**DIGITAL NURTURE 4.0 DEEP SKILLING JAVA FSE-WEEK1**

**NAME: SIVITHA GUNASEKARAN**

**SUPERSET ID: 6413354**

**WEEK 1: DESIGN PATTERNS AND PRINCIPLES**

**Exercise 4: Implementing the Adapter Pattern**

**Scenario:**

You are developing a payment processing system that needs to integrate with multiple third-party payment gateways with different interfaces. Use the Adapter Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **AdapterPatternExample**.
2. **Define Target Interface:**
   * Create an interface **PaymentProcessor** with methods like **processPayment()**.
3. **Implement Adaptee Classes:**
   * Create classes for different payment gateways with their own methods.
4. **Implement the Adapter Class:**
   * Create an adapter class for each payment gateway that implements PaymentProcessor and translates the calls to the gateway-specific methods.
5. **Test the Adapter Implementation:**
   * Create a test class to demonstrate the use of different payment gateways through the adapter.

**CODE SAMPLES:**

**AdapterPatternExample.java**

interface PaymentProcessor {

void processPayment(double amount);

}

class PayPalGateway {

public void makePayment(double amount) {

System.out.println("Payment of ₹" + amount + " made via PayPal.");

}

}

class StripeGateway {

public void stripeCharge(double amount) {

System.out.println("Payment of ₹" + amount + " made via Stripe.");

}

}

class RazorpayGateway {

public void razorTransaction(double amount) {

System.out.println("Payment of ₹" + amount + " made via Razorpay.");

}

}

class PayPalAdapter implements PaymentProcessor {

private PayPalGateway paypal;

public PayPalAdapter() {

this.paypal = new PayPalGateway();

}

public void processPayment(double amount) {

paypal.makePayment(amount);

}

}

class StripeAdapter implements PaymentProcessor {

private StripeGateway stripe;

public StripeAdapter() {

this.stripe = new StripeGateway();

}

public void processPayment(double amount) {

stripe.stripeCharge(amount);

}

}

class RazorpayAdapter implements PaymentProcessor {

private RazorpayGateway razorpay;

public RazorpayAdapter() {

this.razorpay = new RazorpayGateway();

}

public void processPayment(double amount) {

razorpay.razorTransaction(amount);

}

}

public class AdapterPatternExample {

public static void main(String[] args) {

PaymentProcessor paypal = new PayPalAdapter();

PaymentProcessor stripe = new StripeAdapter();

PaymentProcessor razorpay = new RazorpayAdapter();

paypal.processPayment(1200.0);

stripe.processPayment(850.5);

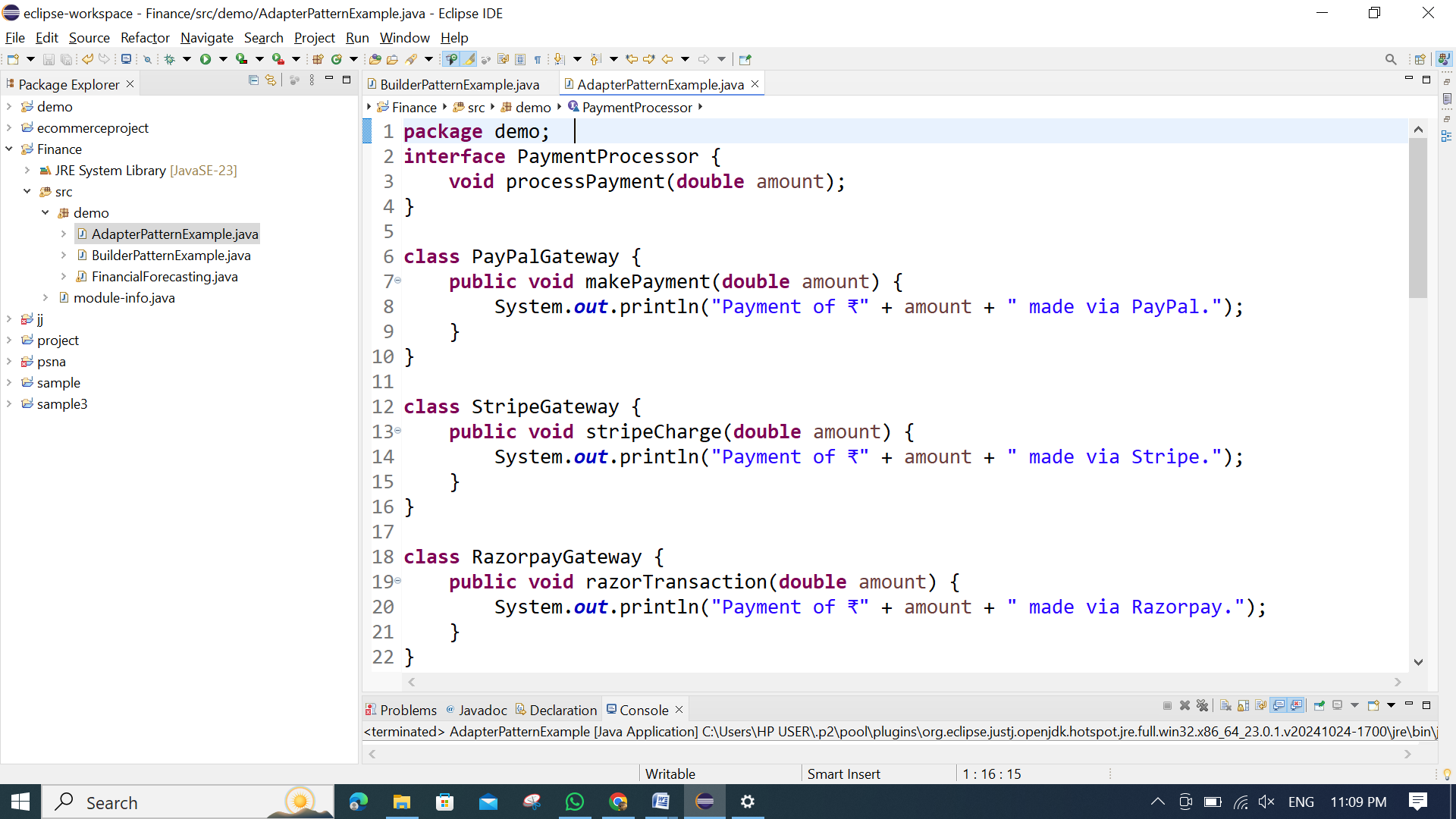
razorpay.processPayment(2300.75);

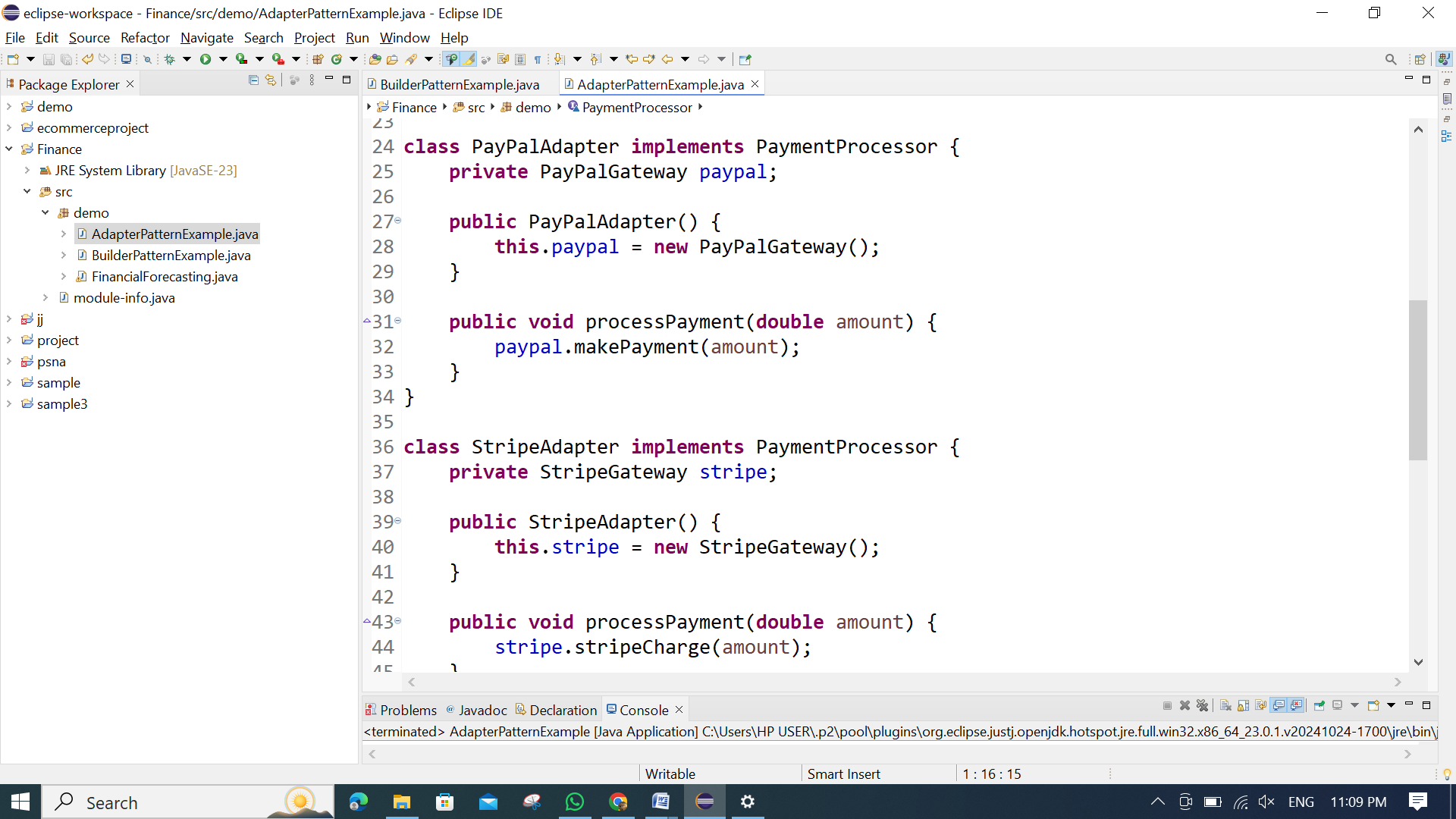
}

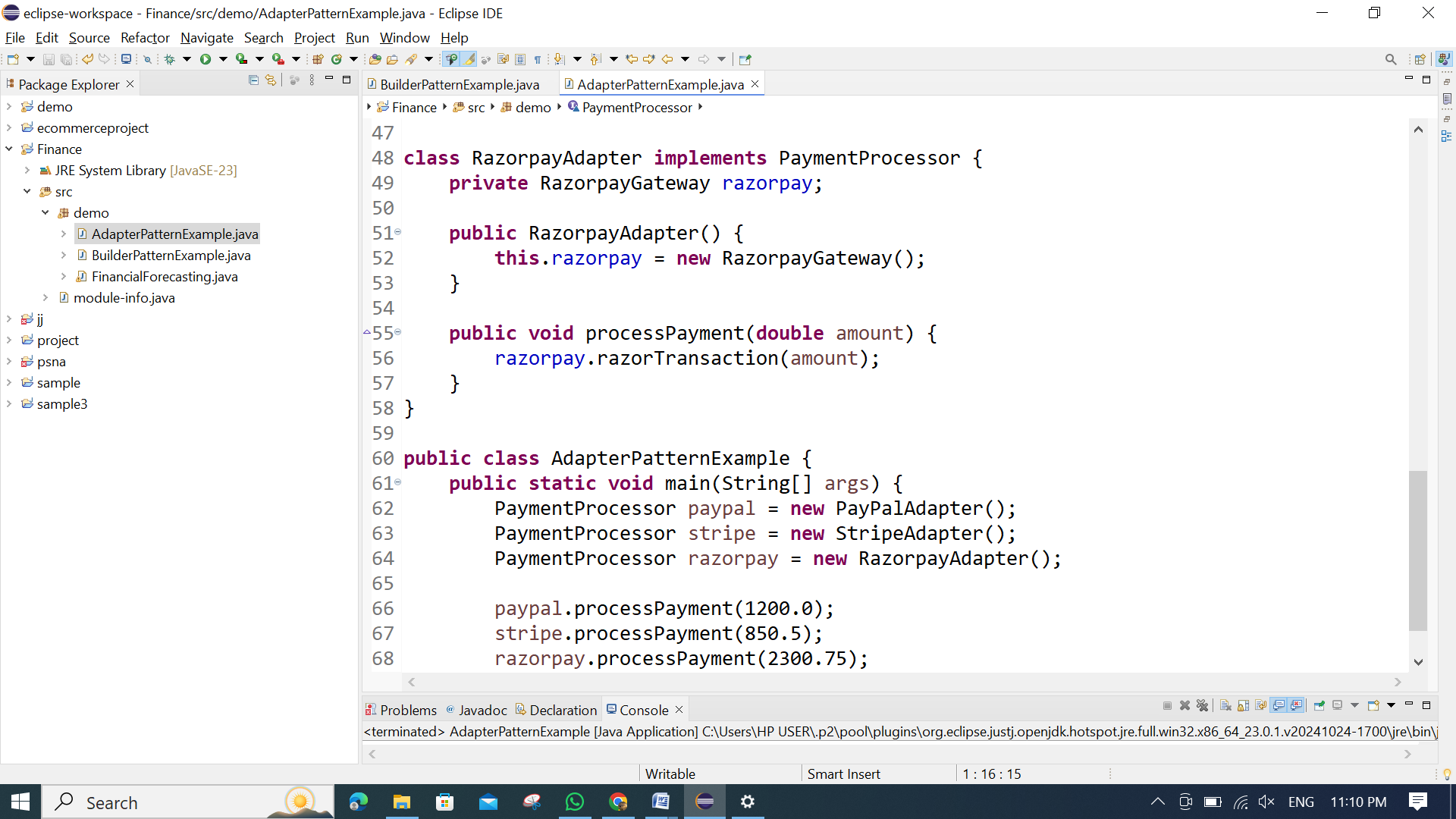
}

**MY SCREENSHOT PROOFS:**

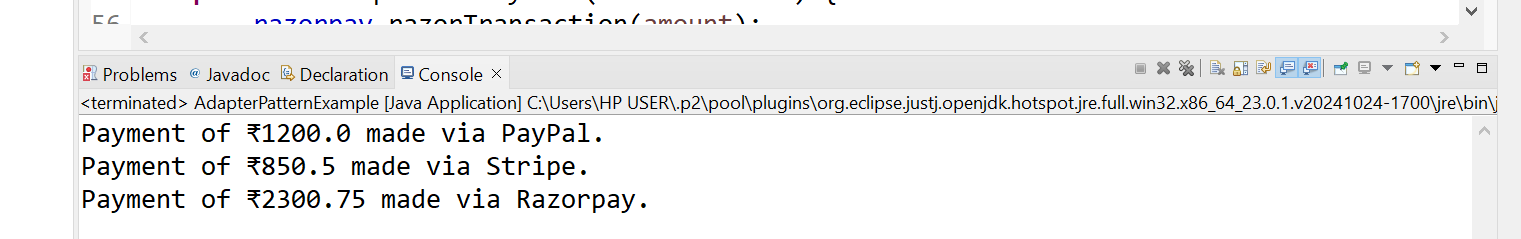
**AdapterPatternExample.java**







**OUTPUT:**

****