

تقرير مشروع: نظام طلبات أونلاين (Online)  
(Ordering System)  
اسم الطالبات :هاجر خالد راجح .  
صفاء صالح سرب  
قمر مراد قشير.  
المادة: Software Design Patterns  
التاريخ: 2025/11/29

الملخص:  
هذا المشروع مصغر يوضح تطبيق ثلاثة أنماط  
تصميم: Factory (Creational)، Adapter، Observer (Behavioral)، ((Structural)).  
الغرض إظهار كيف ينظم الكود عند فصل  
مسؤولية إنشاء المنتجات، التعامل مع بوابة دفع  
خارجية، وإشعار أطراف مختلفة عندما تتغير حالة  
الطلب.

المكونات الأساسية:

Product, Pizza, Drink : واجهة ومُنفذات  
للمنتج.

ProductFactory : مصنع لإنشاء المنتجات  
(Factory Pattern).

Order, OrderSubject : كائن الطلب والنظام  
المسؤول عن إعلام المراقبين.

OrderObserver, Customer,

WarehouseManager : نمط Observer لإشعار  
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 للمشاركين.

لماذا هذه الأنماط؟

Factory: 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**: يحدّد قِصلاً مرزاً بين مصدر التغيير  
(Order) والمستمعين (Customer, WarehouseManager).

**كيفية التشغيل:**

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

2. **javac \*.java**

3. **java Main**

**اقتراحات للتطوير:**

إضافة **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 دكتوراه ف بدف**