DATA STRUCTURE AND ALGORITHMS

CSC-215

PROJECT: INVENTORY MANAGEMENT SYSTEM

Semester:	**Spring 2024**
Program:	**BS-CS**
Faculty:	**Noor-ul-Huda**
Student Name:	**Aakanksha Pardeep Kumar**
Student ID:	**20221-32743**

Project 2: Inventory Management System (Java)

This project aims to develop an inventory management system using your choice of arrays or linked lists. The system should allow you to manage product information and perform various operations to maintain accurate stock levels.

- **Data Structure:**
- **Products:** You can choose to represent products using either:
- **Array:** Create an array of Product objects. Each Product object would contain attributes like product ID (unique identifier), name, price, quantity, etc.
- **Linked List:** Implement a linked list where each node holds a Product object with the same attributes.

- **Functionalities:**
  - **Add Product:** Implement a method to add a new product to the inventory. This might involve:
  a) Prompting the user for product details (name, price, quantity).
  b) Creating a new Product object with the provided information.
  c) Adding the object to the chosen data structure (array or linked list).

  - **Remove Product:** Develop a method to remove an existing product from the inventory. You might need to:
  a) Allow the user to search for a product by ID or name.
  b) Locate the product in the data structure (array or linked list).
  c) Remove the product object from the data structure.
  - **Search Product:** Implement a method to search for a product based on various criteria:
  a) Search by product ID.
  b) Search by product name (potentially using a partial match).

  - **Sort Products:** Design methods to sort the inventory based on different criteria:
  a) Sort by product name (alphabetical order).
  b) Sort by product price (ascending or descending).
  c) Sort by product quantity (low to high or high to low).

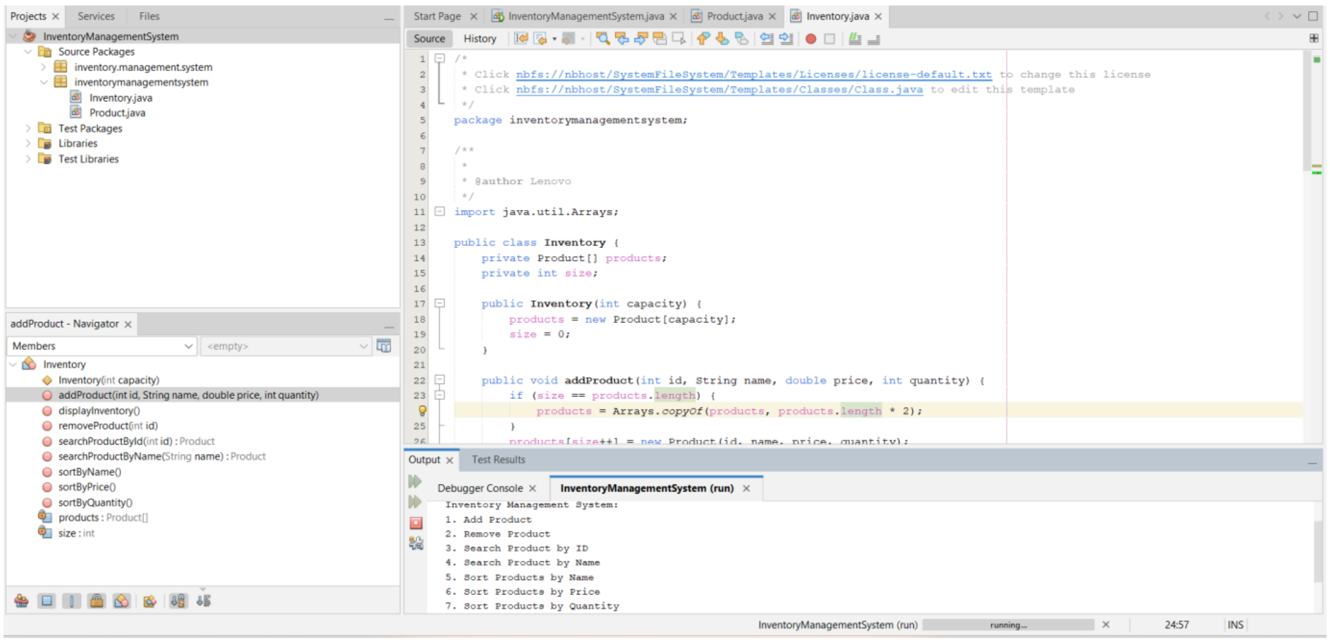
MAIN CLASS

Inventory Management System:

```
package inventory.management.system;

import inventorymanagementsystem.Inventory;

import inventorymanagementsystem.Product;

import java.util.InputMismatchException;

import java.util.Scanner;


public class InventoryManagementSystem {

  public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    Inventory inventory = new Inventory(10);


    while (true) {

      System.out.println("\nInventory Management System:");

      System.out.println("1. Add Product");

      System.out.println("2. Remove Product");

      System.out.println("3. Search Product by ID");

      System.out.println("4. Search Product by Name");

      System.out.println("5. Sort Products by Name");

      System.out.println("6. Sort Products by Price");

      System.out.println("7. Sort Products by Quantity");

      System.out.println("8. Display Inventory");

      System.out.println("9. Exit");

      System.out.print("Enter your choice: ");


      int choice = 0;

      try {

        choice = scanner.nextInt();

      } catch (InputMismatchException e) {

        System.out.println("Invalid input. Please enter a number.");
        scanner.next(); // Clear the invalid input
        continue;
    }
    scanner.nextLine(); // Consume newline

    switch (choice) {
        case 1:
            try {
                System.out.print("Enter Product ID: ");
                int id = scanner.nextInt();
                scanner.nextLine(); // Consume newline
                System.out.print("Enter Product Name: ");
                String name = scanner.nextLine();
                System.out.print("Enter Product Price: ");
                double price = scanner.nextDouble();
                System.out.print("Enter Product Quantity: ");
                int quantity = scanner.nextInt();
                inventory.addProduct(id, name, price, quantity);
            } catch (InputMismatchException e) {
                System.out.println("Invalid input. Please enter the correct data types.");
                scanner.next(); // Clear the invalid input
            }
            break;
        case 2:
            try {
                System.out.print("Enter Product ID to remove: ");
                int id = scanner.nextInt();
                inventory.removeProduct(id);
            } catch (InputMismatchException e) {
                System.out.println("Invalid input. Please enter a numeric Product ID.");
                scanner.next(); // Clear the invalid input
            }
            break;
        case 3:
            try {
                System.out.print("Enter Product ID to search: ");
                int id = scanner.nextInt();
                Product productById = inventory.searchProductById(id);
                System.out.println(productById != null ? productById : "Product not found.");
            } catch (InputMismatchException e) {
                System.out.println("Invalid input. Please enter a numeric Product ID.");
                scanner.next(); // Clear the invalid input
            }
            break;
        case 4:
            System.out.print("Enter Product Name to search: ");
            String name = scanner.nextLine();
            Product productByName = inventory.searchProductByName(name);
            System.out.println(productByName != null ? productByName : "Product not found.");
            break;
        case 5:
            inventory.sortByName();
            System.out.println("Products sorted by name.");
            break;
        case 6:
            inventory.sortByPrice();
            System.out.println("Products sorted by price.");
            break;
        case 7:
            inventory.sortByQuantity();
            System.out.println("Products sorted by quantity.");
            break;
```

```
                case 8:
                    inventory.displayInventory();
                    break;
                case 9:
                    scanner.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice.");
            }
        }
    }
}
```

CODE SNIPPET:



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package inventorymanagementsystem;

/**
 *
 * @author Lenovo
 */
import java.util.Arrays;

public class Inventory {
    private Product[] products;
    private int size;

    public Inventory(int capacity) {
        products = new Product[capacity];
        size = 0;
    }

    public void addProduct(int id, String name, double price, int quantity) {
        if (size == products.length) {
            products = Arrays.copyOf(products, products.length * 2);
        }
        products[size++] = new Product(id, name, price, quantity);
```

```
Inventory Management System:
1. Add Product
2. Remove Product
3. Search Product by ID
4. Search Product by Name
5. Sort Products by Name
6. Sort Products by Price
7. Sort Products by Quantity
```

CODE EXPLANATION:

Methods:

1. main` Method:

- This is the entry point of the program. It's where the execution starts.
- It contains a loop that keeps running indefinitely until the user chooses to exit.
- Inside the loop, it displays a menu of options and performs actions based on user input.

2. addProduct Method:

- This method adds a new product to the inventory.
- It asks the user to input details like ID, name, price, and quantity for the new product.
- Once the input is received, it creates a new product object and adds it to the inventory.

3. removeProduct` Method:

- This method removes a product from the inventory based on its ID.
- It prompts the user to enter the ID of the product they want to remove.
- If the product with the specified ID exists, it removes it from the inventory.

4. searchProductById Method:

- This method searches for a product in the inventory based on its ID.
- It asks the user to enter the ID of the product they want to search for.
- If the product with the specified ID is found, it returns that product; otherwise, it returns null.

5. searchProductByName Method:

- This method searches for a product in the inventory based on its name.
- It prompts the user to enter the name of the product they want to search for.
- If a product with a matching name is found (case-insensitive), it returns that product; otherwise, it returns null.

6. sortByName, sortByPrice, sortByQuantity Methods:

- These methods sort the products in the inventory based on their name, price, and quantity, respectively.
- They rearrange the products in the inventory array according to the specified sorting criteria.

7. displayInventory Method:

- This method displays the current inventory of products.

- It prints out details of each product in the inventory, including ID, name, price, and quantity.

Attributes:

1. Scanner scanner

- This is an object of the `Scanner` class used to read user input from the console.

2. Inventory inventory:

- This is an object of the `Inventory` class, representing the inventory of products.

- It holds an array of products and provides methods to add, remove, search, sort, and display products.

Other Code Parts:

- The code also includes error handling using try-catch blocks to handle input mismatch exceptions.

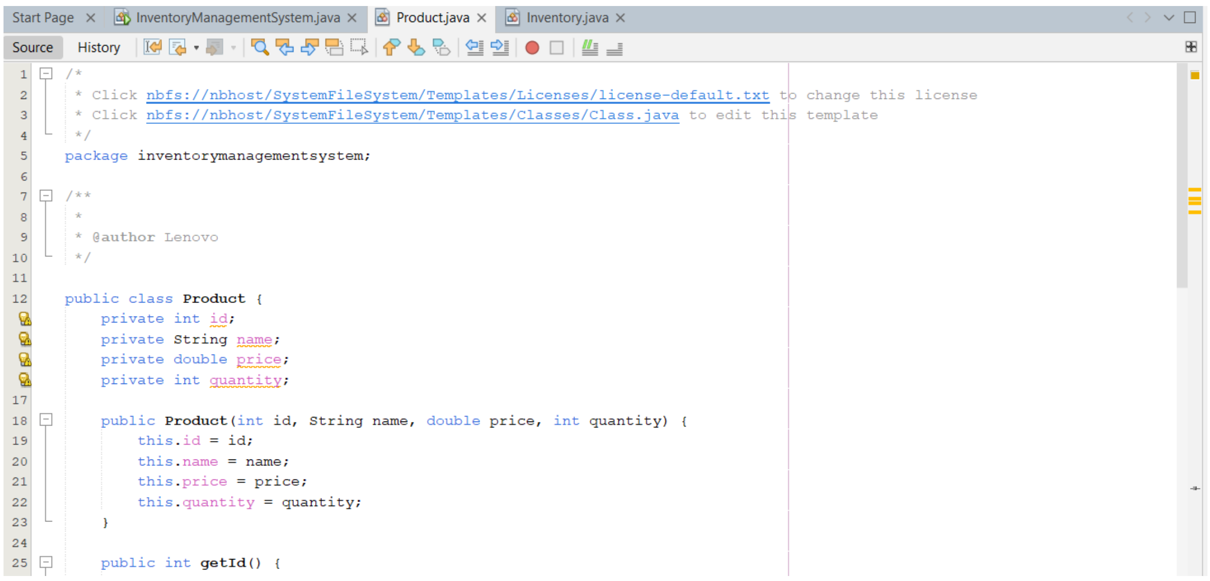
- It utilizes a switch statement to execute different functionalities based on user choices.

Product class:

```
package inventorymanagementsystem;

/**

 *

 * @author Lenovo

 */


public class Product {

   private int id;

   private String name;

   private double price;

   private int quantity;


   public Product(int id, String name, double price, int quantity) {

      this.id = id;

      this.name = name;

      this.price = price;

      this.quantity = quantity;

   }


   public int getId() {

      return id;

   }


   public String getName() {

      return name;

   }


   public double getPrice() {

      return price;

   }
```

```
    public int getQuantity() {
        return quantity;
    }

    @Override
    public String toString() {
        return "Product{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", price=" + price +
                ", quantity=" + quantity +
                '}';
    }
}
```

CODE SNIPPET:



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package inventorymanagementsystem;

/**
 *
 * @author Lenovo
 */

public class Product {
    private int id;
    private String name;
    private double price;
    private int quantity;

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

    public int getId() {
```

CODE EXPLANATION:

Attributes

-id: Represents the unique identifier of the product.

-name: Represents the name of the product.

-price: Represents the price of the product.

- quantity: Represents the quantity of the product available in the inventory.

Constructor

- Initializes a `Product` object with the provided values for `id`, `name`, `price`, and `quantity`.

Getter Methods

getId(): Returns the ID of the product.

getName(): Returns the name of the product.

getPrice(): Returns the price of the product.

getQuantity(): Returns the quantity of the product.

-toString() Method

- Overrides the default `toString()` method to provide a string representation of the `Product` object.

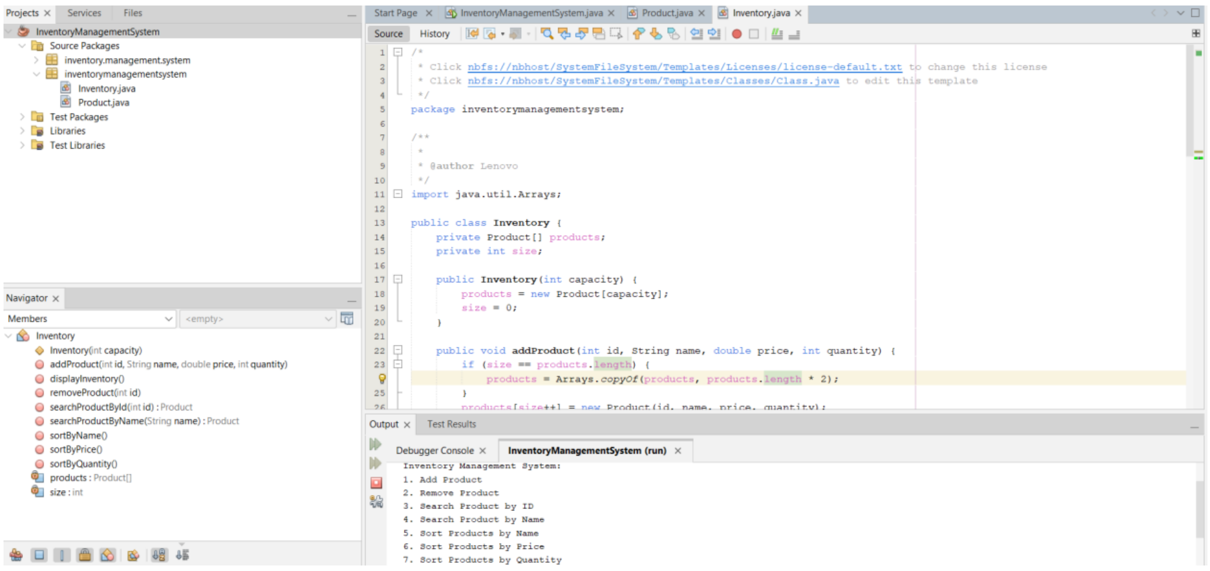
- Returns a string containing the ID, name, price, and quantity of the product, formatted for easy readability.

Inventory class:

```
package inventorymanagementsystem;

/**
 *
 * @author Lenovo
 */
import java.util.Arrays;

public class Inventory {
    private Product[] products;
    private int size;

    public Inventory(int capacity) {
        products = new Product[capacity];
        size = 0;
    }

    public void addProduct(int id, String name, double price, int quantity) {
        if (size == products.length) {
            products = Arrays.copyOf(products, products.length * 2);
        }
        products[size++] = new Product(id, name, price, quantity);
    }

    public void removeProduct(int id) {
        for (int i = 0; i < size; i++) {
            if (products[i].getId() == id) {
```

```
            products[i] = products[--size];

            products[size] = null;

            return;

        }

    }

}


public Product searchProductById(int id) {

    for (int i = 0; i < size; i++) {

        if (products[i].getId() == id) {

            return products[i];

        }

    }

    return null;

}


public Product searchProductByName(String name) {

    for (int i = 0; i < size; i++) {

        if (products[i].getName().equalsIgnoreCase(name)) {

            return products[i];

        }

    }

    return null;

}


public void sortByName() {

    Arrays.sort(products, 0, size, (p1, p2) -> p1.getName().compareToIgnoreCase(p2.getName()));

}


public void sortByPrice() {

    Arrays.sort(products, 0, size, (p1, p2) -> Double.compare(p1.getPrice(), p2.getPrice()));
    }

    public void sortByQuantity() {
        Arrays.sort(products, 0, size, (p1, p2) -> Integer.compare(p1.getQuantity(), p2.getQuantity()));
    }

    public void displayInventory() {
        for (int i = 0; i < size; i++) {
            System.out.println(products[i]);
        }
    }
}
```

CODE SNIPPET:



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package inventorymanagementsystem;

/**
 *
 * @author Lenovo
 */
import java.util.Arrays;

public class Inventory {
    private Product[] products;
    private int size;

    public Inventory(int capacity) {
        products = new Product[capacity];
        size = 0;
    }

    public void addProduct(int id, String name, double price, int quantity) {
        if (size == products.length) {
            products = Arrays.copyOf(products, products.length * 2);
        }
        products[size++] = new Product(id, name, price, quantity);
```

```
Inventory Management System:
1. Add Product
2. Remove Product
3. Search Product by ID
4. Search Product by Name
5. Sort Products by Name
6. Sort Products by Price
7. Sort Products by Quantity
```

CODE EXPLANATION:

Methods:

Constructor: Initializes the product array with a specified capacity and sets the initial size to zero.

addProduct(int id, String name, double price, int quantity): Adds a new product to the inventory. If the array is full, it dynamically resizes the array.

removeProduct(int id): Removes a product from the inventory based on the given ID. If the product is found, it is removed, and the array is adjusted accordingly.

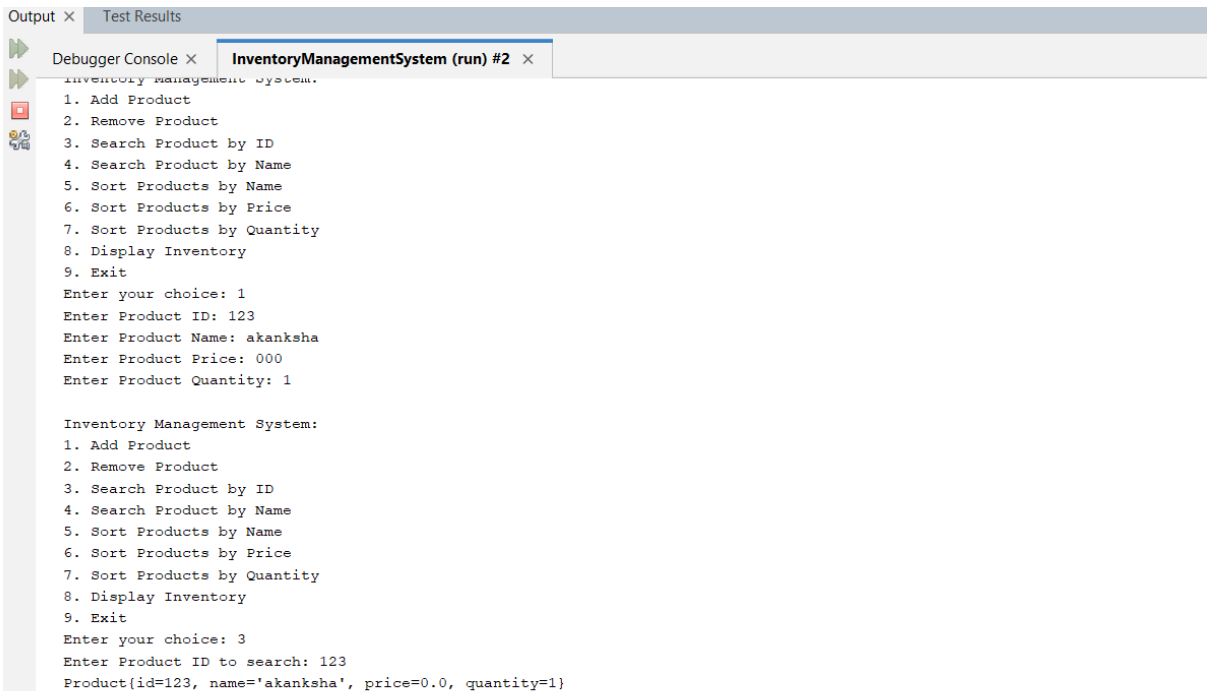
searchProductById(int id): Searches for a product by its ID and returns the product if found.

searchProductByName(String name): Searches for a product by its name and returns the product if found.

sortByName(), sortByPrice(), sortByQuantity(): Sorts the products in the inventory based on their name, price, and quantity, respectively.

displayInventory(): Displays all the products in the inventory.

WHOLE PROGRAM OUTPUT:



```
Inventory Management System:
1. Add Product
2. Remove Product
3. Search Product by ID
4. Search Product by Name
5. Sort Products by Name
6. Sort Products by Price
7. Sort Products by Quantity
8. Display Inventory
9. Exit
Enter your choice: 1
Enter Product ID: 123
Enter Product Name: akanksha
Enter Product Price: 000
Enter Product Quantity: 1

Inventory Management System:
1. Add Product
2. Remove Product
3. Search Product by ID
4. Search Product by Name
5. Sort Products by Name
6. Sort Products by Price
7. Sort Products by Quantity
8. Display Inventory
9. Exit
Enter your choice: 3
Enter Product ID to search: 123
Product{id=123, name='akanksha', price=0.0, quantity=1}
```

SOURCE CODE:

```
package inventory.management.system;


import inventorymanagementsystem.Inventory;

import inventorymanagementsystem.Product;
import java.util.InputMismatchException;
import java.util.Scanner;

public class InventoryManagementSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Inventory inventory = new Inventory(10);

        while (true) {
            System.out.println("\nInventory Management System:");
            System.out.println("1. Add Product");
            System.out.println("2. Remove Product");
            System.out.println("3. Search Product by ID");
            System.out.println("4. Search Product by Name");
            System.out.println("5. Sort Products by Name");
            System.out.println("6. Sort Products by Price");
            System.out.println("7. Sort Products by Quantity");
            System.out.println("8. Display Inventory");
            System.out.println("9. Exit");
            System.out.print("Enter your choice: ");

            int choice = 0;
            try {
                choice = scanner.nextInt();
            } catch (InputMismatchException e) {
                System.out.println("Invalid input. Please enter a number.");
                scanner.next(); // Clear the invalid input
                continue;
            }
            scanner.nextLine(); // Consume newline
```

```
switch (choice) {
  case 1:
    try {
      System.out.print("Enter Product ID: ");
      int id = scanner.nextInt();
      scanner.nextLine(); // Consume newline
      System.out.print("Enter Product Name: ");
      String name = scanner.nextLine();
      System.out.print("Enter Product Price: ");
      double price = scanner.nextDouble();
      System.out.print("Enter Product Quantity: ");
      int quantity = scanner.nextInt();
      inventory.addProduct(id, name, price, quantity);
    } catch (InputMismatchException e) {
      System.out.println("Invalid input. Please enter the correct data types.");
      scanner.next(); // Clear the invalid input
    }
    break;
  case 2:
    try {
      System.out.print("Enter Product ID to remove: ");
      int id = scanner.nextInt();
      inventory.removeProduct(id);
    } catch (InputMismatchException e) {
      System.out.println("Invalid input. Please enter a numeric Product ID.");
      scanner.next(); // Clear the invalid input
    }
    break;
  case 3:
```

```
                try {
                    System.out.print("Enter Product ID to search: ");
                    int id = scanner.nextInt();
                    Product productById = inventory.searchProductById(id);
                    System.out.println(productById != null ? productById : "Product not found.");
                } catch (InputMismatchException e) {
                    System.out.println("Invalid input. Please enter a numeric Product ID.");
                    scanner.next(); // Clear the invalid input
                }
                break;
            case 4:
                System.out.print("Enter Product Name to search: ");
                String name = scanner.nextLine();
                Product productByName = inventory.searchProductByName(name);
                System.out.println(productByName != null ? productByName : "Product not
found.");
                break;
            case 5:
                inventory.sortByName();
                System.out.println("Products sorted by name.");
                break;
            case 6:
                inventory.sortByPrice();
                System.out.println("Products sorted by price.");
                break;
            case 7:
                inventory.sortByQuantity();
                System.out.println("Products sorted by quantity.");
                break;
            case 8:
                inventory.displayInventory();
```

```
                    break;
                case 9:
                    scanner.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice.");
            }
        }
    }
}
package inventorymanagementsystem;

/**
 *
 * @author Lenovo
 */

public class Product {
    private int id;
    private String name;
    private double price;
    private int quantity;

    public Product(int id, String name, double price, int quantity) {
        this.id = id;
        this.name = name;
        this.price = price;
        this.quantity = quantity;
    }
    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    public int getQuantity() {
        return quantity;
    }

    @Override
    public String toString() {
        return "Product{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", price=" + price +
                ", quantity=" + quantity +
                '}';
    }
}
package inventorymanagementsystem;

/**
 *
 * @author Lenovo
 */
import java.util.Arrays;

public class Inventory {
    private Product[] products;
    private int size;

    public Inventory(int capacity) {
        products = new Product[capacity];
        size = 0;
    }

    public void addProduct(int id, String name, double price, int quantity) {
        if (size == products.length) {
            products = Arrays.copyOf(products, products.length * 2);
        }
        products[size++] = new Product(id, name, price, quantity);
    }

    public void removeProduct(int id) {
        for (int i = 0; i < size; i++) {
            if (products[i].getId() == id) {
                products[i] = products[--size];
                products[size] = null;
                return;
            }
        }
    }
    public Product searchProductById(int id) {
        for (int i = 0; i < size; i++) {
            if (products[i].getId() == id) {
                return products[i];
            }
        }
        return null;
    }

    public Product searchProductByName(String name) {
        for (int i = 0; i < size; i++) {
            if (products[i].getName().equalsIgnoreCase(name)) {
                return products[i];
            }
        }
        return null;
    }

    public void sortByName() {
        Arrays.sort(products, 0, size, (p1, p2) ->
p1.getName().compareToIgnoreCase(p2.getName()));
    }

    public void sortByPrice() {
        Arrays.sort(products, 0, size, (p1, p2) -> Double.compare(p1.getPrice(), p2.getPrice()));
    }

    public void sortByQuantity() {
        Arrays.sort(products, 0, size, (p1, p2) -> Integer.compare(p1.getQuantity(),
p2.getQuantity()));
    }
    public void displayInventory() {
        for (int i = 0; i < size; i++) {
            System.out.println(products[i]);
        }
    }
}
```