

addTolnventory(products, scanner);

int menuOption = getMenuOption(scanner); while (menuOption != 0) {

switch (menuOption) { case 1:

displaylnventory(products); break;

case 2:

addlnventory(products, scanner); break;

case 3:

deductlnventory(products, scanner); break;

case 4:

discontinueProduct(products, scanner); break;

default:

System.out.println("Invalid option. Please try again.");

}

menuOption = getMenuOption(scanner);

}

}

public static void displaylnventory(Product[] products) { for (int i = 0; i < products.length; i++) {

System.out.println("Index: " + i + ", Name: " + products[i].getName());

}

}

public static void addTolnventory(Product[] products, Scanner scanner) { for (int i = 0; i < products.length; i++) {

System.out.println("Enter product name:"); String name = scanner.next(); System.out.println("Enter product quantity:"); int quantity = scanner.nextInt(); System.out.println("Enter product price:"); double price = scanner.nextDouble();

System.out.println("Enter product item number:"); int itemNumber = scanner.nextInt();

products[i] = new Product(name, quantity, price, itemNumber);

}

}

public static int getNumProducts(Scanner scanner) { int maxsize;

while (true) { try {

System.out.println("Enter the number of products you would like to add:"); maxsize = scanner.nextInt();

if (maxsize <= 0) {

System.out.println("Invalid input. Please enter a positive number.");

} else {

break;

}

} catch (InputMismatchException e) { System.out.println("Invalid input. Please enter a number."); scanner.next();

}

}

return maxsize;

}

public static int getMenuOption(Scanner scanner) { int menuOption;

while (true) { try {

System.out.println("1. View Inventory"); System.out.println("2. Add Stock"); System.out.println("3. Deduct Stock"); System.out.println("4. Discontinue Product"); System.out.println("0. Exit"); System.out.println("Please enter a menu option:"); menuOption = scanner.nextInt();

if (menuOption < 0 || menuOption > 4) { System.out.println("Invalid option. Please try again.");

} else {

break;

}

} catch (InputMismatchException e) { System.out.println("Invalid input. Please enter a number."); scanner.next();

}

}

return menuOption;

}

public static int getProductNumber(Product[] products, Scanner scanner) { int productChoice;

while (true) { try {

for (int i = 0; i < products.length; i++) {

System.out.println("Index: " + i + ", Name: " + products[i].getName());

}

System.out.println("Please enter the index of the product:"); productChoice = scanner.nextInt();

if (productChoice < 0 || productChoice >= products.length) { System.out.println("Invalid input. Please enter a valid index.");

} else {

break;

}

} catch (InputMismatchException e) { System.out.println("Invalid input. Please enter a number."); scanner.next();

}

}

return productChoice;

}





