1. **PaymentProcessor Interface:**
    - Declares a method **pay(int amount)** that all payment processors must implement.
2. **CreditCard Class:**
    - Implements the **PaymentProcessor** interface.
    - Collects information such as cardholder name, card number, and expiration date during the creation of a **CreditCard** object.
    - Overrides the **pay** method to process payments through a credit card.
3. **PayPal Class:**
    - Implements the **PaymentProcessor** interface.
    - Checks internet connection during the creation of a **PayPal** object.
    - Overrides the **pay** method to process payments through PayPal.
4. **BitCoin Class:**
    - Implements the **PaymentProcessor** interface.
    - Collects a transaction input address during the creation of a **BitCoin** object.
    - Overrides the **pay** method to process payments through BitCoin.
5. **Order Class:**
    - Takes an amount and a payment processor as parameters during its creation.
    - Has a method **process** that calls the **pay** method of the provided payment processor.
6. **Strategy Class (main class):**
    - Creates instances of **Order** based on the user's choice of payment method.
    - The user can choose between Credit Card, PayPal, BitCoin, or exit the shopping cart.
    - Depending on the choice, an instance of **Order** is created with the corresponding payment processor, and the payment is processed.

In summary, the program allows users to choose a payment method (Credit Card, PayPal, or BitCoin) while shopping. The strategy pattern is used to switch between different payment strategies without changing the client code (the main program). This promotes flexibility and makes it easy to add new payment methods in the future