
# CORSO DI LAUREA TRIENNALE IN INFORMATICA - UNIFI

## METODOLOGIE DI PROGRAMMAZIONE A.A. 2021/2022

## “Gestione degli ordini di un supermercato”

Jonathan Alban,
matricola 6106113

Firenze, 10/01/22

INDICE

# 1 - Descrizione Del Progetto

Si vuole implementare la gestione degli ordini da parte dei clienti, in un supermercato che ha scelto come modalità di spesa tramite applicazione. La spesa viene fatta tramite applicazione, in cui il client potrà scegliere vari prodotti e successivamente avere la spesa spedita direttamente a casa. Ci sono prodotti che vengono venduti a pezzi e prodotti che vengono venduti a peso, come i prodotti del reparto orto-frutticolo.
I *"ProductByWeight"* hanno un loro prezzo e un peso, dove il calcolo del loro prezzo finale è il prodotto tra il prezzo e peso. I *"ProductByPiece"*, invece hanno un loro prezzo.
Il cliente sceglierà quali prodotti mettere nel suo carrello, solo alla fine viene fatto il calcolo finale del prezzo totale. Successivamente se il pagamento ha successo si procede alla finalizzazione dell'ordine.

## 2 - Funzionalità Del Sistema

- Due tipi di prodotti: prodotti venduti al peso e prodotti venduti a pezzo.
- Carrello che contiene una lista di prodotti e che calcolerà il prezzo finale.
- Pagamento in più metodi, tramite carta o contanti.
- Monitoraggio dinamico dello stato dell'ordine.

## 3 - Scelte Implementative

L'implementazione del codice è stato realizzato grazie ai principi della programmazione Object-Oriente e ai Design Pattern, al fine di rendere semplice e comprensibile il codice.
Si ha due tipi di prodotti: *ProductByPiece e ProductByWeight*. *ProductByPiece* ha un suo prezzo il quale sarà suo prezzo "finale". *ProductByPiece* ha un peso e un prezzo, il quale il suo prezzo "finale" sarà il prodotto tra prezzo e peso. Il calcolo del prezzo dei prodotti è stato realizzato tramite Visitor. È stato scelto questo pattern, in quanto se vogliamo modificare il calcolo del prezzo dovremmo rimettere mano nelle classi, in questo modo manteniamo la stessa struttura aggiungendo operazioni aggiuntive. Scelti i prodotti da metter nel carrello si può procedere con l'ordine. Prima di poter procedere con l'ordine viene controllato l'esito del pagamento. Per l'implementazione del pagamento è stato usato il pattern Strategy. Tramite questo pattern è possibile avere più

metodi di pagamento, *"MoneyPayment"* e *"CardPayment"*. Con il successo dell'esito del pagamento è possibile procedere all'ordine, il quale verrà registrato nello *"Store"*. Inoltre la classe *"Order"* ha un campo *shipping* il quale viene aggiornato se il nostro ordine è in fase di spedizione o no. Tramite il pattern Observer sarà possibile vedere lo stato dell'ordine: "false" = l'ordine non è ancora in fase di spedizione;  "true" l'ordine è pronto alla spedizione.
Il progetto è facilmente estendibile. Infatti è strutturato in modo da dare la possibilità di aggiungere o rimuovere classe concrete all'Observer, Strategy e Visitor.
Le classi e interfacce, come i vari test, sono state suddivise in pacchetti in base al loro ruolo all'interno del progetto.
Per i testare il codice e il suo corretto funzionamento sono stati fatti vari test per i metodi delle classe, tramite JUnit e la libreria AssertJ.