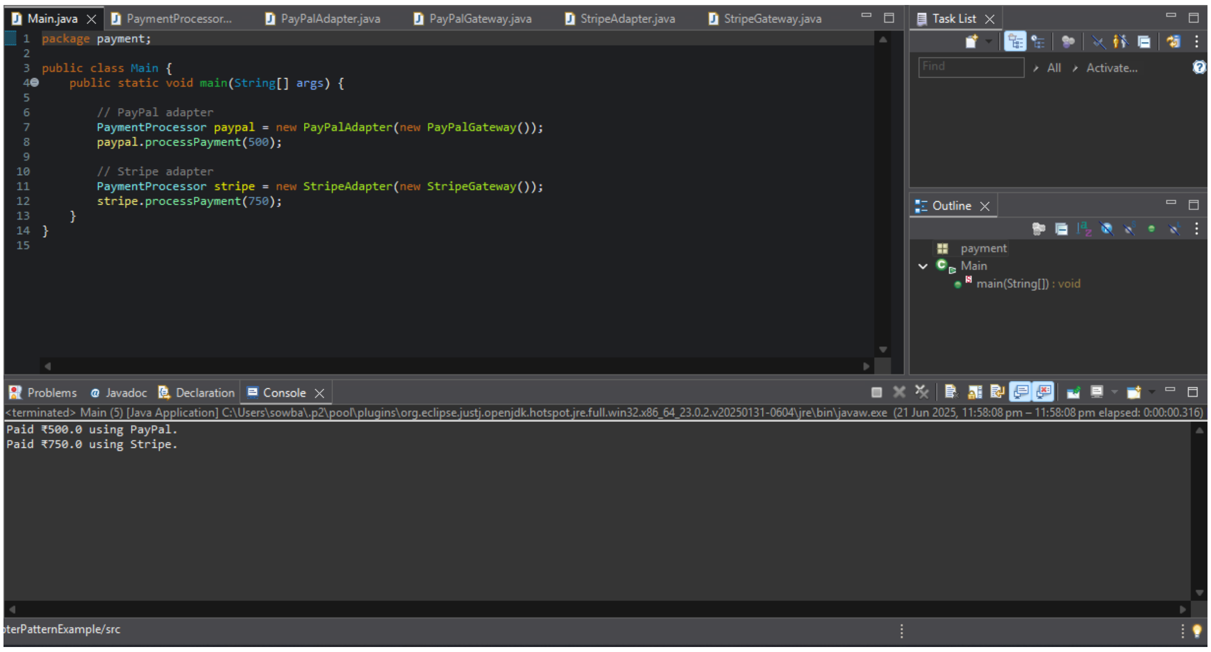
Exercise 4: Implementing the Adapter Pattern

OUTPUT:



```java
package payment;

public class Main {
    public static void main(String[] args) {

        // PayPal adapter
        PaymentProcessor paypal = new PayPalAdapter(new PayPalGateway());
        paypal.processPayment(500);

        // Stripe adapter
        PaymentProcessor stripe = new StripeAdapter(new StripeGateway());
        stripe.processPayment(750);
    }
}
```

```
<terminated> Main (5) [Java Application] C:\Users\sowba\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.2.v20250131-0604\jre\bin\javaw.exe  (21 Jun 2025, 11:58:08 pm – 11:58:08 pm elapsed: 0:00:00.316)
Paid ₹500.0 using PayPal.
Paid ₹750.0 using Stripe.
```

The Adapter Pattern allows integration of multiple third-party payment gateways with different interfaces by using adapter classes. It promotes flexibility, code reusability, and seamless compatibility without changing existing code.