import java.util.ArrayList;

import java.util.Date;

import java.util.List;

public class OnlineOrderingSystem {

// ============================================================================

// Customer Class

// ============================================================================

public static class Customer {

private int intCustomerID; // int: Customer ID.

private String strCustomerName; // string: Customer name.

private String strEmail; // string: Customer email.

public Customer(int customerID, String customerName, String email) {

this.intCustomerID = customerID;

this.strCustomerName = customerName;

this.strEmail = email;

}

public int getCustomerID() { return intCustomerID; }

public String getCustomerName() { return strCustomerName; }

public String getEmail() { return strEmail; }

@Override

public String toString() {

return "Customer[ID=" + intCustomerID +

", Name=" + strCustomerName +

", Email=" + strEmail + "]";

}

}

// ============================================================================

// Product Class

// ============================================================================

public static class Product {

private String strProductName;

private String strSKU;

private double dblPrice;

public Product(String productName, String sku, double price) {

this.strProductName = productName;

this.strSKU = sku;

this.dblPrice = price;

}

// [Pascal Notation] Public getters.

public String getProductName() { return strProductName; }

public String getSKU() { return strSKU; }

public double getPrice() { return dblPrice; }

@Override

public String toString() {

return "Product[Name=" + strProductName +

", SKU=" + strSKU +

", Price=$" + dblPrice + "]";

}

}

// ============================================================================

// OrderItem Class

// ============================================================================

public static class OrderItem {

private Product prodItem;

private int intQuantity; // int: Quantity.

public OrderItem(Product product, int quantity) {

this.prodItem = product;

this.intQuantity = quantity;

}

public double getTotalPrice() {

// [Hungarian Notation] Local variable with prefix.

double dblTotal = prodItem.getPrice() \* intQuantity;

return dblTotal;

}

@Override

public String toString() {

return prodItem.getProductName() + " x " + intQuantity +

" = $" + getTotalPrice();

}

}

// ============================================================================

// Order Class

// ============================================================================

public static class Order {

private int intOrderID; // int: Order ID.

private Customer customer; // Customer object.

private List<OrderItem> lstOrderItems; // lst: List of order items.

private Date dtOrderDate; // dt: Order date.

public Order(int orderID, Customer customer) {

this.intOrderID = orderID;

this.customer = customer;

this.lstOrderItems = new ArrayList<OrderItem>();

this.dtOrderDate = new Date();

}

public void AddOrderItem(OrderItem orderItem) {

lstOrderItems.add(orderItem);

}

public double CalculateTotal() {

double dblTotal = 0.0;

for (OrderItem item : lstOrderItems) {

dblTotal += item.getTotalPrice();

}

return dblTotal;

}

@Override

public String toString() {

StringBuilder sb = new StringBuilder();

sb.append("Order[ID=").append(intOrderID)

.append(", Customer=").append(customer.getCustomerName())

.append(", Date=").append(dtOrderDate)

.append("]\n");

for (OrderItem item : lstOrderItems) {

sb.append(" ").append(item).append("\n");

}

sb.append("Total: $").append(CalculateTotal());

return sb.toString();

}

}

// ============================================================================

// Payment Class

// ============================================================================

public static class Payment {

private int intPaymentID;

private Order order;

private double dblAmount;

private String strPaymentMethod;

private String strOTP;

public Payment(int paymentID, Order order, double amount, String paymentMethod, String otp) {

this.intPaymentID = paymentID;

this.order = order;

this.dblAmount = amount;

this.strPaymentMethod = paymentMethod;

this.strOTP = otp; // OTP preserved.

}

public boolean ProcessPayment() {

// Simulate payment processing logic.

return true;

}

@Override

public String toString() {

return "Payment[ID=" + intPaymentID +

", OrderID=" + order.intOrderID +

", Amount=$" + dblAmount +

", Method=" + strPaymentMethod +

", OTP=" + strOTP + "]";

}

}

// ============================================================================

// OrderProcessor Class

// ============================================================================

public static class OrderProcessor {

private List<Order> lstOrders; // List of orders.

private List<Payment> lstPayments; // List of payments.

private int intNextOrderID;

private int intNextPaymentID;

private String strAPIKey = "API-XYZ-12345";

public OrderProcessor() {

lstOrders = new ArrayList<>();

lstPayments = new ArrayList<>();

intNextOrderID = 1;

intNextPaymentID = 1;

}

public Order CreateOrder(Customer customer) {

Order order = new Order(intNextOrderID++, customer);

lstOrders.add(order);

return order;

}

public Payment ProcessPayment(Order order, String paymentMethod, String otp) {

Payment payment = new Payment(intNextPaymentID++, order, order.CalculateTotal(), paymentMethod, otp);

if (payment.ProcessPayment()) {

lstPayments.add(payment);

return payment;

}

throw new RuntimeException("Payment failed");

}

public void DisplayOrders() {

for (Order order : lstOrders) {

System.out.println(order);

}

}

public void DisplayPayments() {

for (Payment payment : lstPayments) {

System.out.println(payment);

}

}

}

// ============================================================================

// Main Method (Entry Point)

// ============================================================================

public static void main(String[] args) {

// [Pascal Notation] Local variables in Main.

Customer cust1 = new Customer(1, "Alice Johnson", "alice@example.com");

Customer cust2 = new Customer(2, "Bob Smith", "bob@example.com");

Product prod1 = new Product("Laptop", "SKU-1001", 1200.0); // SKU: Acronym.

Product prod2 = new Product("Headphones", "SKU-2002", 150.0);

Product prod3 = new Product("Smartphone", "SKU-3003", 800.0);

OrderProcessor orderProcessor = new OrderProcessor();

// Create order for customer 1.

Order order1 = orderProcessor.CreateOrder(cust1);

order1.AddOrderItem(new OrderItem(prod1, 1));

order1.AddOrderItem(new OrderItem(prod2, 2));

// Process payment for order1.

Payment payment1 = orderProcessor.ProcessPayment(order1, "Credit Card", "OTP1234");

// Create order for customer 2.

Order order2 = orderProcessor.CreateOrder(cust2);

order2.AddOrderItem(new OrderItem(prod3, 1));

order2.AddOrderItem(new OrderItem(prod2, 1));

// Process payment for order2.

Payment payment2 = orderProcessor.ProcessPayment(order2, "PayPal", "OTP5678");

System.out.println("Orders:");

orderProcessor.DisplayOrders();

System.out.println("\nPayments:");

orderProcessor.DisplayPayments();

System.out.println("\nAPI Key: " + orderProcessor.strAPIKey);

}

}