**SHOPPING CART Application MINIPROJECT**

CODE-

package shoppingCartApplication;

import java.util.ArrayList;

import java.util.Scanner;

class Item {

String name;

double price;

int quantity;

public Item(String name, double price, int quantity) {

this.name = name;

this.price = price;

this.quantity = quantity;

}

}

class TreeNode {

Item item;

TreeNode left, right;

public TreeNode(Item item) {

this.item = item;

this.left = this.right = null;

}

}

class BST {

TreeNode root;

ArrayList<Item> hardcodedItems;

public BST() {

this.root = null;

this.hardcodedItems = new ArrayList<>();

}

public void insert(Item item) {

root = insertRecursive(root, item);

}

private TreeNode insertRecursive(TreeNode root, Item item) {

if (root == null) {

return new TreeNode(item);

}

if (item.price < root.item.price) {

root.left = insertRecursive(root.left, item);

} else if (item.price > root.item.price) {

root.right = insertRecursive(root.right, item);

}

return root;

}

public void insertHardcodedItem(Item item) {

hardcodedItems.add(item);

}

public void removeHardcodedItem(Item item) {

hardcodedItems.remove(item);

}

public void displayAllItems() {

displayAllItems(root);

for (Item item : hardcodedItems) {

System.***out***.println("Name: " + item.name + " - Price: " + item.price + " - Quantity: " + item.quantity);

}

}

private void displayAllItems(TreeNode root) {

if (root != null) {

displayAllItems(root.left);

System.***out***.println("Name: " + root.item.name + " - Price: " + root.item.price + " - Quantity: " + root.item.quantity);

displayAllItems(root.right);

}

}

public Item searchItemByName(String itemName) {

TreeNode foundNode = searchItem(root, itemName);

if (foundNode != null) {

return foundNode.item;

}

for (Item item : hardcodedItems) {

if (item.name.equals(itemName)) {

return item;

}

}

return null;

}

private TreeNode searchItem(TreeNode root, String itemName) {

if (root == null || root.item.name.equals(itemName)) {

return root;

}

if (itemName.compareTo(root.item.name) < 0) {

return searchItem(root.left, itemName);

} else {

return searchItem(root.right, itemName);

}

}

public void removeItemByName(String itemName) {

root = removeItemByName(root, itemName);

for (Item item : hardcodedItems) {

if (item.name.equals(itemName)) {

hardcodedItems.remove(item);

break;

}

}

}

private TreeNode removeItemByName(TreeNode root, String itemName) {

if (root == null) {

return root;

}

if (itemName.compareTo(root.item.name) < 0) {

root.left = removeItemByName(root.left, itemName);

} else if (itemName.compareTo(root.item.name) > 0) {

root.right = removeItemByName(root.right, itemName);

} else {

if (root.left == null) {

return root.right;

} else if (root.right == null) {

return root.left;

}

root.item = minValue(root.right);

root.right = removeItemByName(root.right, root.item.name);

}

return root;

}

private Item minValue(TreeNode root) {

Item minValue = root.item;

while (root.left != null) {

minValue = root.left.item;

root = root.left;

}

return minValue;

}

public ArrayList<Item> getAvailableItems() {

ArrayList<Item> items = new ArrayList<>();

getAvailableItems(root, items);

items.addAll(hardcodedItems);

return items;

}

private void getAvailableItems(TreeNode root, ArrayList<Item> items) {

if (root != null) {

getAvailableItems(root.left, items);

items.add(root.item);

getAvailableItems(root.right, items);

}

}

}

class Cart {

private ArrayList<Item> cart;

public Cart() {

cart = new ArrayList<>();

}

public void addItemToCart(String itemName, int quantity, BST bst) {

Item availableItem = bst.searchItemByName(itemName);

if (availableItem != null && availableItem.quantity >= quantity) {

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

cart.add(new Item(itemName, availableItem.price, 1));

}

availableItem.quantity -= quantity;

} else if (availableItem != null) {

System.***out***.println("Insufficient quantity available for item: " + itemName);

} else {

System.***out***.println("Item not found in the inventory: " + itemName);

}

}

public void removeItemFromCart(String itemName, int quantity, BST bst) {

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

Item removedItem = null;

for (Item item : cart) {

if (item.name.equals(itemName)) {

removedItem = item;

break;

}

}

if (removedItem != null) {

cart.remove(removedItem);

Item originalItem = bst.searchItemByName(removedItem.name);

originalItem.quantity += 1;

} else {

System.***out***.println("Item not found in the cart: " + itemName);

}

}

}

public void displayCart() {

System.***out***.println("Items in the cart:");

for (Item item : cart) {

System.***out***.println("Name: " + item.name + " - Price: " + item.price);

}

}

public double calculateTotalBill() {

double totalBill = 0.0;

for (Item item : cart) {

totalBill += item.price;

}

return totalBill;

}

public void checkout() {

double totalBill = calculateTotalBill();

System.***out***.println("Checkout - Total Bill: " + totalBill);

}

}

public class ShoppingCart {

public static void main(String[] args) {

BST bst = new BST();

Scanner scanner = new Scanner(System.***in***);

// Adding sample items to the inventory

bst.insertHardcodedItem(new Item("SamsungA23", 18000, 4));

bst.insertHardcodedItem(new Item("RedmiNote11", 13000, 8));

bst.insertHardcodedItem(new Item("OppoA15", 15500, 9));

int choice;

do {

System.***out***.println("Menu:");

System.***out***.println("1. Seller (Owner)");

System.***out***.println("2. User (Customer)");

System.***out***.println("3. Exit");

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

choice = scanner.nextInt();

System.***out***.println();

switch (choice) {

case 1:

// Seller (Owner) menu

int sellerChoice;

do {

System.***out***.println("Seller Menu:");

System.***out***.println("1. Add product to inventory");

System.***out***.println("2. Remove product from inventory");

System.***out***.println("3. Display all available products");

System.***out***.println("4. Exit (Seller)");

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

sellerChoice = scanner.nextInt();

switch (sellerChoice) {

case 1:

// Add product to inventory

System.***out***.print("Enter product name: ");

String name = scanner.next();

System.***out***.print("Enter product price: ");

double price = scanner.nextDouble();

System.***out***.print("Enter product quantity: ");

int quantity = scanner.nextInt();

bst.insert(new Item(name, price, quantity));

System.***out***.println();

break;

case 2:

// Remove product from inventory

System.***out***.print("Enter product name to remove: ");

String nameToRemove = scanner.next();

bst.removeItemByName(nameToRemove);

System.***out***.println(nameToRemove + " product removed successfully!");

System.***out***.println();

break;

case 3:

// Display all available products

ArrayList<Item> availableItems = bst.getAvailableItems();

System.***out***.println("Available products:");

for (Item item : availableItems) {

System.***out***.println("Name: " + item.name + " - Price: " + item.price + " - Quantity: " + item.quantity);

}

System.***out***.println();

break;

case 4:

System.***out***.println("Exiting Seller menu.");

System.***out***.println();

break;

default:

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

break;

}

} while (sellerChoice != 4);

break;

case 2:

// User (Customer) menu

int userChoice;

Cart cart = new Cart();

do {

System.***out***.println("User Menu:");

System.***out***.println("1. Display available products");

System.***out***.println("2. Add products to cart");

System.***out***.println("3. Remove products from cart");

System.***out***.println("4. Display cart");

System.***out***.println("5. Checkout");

System.***out***.println("6. Exit (User)");

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

userChoice = scanner.nextInt();

switch (userChoice) {

case 1:

// Display available products

ArrayList<Item> availableItems = bst.getAvailableItems();

System.***out***.println("Available products:");

for (Item item : availableItems) {

System.***out***.println("Name: " + item.name + " - Price: " + item.price + " - Quantity: " + item.quantity);

}

System.***out***.println();

break;

case 2:

// Add products to cart

System.***out***.print("Enter the name of the item you want to buy: ");

String itemNameToAdd = scanner.next();

System.***out***.print("Enter the quantity you want to buy: ");

int quantityToAdd = scanner.nextInt();

cart.addItemToCart(itemNameToAdd, quantityToAdd, bst);

System.***out***.println();

break;

case 3:

// Remove products from cart

System.***out***.print("Enter the name of the item you want to remove from the cart: ");

String itemNameToRemove = scanner.next();

System.***out***.print("Enter the quantity you want to remove: ");

int quantityToRemove = scanner.nextInt();

cart.removeItemFromCart(itemNameToRemove, quantityToRemove, bst);

System.***out***.println();

break;

case 4:

// Display cart

cart.displayCart();

System.***out***.println();

break;

case 5:

cart.checkout();

System.***out***.println();

break;

case 6:

System.***out***.println("Exiting User menu.");

System.***out***.println();

break;

default:

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

System.***out***.println();

break;

}

} while (userChoice != 6);

break;

case 3:

System.***out***.println("Exiting the shopping cart application.");

System.***out***.println();

break;

default:

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

System.***out***.println();

break;

}

} while (choice != 3);

}

}

OUTPUT-

Menu:

1. Seller (Owner)

2. User (Customer)

3. Exit

Enter your choice: 1

Seller Menu:

1. Add product to inventory

2. Remove product from inventory

3. Display all available products

4. Exit (Seller)

Enter your choice: 3

Available products:

Name: SamsungA23 - Price: 18000.0 - Quantity: 4

Name: RedmiNote11 - Price: 13000.0 - Quantity: 8

Name: OppoA15 - Price: 15500.0 - Quantity: 9

Seller Menu:

1. Add product to inventory

2. Remove product from inventory

3. Display all available products

4. Exit (Seller)

Enter your choice: 1

Enter product name: vivov9

Enter product price: 10400

Enter product quantity: 8

Seller Menu:

1. Add product to inventory

2. Remove product from inventory

3. Display all available products

4. Exit (Seller)

Enter your choice: 2

Enter product name to remove: RedmiNote11

RedmiNote11 product removed successfully!

Seller Menu:

1. Add product to inventory

2. Remove product from inventory

3. Display all available products

4. Exit (Seller)

Enter your choice: 3

Available products:

Name: vivov9 - Price: 10400.0 - Quantity: 8

Name: SamsungA23 - Price: 18000.0 - Quantity: 4

Name: OppoA15 - Price: 15500.0 - Quantity: 9

Seller Menu:

1. Add product to inventory

2. Remove product from inventory

3. Display all available products

4. Exit (Seller)

Enter your choice: 4

Exiting Seller menu.

Menu:

1. Seller (Owner)

2. User (Customer)

3. Exit

Enter your choice: 2

User Menu:

1. Display available products

2. Add products to cart

3. Remove products from cart

4. Display cart

5. Checkout

6. Exit (User)

Enter your choice: 1

Available products:

Name: vivov9 - Price: 10400.0 - Quantity: 8

Name: SamsungA23 - Price: 18000.0 - Quantity: 4

Name: OppoA15 - Price: 15500.0 - Quantity: 9

User Menu:

1. Display available products

2. Add products to cart

3. Remove products from cart

4. Display cart

5. Checkout

6. Exit (User)

Enter your choice: 2

Enter the name of the item you want to buy: SamsungA23

Enter the quantity you want to buy: 1

User Menu:

1. Display available products

2. Add products to cart

3. Remove products from cart

4. Display cart

5. Checkout

6. Exit (User)

Enter your choice: 2

Enter the name of the item you want to buy: OppoA15

Enter the quantity you want to buy: 2

User Menu:

1. Display available products

2. Add products to cart

3. Remove products from cart

4. Display cart

5. Checkout

6. Exit (User)

Enter your choice: 4

Items in the cart:

Name: SamsungA23 - Price: 18000.0

Name: OppoA15 - Price: 15500.0

Name: OppoA15 - Price: 15500.0

User Menu:

1. Display available products

2. Add products to cart

3. Remove products from cart

4. Display cart

5. Checkout

6. Exit (User)

Enter your choice: 3

Enter the name of the item you want to remove from the cart: OppoA15

Enter the quantity you want to remove: 1

User Menu:

1. Display available products

2. Add products to cart

3. Remove products from cart

4. Display cart

5. Checkout

6. Exit (User)

Enter your choice: 4

Items in the cart:

Name: SamsungA23 - Price: 18000.0

Name: OppoA15 - Price: 15500.0

User Menu:

1. Display available products

2. Add products to cart

3. Remove products from cart

4. Display cart

5. Checkout

6. Exit (User)

Enter your choice: 5

Checkout - Total Bill: 33500.0

User Menu:

1. Display available products

2. Add products to cart

3. Remove products from cart

4. Display cart

5. Checkout

6. Exit (User)

Enter your choice: 6

Exiting User menu.

Menu:

1. Seller (Owner)

2. User (Customer)

3. Exit

Enter your choice: 3

Exiting the shopping cart application.