

১। Singleton Pattern – MenuManager

MenuManager ক্লাস শুধুমাত্র একটি instance তৈরি করতে দেয় (`__instance`)।

কোডে `get_instance()` method ব্যবহার করে সেই একটাই instance বারবার রিটার্ন করা হয়।

এটি নিশ্চিত করে যে মেনুর তালিকা সিস্টেমজুড়ে এক জায়গা থেকেই ব্যবহৃত হচ্ছে।

২। Factory Pattern – OrderFactory

`OrderFactory.create_order(order_type)` method বিভিন্ন ধরনের Order (DineInOrder, TakeawayOrder, DeliveryOrder) তৈরি করে।

মেইন প্রোগ্রামে জানার দরকার নেই কোন ক্লাস কিভাবে তৈরি হচ্ছে – শুধু order type দিলে উপযুক্ত Object পাওয়া যায়।

৩। Strategy Pattern – PaymentStrategy

PaymentStrategy interface define করে এবং CashPayment, CardPayment, ও ThirdPartyPaymentAdapter বিভিন্ন কৌশল হিসেবে কাজ করে।

প্রোগ্রাম runtime-এ user এর ইনপুট অনুযায়ী payment strategy বেছে নেয়, ফলে payment করার পদ্ধতি পরিবর্তনযোগ্য।

৪। Decorator Pattern – SpecialRequestDecorator

SpecialRequestDecorator মূল Order ক্লাসের behavior না বদলে তার উপরে অতিরিক্ত ফিচার (special request) যোগ করে।

এর মাধ্যমে একটি order এর সাথে নতুন অনুরোধ যুক্ত করা যায়, মূল Order ক্লাস একেবারে touch না করেই।

৫ | Adapter Pattern – ThirdPartyPaymentAdapter

ThirdPartyPayment নামের ক্লাস PaymentStrategy interface ফলো করে না।

তাই ThirdPartyPaymentAdapter ব্যবহার করে ওকে adapt করা হয়েছে – যেন সে PaymentStrategy হিসেবে ব্যবহারযোগ্য হয়।

৬ | Observer Pattern – OrderNotifier ও Customer

OrderNotifier observer subject হিসেবে কাজ করে।

Customer অবজেক্টগুলো observer হিসেবে সাবসক্রাইব করে।

যখন কোনো update আসে, তখন সকল customer কে notify করে দেওয়া হয়।