Create a Restaurant Order Management System using Data Structures in java

(Major project)

Program :

```
import java.util.*;


class MenuItem {
    private String name;
    private double price;

    public MenuItem(String name, double price) {
        this.name = name;
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }
}
```

```
class OrderItem {
    private MenuItem menuItem;
    private int quantity;

    public OrderItem(MenuItem menuItem, int quantity) {
        this.menuItem = menuItem;
        this.quantity = quantity;
    }

    public MenuItem getMenuItem() {
        return menuItem;
    }

    public int getQuantity() {
        return quantity;
    }

    public double getTotalPrice() {
        return menuItem.getPrice() * quantity;
    }
}

class Order {
    private List<OrderItem> items;

    public Order() {
```

```
        this.items = new ArrayList<>();
    }

    public void addItem(OrderItem item) {
        items.add(item);
    }

    public List<OrderItem> getItems() {
        return items;
    }

    public double getTotalOrderPrice() {
        double total = 0;
        for (OrderItem item : items) {
            total += item.getTotalPrice();
        }
        return total;
    }
}

class OrderManager {
    private Map<Integer, Order> orders;
    private int orderIdCounter;

    public OrderManager() {
        this.orders = new HashMap<>();
        this.orderIdCounter = 1;
```

```
}

public int createOrder() {
    int orderId = orderIdCounter++;
    orders.put(orderId, new Order());
    return orderId;
}

public void addItemToOrder(int orderId, OrderItem item) {
    Order order = orders.get(orderId);
    if (order != null) {
        order.addItem(item);
    } else {
        System.out.println("Order not found.");
    }
}

public void displayOrder(int orderId) {
    Order order = orders.get(orderId);
    if (order != null) {
        System.out.println("Order " + orderId + ":");
        for (OrderItem item : order.getItems()) {
            System.out.println(item.getQuantity() + "x " + item.getMenuItem().getName() +
                    "- $" + item.getTotalPrice());
        }
        System.out.println("Total Price: $" + order.getTotalOrderPrice());
    } else {
```

```
            System.out.println("Order not found.");
        }
    }
}

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

        // Create an order
        int orderId = orderManager.createOrder();

        // Display menu
        MenuItem menu1 = new MenuItem("Nuggets", 4.0);
        MenuItem menu2 = new MenuItem("Pizza", 3.5);
        System.out.println("Menu:");
        System.out.println("1. " + menu1.getName() + "- $" + menu1.getPrice());
        System.out.println("2. " + menu2.getName() + "- $" + menu2.getPrice());

        // Take user input to add items to the order
        char addMore;
        do {
            System.out.print("Enter the item number (1 or 2): ");
            int itemNumber = scanner.nextInt();
            System.out.print("Enter the quantity: ");
            int quantity = scanner.nextInt();
```

```
            MenuItem selectedMenuItem = (itemNumber == 1) ? menu1 : menu2;
            OrderItem orderItem = new OrderItem(selectedMenuItem, quantity);

            orderManager.addItemToOrder(orderId, orderItem);

            System.out.print("Do you want to add more items? (y/n): ");
            addMore = scanner.next().charAt(0);
        } while (addMore == 'y' || addMore == 'Y');

        // Display the final order
        orderManager.displayOrder(orderId);
    }
}
```

output:

Menu:

1. Nuggets - $4.0

2. Pizza - $3.5

Enter the item number (1 or 2): 1

Enter the quantity: 3

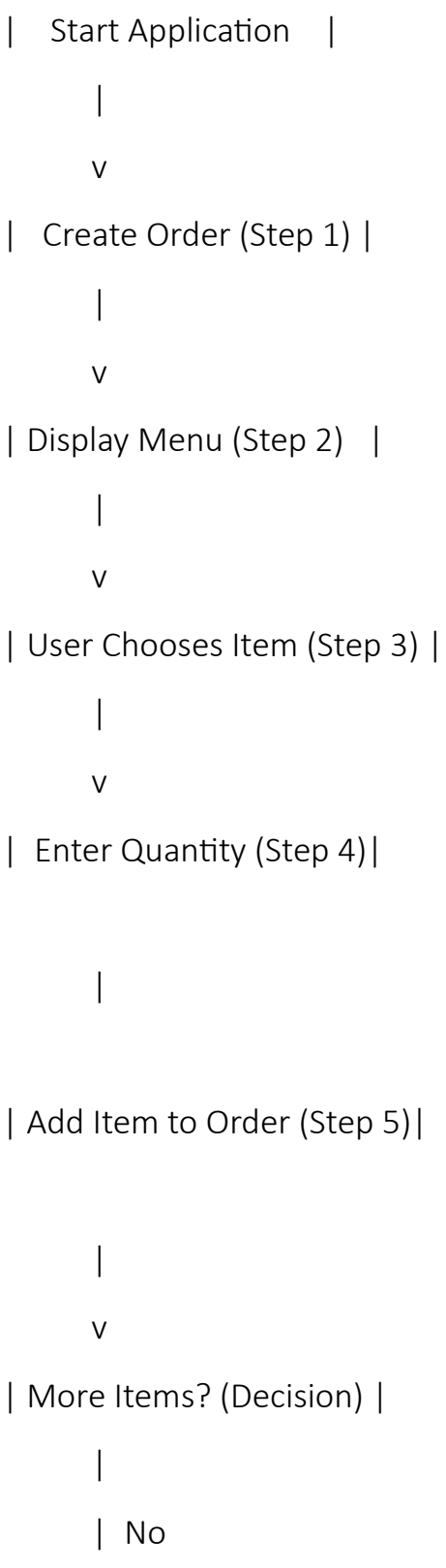
Do you want to add more items? (y/n): n

Order 1:

3x Nuggets - $12.0

Total Price: $12.0

Flow chart of program:


| Start Application |
v
| Create Order (Step 1) |
v
| Display Menu (Step 2) |
v
| User Chooses Item (Step 3) |
v
| Enter Quantity (Step 4)|
|
| Add Item to Order (Step 5)|
v
| More Items? (Decision) |
| No

v

| Display Final Order (Step 6)|

|

v

| End Application |

Git hub link:

https://github.com/Sameerabd386/root

Linked in link:

https://www.linkedin.com/in/mohammad-sameer-6aba4222a

Submitted by :

Mohammad Sameer