Exercise 8: Strategy Pattern- Payment Selection App

Approach and Understanding:

In this program, I made a **simple payment system** where the user can **choose how to pay** — either by **Card** or **PayPal**. The system lets the user **enter the payment type and amount** every time, and it uses the **Strategy Pattern** to select the correct payment method while the program is running.

**Code**

import java.util.Scanner;

interface Pay {

void pay(double amt);

}

class Card implements Pay { String num;

Card(String num) { this.num = num;

}

public void pay(double amt) {

System.out.println("Paid $" + amt + " using card " + num);

}

}

class PayPal implements Pay { String mail;

PayPal(String mail) { this.mail = mail;

}

public void pay(double amt) {

System.out.println("Paid $" + amt + " using PayPal " + mail);

}

}

class UsePay { Pay way;

UsePay(Pay way) { this.way = way;

}

void doPay(double amt) { way.pay(amt);

}

}

public class PayApp {

public static void main(String[] args) { Scanner sc = new Scanner(System.in); UsePay use = null;

while (true) {

System.out.print("\nType 'card', 'paypal' or 'stop': "); String input = sc.nextLine();

if (input.equalsIgnoreCase("stop")) { break;

}

System.out.print("Amount: ");

double amt = Double.parseDouble(sc.nextLine());

if (input.equalsIgnoreCase("card")) { System.out.print("Card number: "); String num = sc.nextLine();

use = new UsePay(new Card(num));

} else if (input.equalsIgnoreCase("paypal")) { System.out.print("PayPal email: ");

String mail = sc.nextLine();

use = new UsePay(new PayPal(mail));

} else {

System.out.println("Try again, wrong input.");

continue;

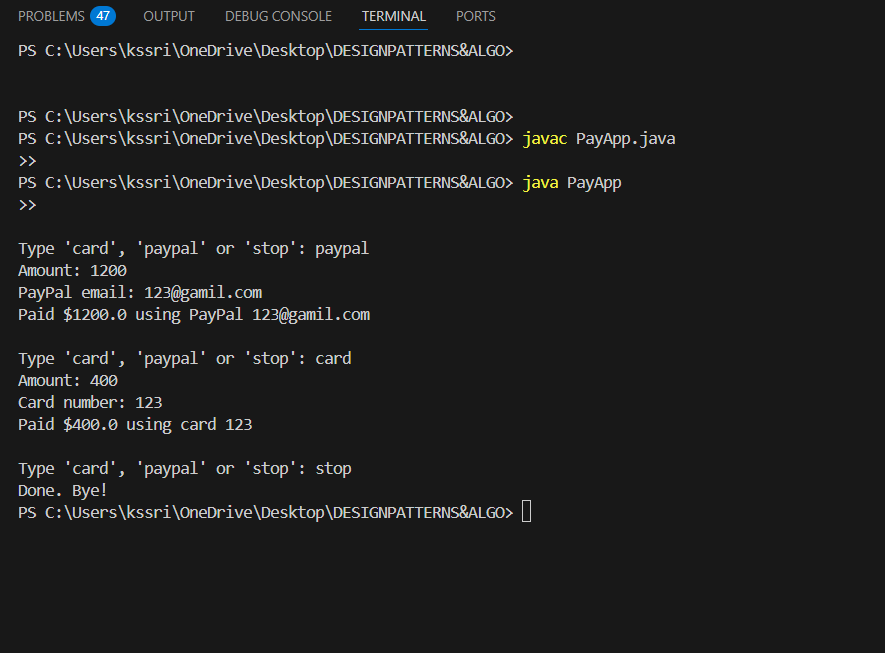
}

use.doPay(amt);

}

sc.close(); System.out.println("Done. Bye!");

}



}

**OUTPUT:**