**Ex No: Wrapper, Exception Handling and Collections**

**Date:**

**AIM**

The aim of the provided Java programs is to implement the Wrapper, Exception Handling and Collections

**STEP 1: Product Management Program**

* Define a Product class with attributes ID, name, price, and quantity.
* Create an ArrayList<Product> to store products.
* Implement methods for adding a product, listing all products, searching for a product by ID, removing a product, and updating the price of a product.
* Use a loop to display a menu, take user input, and perform corresponding actions based on the user's choice.

**STEP 2: Autoboxing and Unboxing Program:**

* Define a class Main with a main method.
* Declare variables of primitive types (int, double).
* Demonstrate autoboxing by assigning primitive values to their corresponding wrapper classes.
* Demonstrate unboxing by assigning wrapper class values to primitive variables.
* Print the values before and after autoboxing/unboxing.

**STEP 3: Amazon Store Program:**

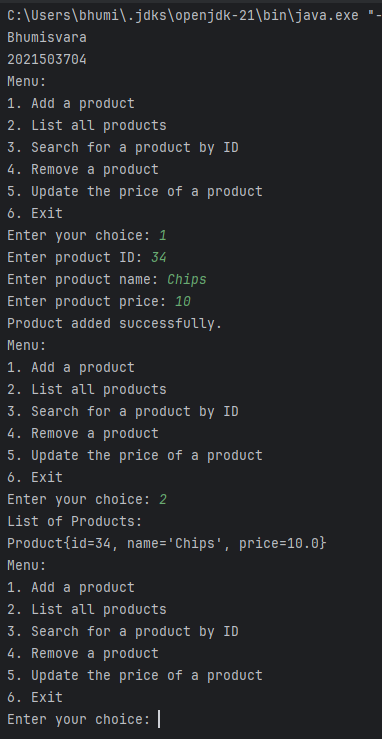
* Define a Product class with attributes name, price, and quantity.
* Implement custom exceptions for handling product-related errors (already exists, not found, insufficient quantity).
* Create an AmazonStore class with methods for adding products, adding products to the cart, listing products, and viewing the cart.
* Use a loop in the main method to display a menu for the user to interact with the AmazonStore application.
* Handle exceptions appropriately for user input errors.

1. **Write a Product class with the following attributes:id (int),name (String),price (double). Create an ArrayList to store Product objects. Ask the user to add products to the list. For each product, prompt the user for the id, name, and price and create a Product object.Add the Product object to the ArrayList.After adding products, display a menu to the user with the following options:List all products,Search for a product by ID, remove the product and update the price of the product.**

**CODE**

port java.util.ArrayList;  
import java.util.Scanner;  
class Product {  
 private int id;  
 private String name;  
 private double price;  
 public Product(int id, String name, double price) {  
 this.id = id;  
 this.name = name;  
 this.price = price;  
 }  
 public int getId() {  
 return id;  
 }  
public String getName() {  
 return name;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
 public void setPrice(double price) {  
 this.price = price;  
 }  
 @Override  
 public String toString() {  
 return "Product{" +  
 "id=" + id +  
 ", name='" + name + '\'' +  
 ", price=" + price +  
 '}';  
 }  
}  
public class Main {  
 private static ArrayList<Product> *products* = new ArrayList<>();  
 private static Scanner *scanner* = new Scanner(System.*in*);  
 public static void main(String[] args) {  
 boolean exit = false;  
  
 System.*out*.println("Bhumisvara ");  
 System.*out*.println("2021503704 ");  
  
 while (!exit) {  
 System.*out*.println("Menu:");  
 System.*out*.println("1. Add a product");  
 System.*out*.println("2. List all products");  
 System.*out*.println("3. Search for a product by ID");  
 System.*out*.println("4. Remove a product");  
 System.*out*.println("5. Update the price of a product");  
 System.*out*.println("6. Exit");  
 System.*out*.print("Enter your choice: ");  
 int choice = *scanner*.nextInt();  
 *scanner*.nextLine(); // Consume newline  
 switch (choice) {  
 case 1:  
 *addProduct*();  
 break;  
 case 2:  
 *listProducts*();  
 break;  
 case 3:  
 *searchProductById*();  
 break;  
 case 4:  
 *removeProduct*();  
 break;  
 case 5:  
 *updateProductPrice*();  
 break;  
 case 6:  
 exit = true;  
 break;  
 default:  
 System.*out*.println("Invalid choice. Please try again.");  
 }  
 }  
 }  
 private static void addProduct() {  
 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();  
 *scanner*.nextLine(); // Consume newline  
 Product product = new Product(id, name, price);  
 *products*.add(product);  
 System.*out*.println("Product added successfully.");  
 }  
 private static void listProducts() {  
 System.*out*.println("List of Products:");  
 for (Product product : *products*) {  
 System.*out*.println(product);  
 }  
 }  
 private static void searchProductById() {  
 System.*out*.print("Enter product ID to search: ");  
 int searchId = *scanner*.nextInt();  
 *scanner*.nextLine(); // Consume newline  
 for (Product product : *products*) {  
 if (product.getId() == searchId) {  
 System.*out*.println("Product found: " + product);  
 return;  
 }  
 }  
  
 System.*out*.println("Product not found with ID: " + searchId);  
 }  
 private static void removeProduct() {  
 System.*out*.print("Enter product ID to remove: ");  
 int removeId = *scanner*.nextInt();  
 *scanner*.nextLine(); // Consume newline  
 boolean removed = false;  
 for (Product product : *products*) {  
 if (product.getId() == removeId) {  
 *products*.remove(product);  
 System.*out*.println("Product removed: " + product);  
 removed = true;  
 break;  
 }  
 }  
 if (!removed)  
 System.*out*.println("Product not found with ID: " + removeId);  
 }  
 private static void updateProductPrice() {  
 System.*out*.print("Enter product ID to update price: ");  
 int updateId = *scanner*.nextInt();  
 *scanner*.nextLine(); // Consume newline  
 boolean updated = false;  
 for (Product product : *products*) {  
 if (product.getId() == updateId) {  
 System.*out*.print("Enter new price for the product: ");  
 double newPrice = *scanner*.nextDouble();  
 *scanner*.nextLine(); // Consume newline  
 product.setPrice(newPrice);  
 System.*out*.println("Price updated for product ID " + updateId);  
 updated = true;  
 break;  
 }  
 }  
  
 if (!updated)  
 System.*out*.println("Product not found with ID: " + updateId);  
 }  
}

**OUTPUT**



1. **Write a BoxAndUnbox class. Write a method to perform Autoboxing and unboxing of all primitive types in java [CO2: BL:3] points: 25**

**int a=10;**

**Integer intObj=new Integer(a);**

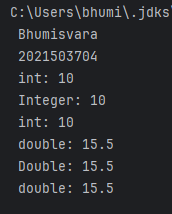
**int aUnbox=intObj.intValue();**

**//doubleValue() for Double and similarly for other primitive types**

CODE

public class 3704 {  
 public static void main(String[] args) {  
 int a = 10;  
  
 // Autoboxing Converting int to object  
 Integer intObj = a;  
  
 // Unboxing Converting object to int  
 int aUnbox = intObj;  
 System.*out*.println(" Bhumisvara");  
 System.*out*.println(" 2021503704");  
 System.*out*.println(" int: " + a);  
 System.*out*.println(" Integer: " + intObj);  
 System.*out*.println(" int: " + aUnbox);  
  
 double b = 15.5;  
  
  
 Double doubleObj = b;  
  
  
 double bUnbox = doubleObj;  
  
 System.*out*.println(" double: " + b);  
 System.*out*.println(" Double: " + doubleObj);  
 System.*out*.println(" double: " + bUnbox);  
  
  
 }  
}

**OUTPUT**



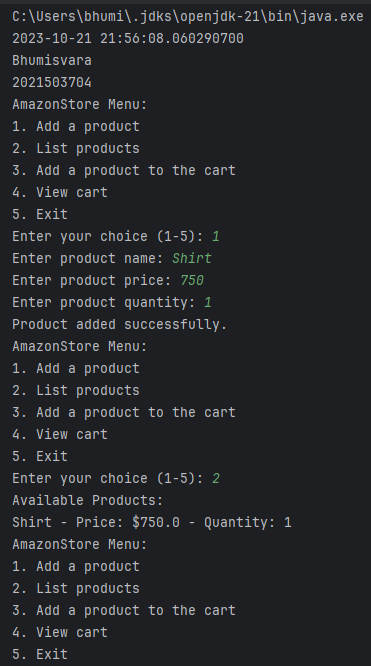
1. **Write a Java program for an Amazonstore to handle the customer orders. The application has Product, Amazonstore, and Main classes with specific method's purpose and functionality:**

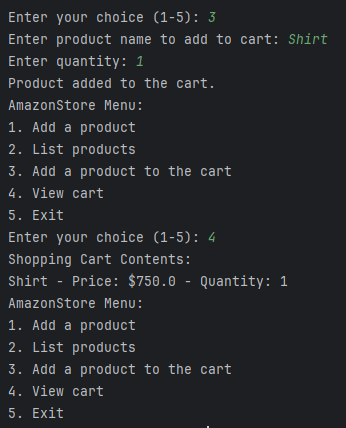
**CODE**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.\*;  
  
class Product {  
 private String name;  
 private double price;  
 public int quantity;  
  
 public Product(String name, double price, int quantity) {  
 this.name = name;  
 this.price = price;  
 this.quantity = quantity;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public int getQuantity() {  
 return quantity;  
 }  
}  
class ProductAlreadyExistsException extends Exception {  
 public ProductAlreadyExistsException(String message) {  
 super(message);  
 }  
}  
class ProductNotFoundException extends Exception {  
 public ProductNotFoundException(String message) {  
 super(message);  
 }  
}  
  
class InsufficientQuantityException extends Exception {  
 public InsufficientQuantityException(String message) {  
 super(message);  
 }  
}  
class AmazonStore {  
 private Map<String, Product> productMap;  
 private Set<String> productNames;  
 private List<Product> cart;  
 public AmazonStore() {  
 productMap = new HashMap<>();  
 productNames = new HashSet<>();  
 cart = new ArrayList<>();  
 }  
  
 public void addProduct(String name, double price, int quantity) throws ProductAlreadyExistsException {  
 if (productNames.contains(name)) {  
 throw new ProductAlreadyExistsException("Product already exists in the store.");  
 }  
 Product product = new Product(name, price, quantity);  
 productMap.put(name, product);  
 productNames.add(name);  
 }  
 public void addToCart(String name, int quantity) throws ProductNotFoundException, InsufficientQuantityException {  
 Product product = productMap.get(name);  
 if (product == null) {  
 throw new ProductNotFoundException("Product not found in the store.");  
 }  
  
 if (product.getQuantity() < quantity) {  
 throw new InsufficientQuantityException("Insufficient quantity in stock.");  
 }  
 productMap.get(name).quantity -= quantity;  
 cart.add(new Product(name, product.getPrice(), quantity));  
 }  
 public void listProducts() {  
 System.*out*.println("Available Products:");  
 for (Product product : productMap.values()) {  
 System.*out*.println(product.getName() + " - Price: $" + product.getPrice() + " - Quantity: " + product.getQuantity());  
 }  
 }  
 public void viewCart() {  
 System.*out*.println("Shopping Cart Contents:");  
 for (Product product : cart) {  
 System.*out*.println(product.getName() + " - Price: $" + product.getPrice() + " - Quantity: " + product.getQuantity());  
 }  
 }  
}  
public class b3704{  
 public static void main(String[] args) {  
 LocalTime myobj = LocalTime.*now*();  
 LocalDate obj = LocalDate.*now*();  
 System.*out*.println(obj+" "+myobj);  
 System.*out*.println("Bhumisvara");

System.*out*.println("2021503704");  
 AmazonStore store = new AmazonStore();  
 Scanner scanner = new Scanner(System.*in*);  
 while (true) {  
 System.*out*.println("AmazonStore Menu:");  
 System.*out*.println("1. Add a product");  
 System.*out*.println("2. List products");  
 System.*out*.println("3. Add a product to the cart");  
 System.*out*.println("4. View cart");  
 System.*out*.println("5. Exit");  
 System.*out*.print("Enter your choice (1-5): ");  
  
 int choice = scanner.nextInt();  
 scanner.nextLine();  
  
 try {  
 switch (choice) {  
 case 1:  
 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();  
 scanner.nextLine();  
 store.addProduct(name, price, quantity);  
 System.*out*.println("Product added successfully.");  
 break;  
 case 2:  
 store.listProducts();  
 break;  
 case 3:  
 System.*out*.print("Enter product name to add to cart: ");  
 String cartItemName = scanner.nextLine();  
 System.*out*.print("Enter quantity: ");  
 int cartItemQuantity = scanner.nextInt();  
 scanner.nextLine();  
 store.addToCart(cartItemName, cartItemQuantity);  
 System.*out*.println("Product added to the cart.");  
 break;  
 case 4:  
 store.viewCart();  
 break;  
 case 5:  
 System.*out*.println("Exiting AmazonStore. Goodbye!");  
 scanner.close();  
 System.*exit*(0);  
  
 default:  
 System.*out*.println("Invalid choice. Please enter a valid option (1-5).");  
 }  
 } catch (ProductAlreadyExistsException | ProductNotFoundException | InsufficientQuantityException e) {  
 System.*err*.println("Error: " + e.getMessage());  
 }  
 }  
 }  
}

**OUTPUT**





**RESULT**

Thus, The Wrapper, Exception Handling and Collections program has been successfully implemented.