WEEK – 1

Exercise 1: Inventory Management System

Nipuna Amanapu

[namanapu@gitam.in](mailto:namanapu@gitam.in)

superset id : 6432842

**Exercise 1: Inventory Management System**

**Scenario:**

You are developing an inventory management system for a warehouse. Efficient data storage and retrieval are crucial.

**Steps:**

1. **Understand the Problem:**
   * Explain why data structures and algorithms are essential in handling large inventories.
   * Discuss the types of data structures suitable for this problem.
2. **Setup:**
   * Create a new project for the inventory management system.
3. **Implementation:**
   * Define a class Product with attributes like **productId**, **productName**, **quantity**, and **price**.
   * Choose an appropriate data structure to store the products (e.g., ArrayList, HashMap).
   * Implement methods to add, update, and delete products from the inventory.
4. **Analysis:**
   * Analyze the time complexity of each operation (add, update, delete) in your chosen data structure.
   * Discuss how you can optimize these operations.

Solution:

1. Product.java

public class Product {

private int productId;

private String productName;

private int quantity;

private double price;

public Product(int productId, String productName, int quantity, double price) {

this.productId = productId;

this.productName = productName;

this.quantity = quantity;

this.price = price;

}

public int getProductId() {

return productId;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

public void setPrice(double price) {

this.price = price;

}

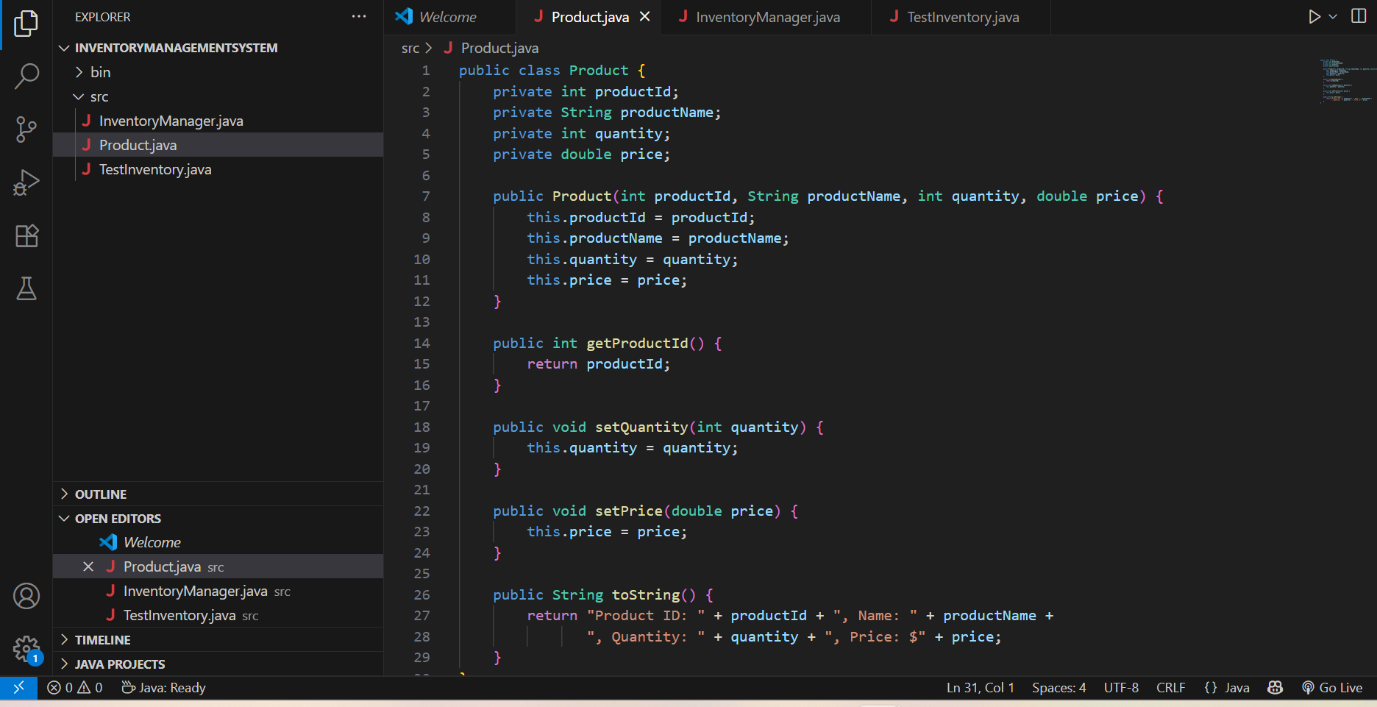
public String toString() {

return "Product ID: " + productId + ", Name: " + productName +

", Quantity: " + quantity + ", Price: $" + price;

}

}



1. InventoryManager.java

import java.util.HashMap;

public class InventoryManager {

private HashMap<Integer, Product> inventory = new HashMap<>();

public void addProduct(Product product) {

inventory.put(product.getProductId(), product);

System.out.println("Product added: " + product);

}

public void updateProduct(int productId, int newQuantity, double newPrice) {

Product product = inventory.get(productId);

if (product != null) {

product.setQuantity(newQuantity);

product.setPrice(newPrice);

System.out.println("Product updated: " + product);

} else {

System.out.println("Product not found.");

}

}

public void deleteProduct(int productId) {

Product removed = inventory.remove(productId);

if (removed != null) {

System.out.println("Product deleted: " + removed);

} else {

System.out.println("Product not found.");

}

}

public void displayAllProducts() {

if (inventory.isEmpty()) {

System.out.println("Inventory is empty.");

} else {

for (Product product : inventory.values()) {

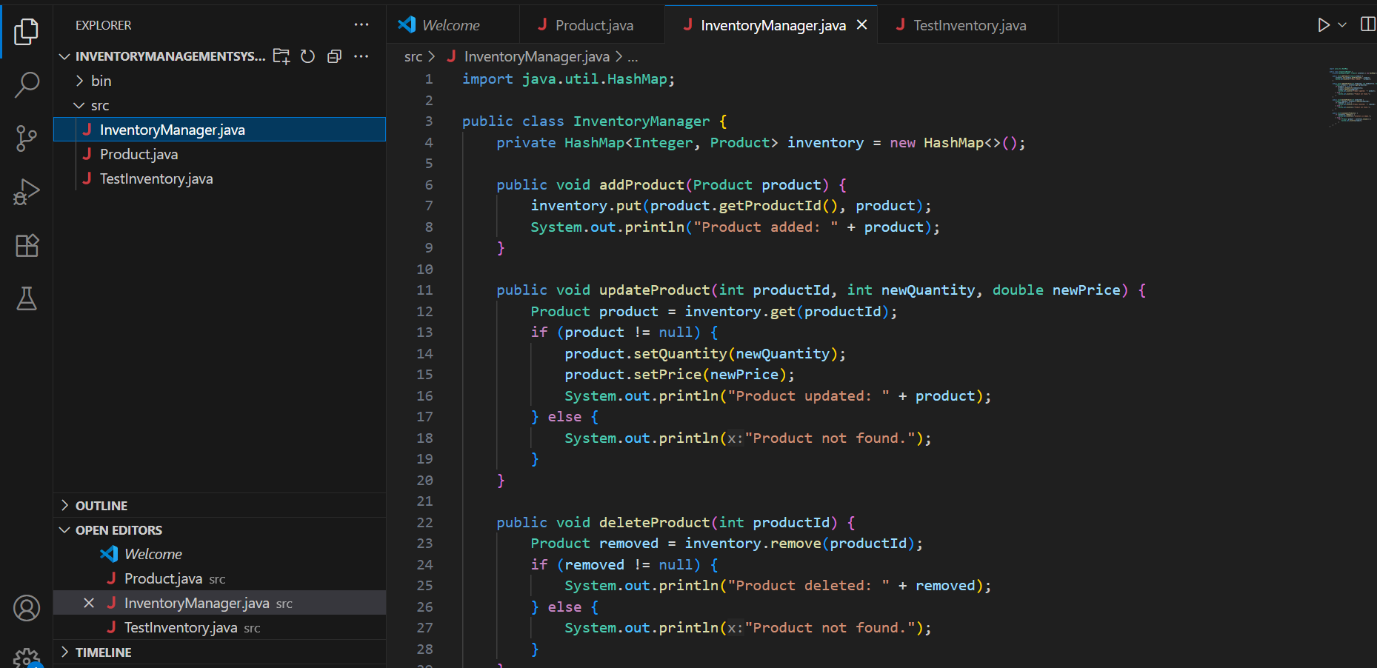
System.out.println(product);

}

}

}

}





1. TestInventory.java

public class TestInventory {

public static void main(String[] args) {

InventoryManager manager = new InventoryManager();

manager.addProduct(new Product(101, "Laptop", 10, 75000));

manager.addProduct(new Product(102, "Mouse", 50, 500));

manager.addProduct(new Product(103, "Keyboard", 30, 1200));

System.out.println("\nAll Products:");

manager.displayAllProducts();

System.out.println("\nUpdating product with ID 102...");

manager.updateProduct(102, 60, 550);

System.out.println("\nDeleting product with ID 101...");

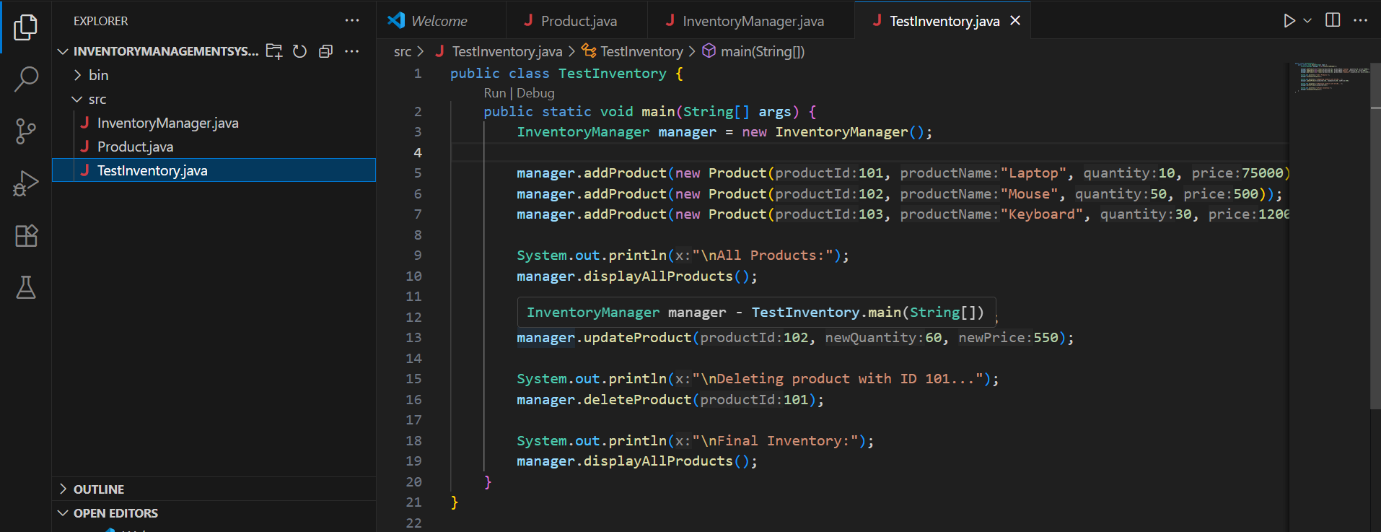
manager.deleteProduct(101);

System.out.println("\nFinal Inventory:");

manager.displayAllProducts();

}

}



Output:

