

Inheritance Question

Payment Processing System

Create a class hierarchy for a payment processing system. Define a base class `Payment` with a method `processPayment()`. Then, create two subclasses, `CreditCardPayment` and `PayPalPayment`, which provide their own implementations of the `processPayment()` method.

1. Implement the `Payment` class and its subclasses.
2. Create a list of `Payment` objects, including both credit card and `PayPal` payments.
3. Iterate through the list and call the `processPayment()` method on each payment, demonstrating runtime polymorphism.

```
2
3  @ public abstract class Payment { 3 usages 3 inheritors
4      int amount = 0; // actual Amount of Payment 3 usages
5      public abstract PaymentStatus processPayment(int amount) ; no usages 3 implementations
6
7      public int getAmount() { return amount; } no usages
8
9      public void setAmount(int amount) { this.amount = amount; } no usages
10
11     public void doPrint() no usages
12     {
13         System.out.println("Amount to be Paid : - "+amount);
14     }
15 }
16
17 enum PaymentStatus { 4 usages
18     PaymentMade, no usages
19     Approved, no usages
20     Denied, no usages
21     Failed no usages
22 }
23
```

```

24 class CashOnDelivery extends Payment { no usages
25     int locationPincode; no usages
26     boolean validLocation; no usages
27
28     @Override no usages
29     public PaymentStatus processPayment(int amount) {
30         /* cash on delivery is not applicable for specific locations*/
31         /*if invalid location : return PaymentStatus.Denied*/
32         /*else return : Approved*/
33
34         // super.amount will be same argument amount
35         return null;
36     }
37
38     public boolean isValidLocation() { no usages
39         long invalidLocations[] = {110014, 110016, 110033};
40         return false;
41     }
42 }
43

```

```

46
47 class UPIPayment extends Payment { 1 usage
48     @Override 4 usages
49     public PaymentStatus processPayment(int amount) {
50         // provide 5% cash back
51         // PaymentStatus : success
52         return null;
53     }
54 }
55
56 class CreditCard extends Payment { 1 usage
57     int maxLimit = 50000; no usages
58     @Override 4 usages
59     public PaymentStatus processPayment(int amount) {
60         // Payment denied if amount is greater than maxLimit
61         // Payment status : Failed
62         // Add 3% extra surcharge also
63         return null;
64     }
65 }
66

```

```

2
3 public class MainPaymentClass {
4     public static void main(String[] args) {
5
6         Payment p;
7
8         // ---- first test case ----
9         p = new CashOnDelivery( locationPincode: 110014);
10        PaymentStatus status = p.processPayment( amount: 68000);
11        if(status == PaymentStatus.Approved )
12        {
13            // print the payment amount to be made
14        }
15        else {
16            // print the error message
17        }
18        p = new CashOnDelivery( locationPincode: 110048);
19        status = p.processPayment( amount: 68000);
20        if(status == PaymentStatus.Approved )
21        {
22            // print the payment amount to be made
23        }
24        else {
25            // print the error message
26        }
27
28        // --- Second Test Case ----
29
30        // --- Second Test Case ----
31
32        p = new UPIPayment();
33        status = p.processPayment( amount: 60000);
34        if(status == PaymentStatus.Approved )
35        {
36            // print the payment amount to be made
37            // in this case output should be 57000
38            p.doPrint(); // cannot change this code
39        }
40        else {
41            // print the error message
42        }
43    } //end main
44 } //end class

```

```
42
43 // -----Third Test case -----
44 p = new CreditCard();
45 status = p.processPayment( amount: 60000);
46 if(status == PaymentStatus.Approved )
47 {
48     // print the message if approved
49 }
50 else {
51     // print the error message
52 }
53
54 status = p.processPayment( amount: 40000);
55 if(status == PaymentStatus.Approved )
56 {
57     // print the message if approved
58     // Payment amount 41200
59 }
60 else {
61     // print the error message
62 }
63 }//end main
64 }//end class
65
```