WEEK – 1

Implementing the Adapter Pattern

Nipuna Amanapu

[namanapu@gitam.in](mailto:namanapu@gitam.in)

superset id : 6432842

**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.

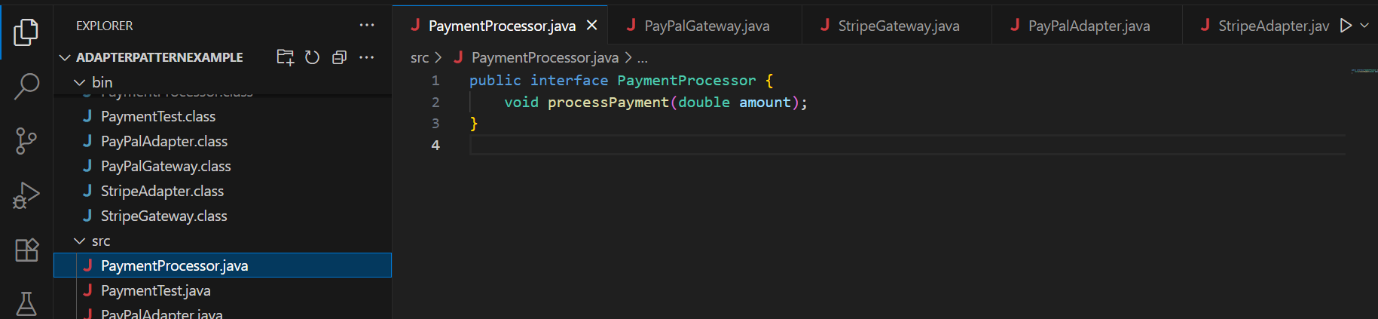
Solution:

1)PaymentProcessor.java

public interface PaymentProcessor {

void processPayment(double amount);

}



2) PayPalGateway.java

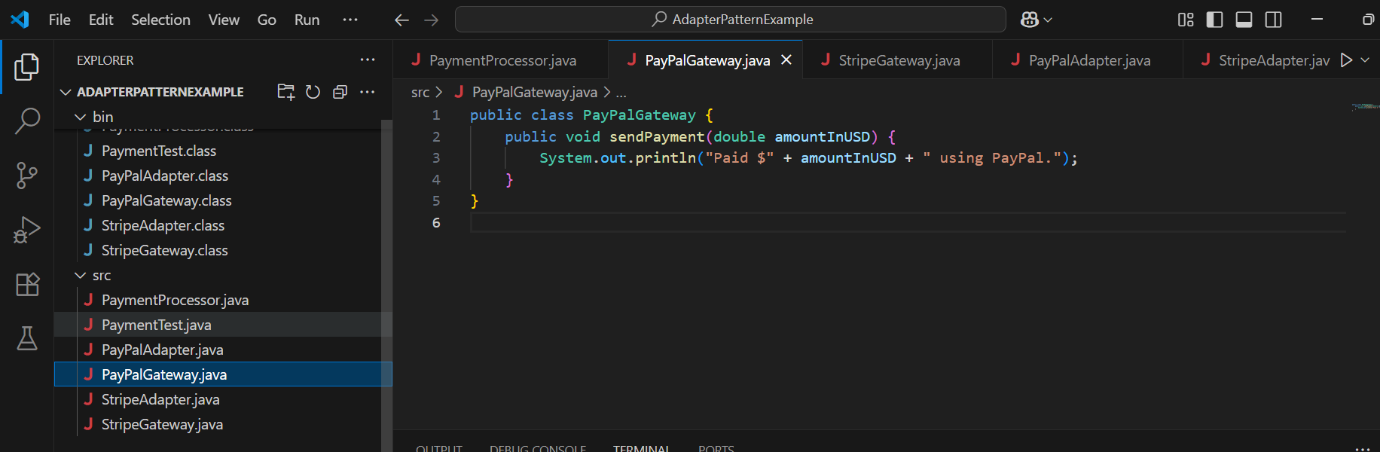
public class PayPalGateway {

public void sendPayment(double amountInUSD) {

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

}

}



3) StripeGateway.java

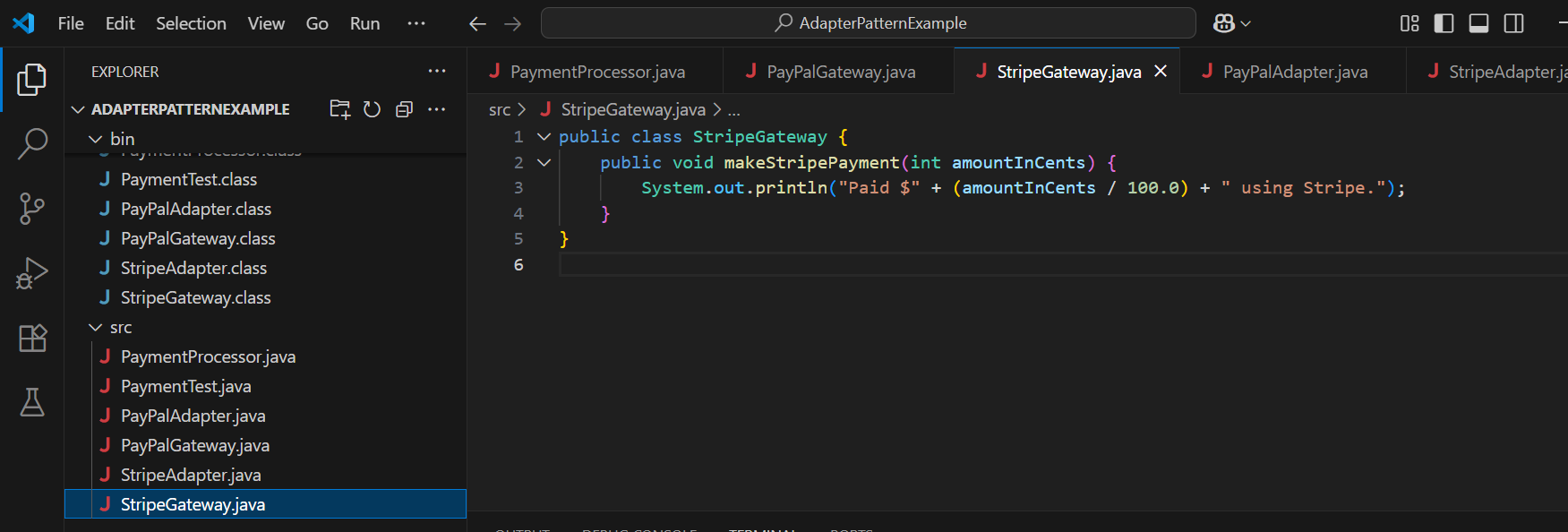
public class StripeGateway {

public void makeStripePayment(int amountInCents) {

System.out.println("Paid $" + (amountInCents / 100.0) + " using Stripe.");

}

}



4) PayPalAdapter.java

public class PayPalAdapter implements PaymentProcessor {

private PayPalGateway paypal;

public PayPalAdapter(PayPalGateway paypal) {

this.paypal = paypal;

}

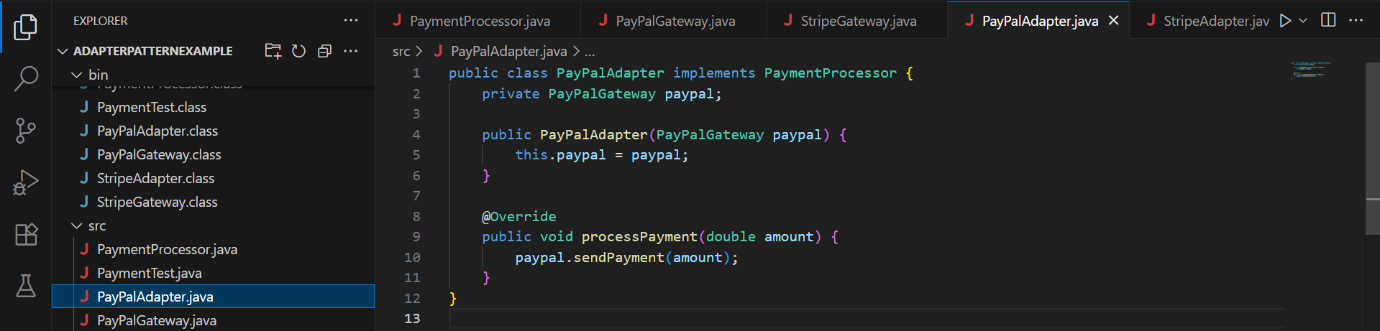
@Override

public void processPayment(double amount) {

paypal.sendPayment(amount);

}

}



5) StripeAdapter.java

public class StripeAdapter implements PaymentProcessor {

private StripeGateway stripe;

public StripeAdapter(StripeGateway stripe) {

this.stripe = stripe;

}

@Override

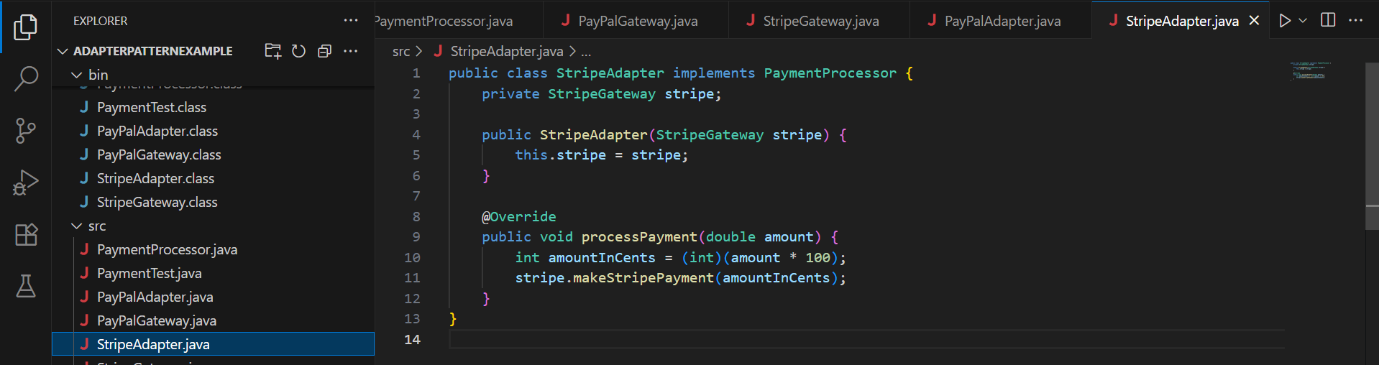
public void processPayment(double amount) {

int amountInCents = (int)(amount \* 100);

stripe.makeStripePayment(amountInCents);

}

}



6) PaymentTest.java

public class PaymentTest {

public static void main(String[] args) {

PayPalGateway paypal = new PayPalGateway();

PaymentProcessor paypalProcessor = new PayPalAdapter(paypal);

paypalProcessor.processPayment(100.50);

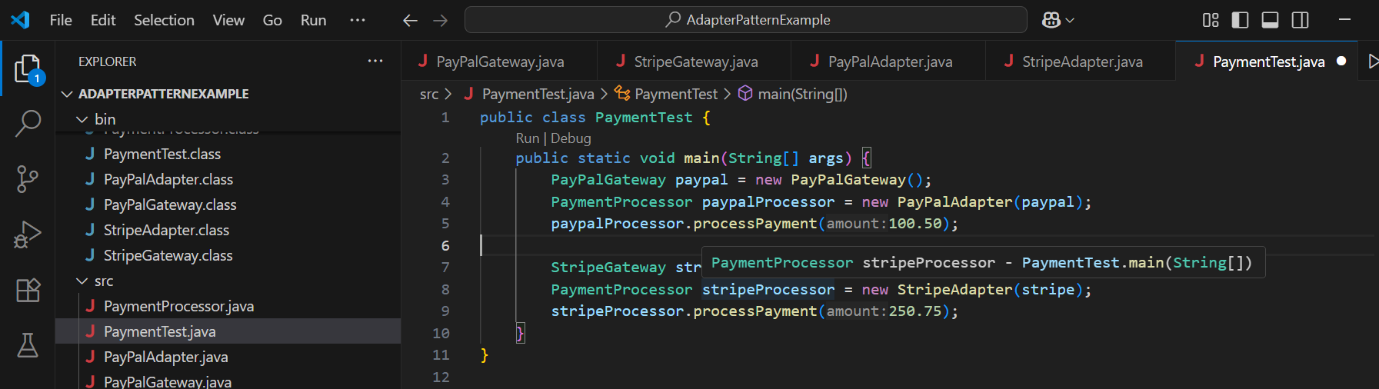
StripeGateway stripe = new StripeGateway();

PaymentProcessor stripeProcessor = new StripeAdapter(stripe);

stripeProcessor.processPayment(250.75);

}

}



Output:

