

تقرير مشروع: نظام طلبات أونلاين (*Online Ordering System*)

اسم الطالبات: هاجر خالد راجح .

صفاء صالح سرب

قمر مراد قشیر.

المادة: *Software Design Patterns*

التاريخ: 2025/11/29

الملخص:

هذا المشروع يوضح تطبيق ثلاثة أنماط تصميم: *Factory (Creational)*, *Adapter (Structural)*, *Observer (Behavioral)*.

الغرض إظهار كيف ينجز الكود عند فصل مسؤولية إنشاء المنتجات، التعامل مع بوابة دفع خارجية، وإشعار أطراف مختلفة عندما تتغير حالة الطلب.

المكونات الأساسية:

: *Product, Pizza, Drink* للمنتج.

: *ProductFactory*. (*Factory Pattern*)

: *Order, OrderSubject* المُسؤول عن إعلام المراقبين.

OrderObserver, Customer,

نط *Observer* لإشعار *WarehouseManager* التغييرات.

ThirdPartyPaymentGateway, PaymentProcessor, PaymentAdapter محاكاة لمزود دفع خارجي وربطه عبر *Adapter*.

التنفيذ:

عند إنشاء *Order* يتم إعلام المشتركين بحالة "Created".

بعد إضافة العناصر يُحسب المجموع؛ يتم تنفيذ الدفع عبر *PaymentAdapter* الذي يستعمل *ThirdPartyPaymentGateway*.

عند نجاح الدفع يتم تغيير حالة الطلب إلى *Confirmed* ثم يتم محاكاة شحن وتسليم بتغيير الحالة إلى *Shipped* و *Delivered*؛ كل تغيير يرسل إشعاراً للمشتركين.

لماذا هذه الأنماط؟

: يفصل إنشاء المنتجات عن استخدامها، ما يسهل إضافة أنواع جديدة دون تغيير منادي الكود.

Adapter: يتيح لنا استخدام مزود دفع خارجي بواجهة مختلفة دون تغيير منطق النظام.

Observer: يحقق فصلًا مرزاً بين مصدر التغيير *Customer*, والمستهلكين (*Order*) و(*WarehouseManager*).

كيفية التشغيل:

1. وضع جميع ملفات `.java` في مجلد واحد.

`javac *.java .2`

`java Main .3`

اقتراحات للتطوير:

Decorator لإضافة خيارات (مثلاً إضافات للبيتزا).

حفظ الطلبات في قاعدة بيانات أو ملف `JSON`.
واجهة رسومية بسيطة لعرض الطلبات وحالتها.

الخلاصة:

المشروع يلبي متطلبات الواجب بتطبيق نمط من كل فئة وتصميم مكونات واضحة قابلة للتوسيع.

هل هذا مطلب دكتوره ف بدق