Strategy Pattern

The Strategy Pattern is a behavioral design pattern that lets you define a family of algorithms, encapsulate each one in a separate class, and make them interchangeable at runtime.

Problem Statement

Imagine you are building an e-commerce shopping cart (like Amazon or Flipkart). At checkout, the user can choose different payment methods:

Credit Card

PayPal

UPI

Net Banking

Wallet

Without Strategy Pattern Challenges

Violation of Single Responsibility Principle

Violates Open/Closed Principle

Code Duplication & Inconsistency

Tight Coupling

Use cases

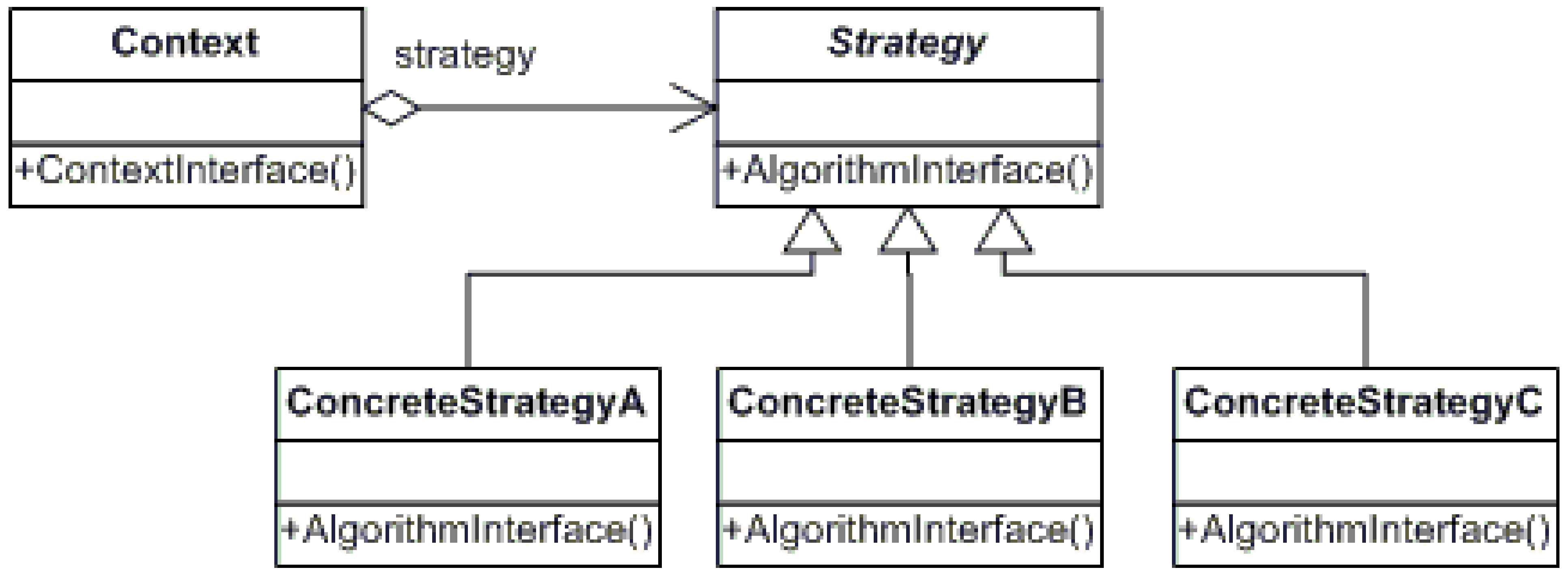
Payment systems – to switch between Credit Card, UPI, PayPal at runtime.

Discount calculation – to apply different discount rules without changing billing code.

Sorting logic – to change sorting criteria (price, rating, name) dynamically.

Authentication mechanisms – to support password, OTP, OAuth, or biometric login.

UML Diagram



It removes large if-else or switch blocks by encapsulating different algorithms into separate classes and selecting them at runtime.