

2024/25

ODD

Object Design Document Digitronics



Object Design Document

CHIERCHIA PAOLO GIOVANNI, DI MICCO VINCENZO, ZOCCOLA DOMENICO

Revision History	1
Team Members	1
1. Introduzione.....	1
1.1 Definizione, acronimi e abbreviazioni	2
1.2 Riferimenti.....	2
2. Design pattern.....	2
2.1 Pattern Adapter	2
2.2 Pattern Façade.....	3
3. Glossario	4

Revision History

Data	Versione	Descrizione	Autori
22/12/2024	0.1	Façade pattern	Tutto il team
08/01/2025	0.2	Adapter pattern	GPC
16/01/2025	1.0	Revisione finale	Tutto il team

Team Members

Nome	Ruolo nel Progetto	Acronimo	Informazioni di contatto
Giovanni Paolo Chierchia	Team Member	GPC	g.chierchia8@studenti.unisa.it
Vincenzo Di Micco	Team Member	VDM	v.dimicco4@studenti.unisa.it
Domenico Zoccola	Team Member	DZ	d.zoccola3@studenti.unisa.it

1. Introduzione

Dopo aver completato i documenti RAD e SDD, abbiamo delineato in modo il funzionamento previsto del nostro sistema e gli obiettivi che intendiamo raggiungere, senza entrare nei dettagli specifici relativi all'implementazione. Il seguente documento mostra i Design pattern utilizzati.

1.1 Definizione, acronimi e abbreviazioni

- RAD: Requirements Analysis Document
- SDD: System Design Document
- ODD: Object Design Document
- Design pattern: è una soluzione progettuale generale e riutilizzabile per un problema comune che si presenta durante lo sviluppo software.

1.2 Riferimenti

- Slides del corso presenti sulla piattaforma e-learning;
- Libro di testo "Object Oriented Software Engineering Using UML Patterns and Java Prentice Hall 2010 Bernd Bruegge Allen H.Dutoit".
- RAD e SDD.

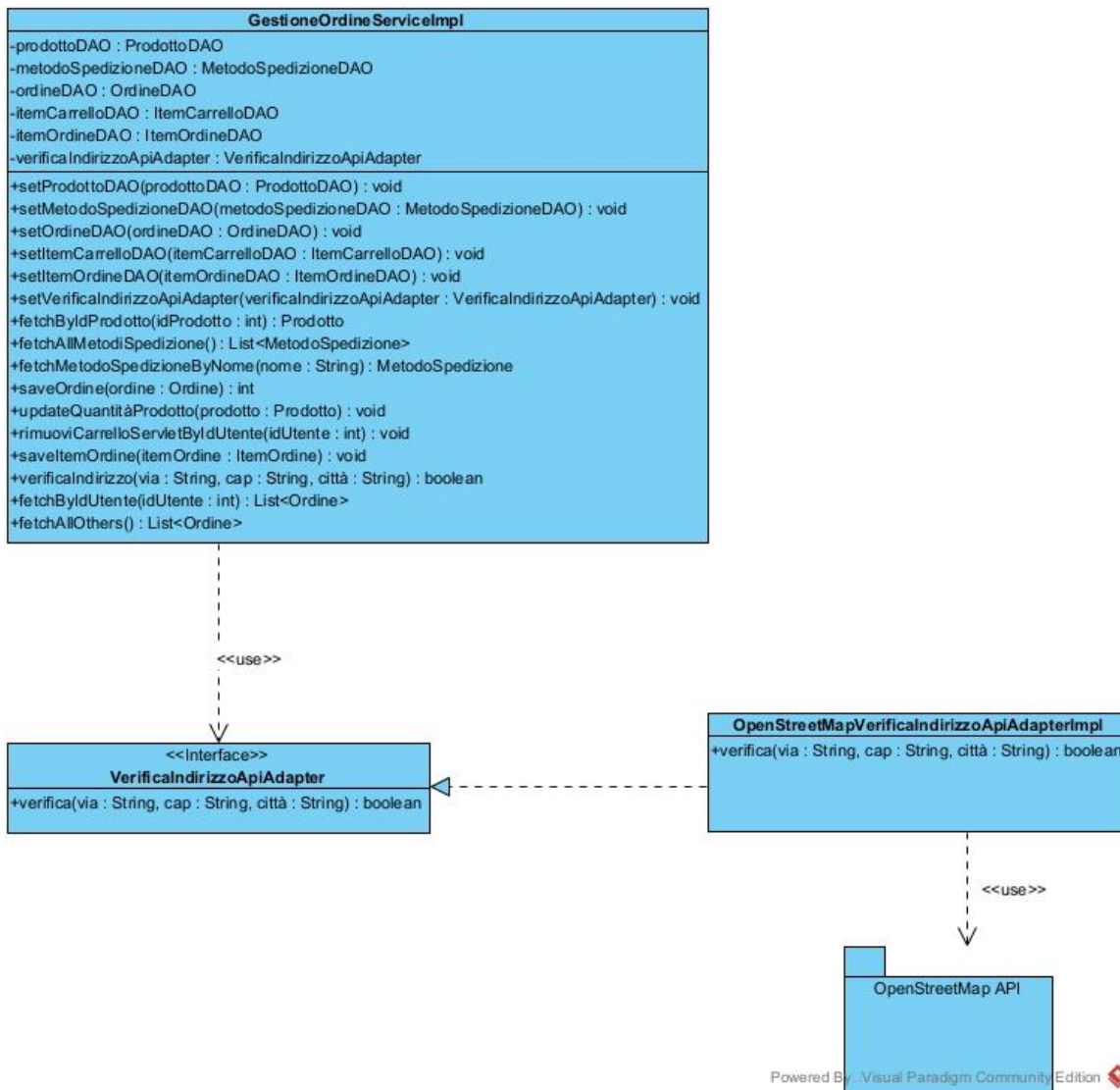
2. Design pattern

2.1 Pattern Adapter

L'**Adapter** è un design pattern di tipo strutturale che facilita il riutilizzo di componenti esistenti, provenienti da progetti precedenti o di tipo off-the-shelf (**COTS**). Questo pattern incapsula le componenti in modo da adattarle al nuovo sistema, riducendo al minimo l'impatto sul progetto e garantendo una maggiore compatibilità senza modificare direttamente il codice delle componenti originali. Questo viene realizzato mediante una classe "Adapter", il cui compito è trasformare i dati in un formato comprensibile per il sistema senza modificare il resto del sistema.

Digitronics, prima di effettuare l'ordine, verifica che l'indirizzo di spedizione (formato da via, CAP e città) esista effettivamente in Italia. Questo controllo viene effettuato per evitare l'inserimento di indirizzi non validi da parte dell'utente. Il controllo viene effettuato tramite un servizio di Geocoding esterno al sistema che viene offerto in modo interamente gratuito da **OpenStreetMap** tramite la API. È possibile inviare richieste alla API tramite una richiesta HTTP invia dati al server specificando metodo, URL e parametri (via, CAP e città). OpenStreetMap elabora la richiesta e restituisce una risposta JSON, una struttura dati leggibile e organizzata in formato chiave-valore da cui è possibile estrarre i dati per capire se l'indirizzo esiste effettivamente.

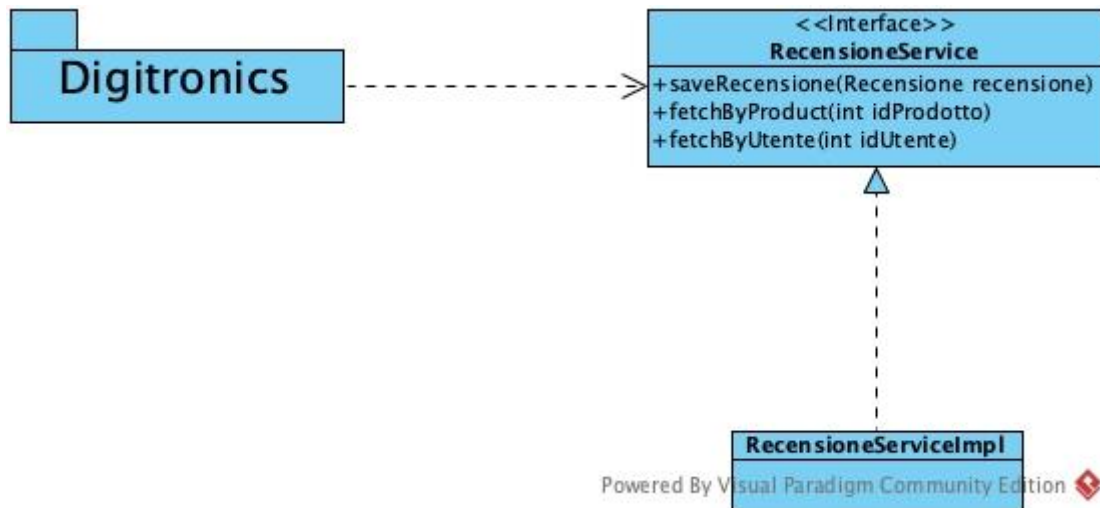
L'adapter **OpenStreetMapVerificaIndirizzoApiAdapterImpl** si occupa di inviare la richiesta alla API inserendo nella richiesta HTTP i dati relativi all'indirizzo di spedizione e di tradurre la risposta dell'API tramite adapter



2.2 Pattern Façade

Il **Façade** è un design pattern che, tramite l'implementazione di un'interfaccia semplificata, consente di accedere a sottosistemi più complessi. In questo modo, la complessità delle librerie, dei framework o dei set di classi utilizzati viene nascosta al sistema. Questo approccio garantisce un elevato **disaccoppiamento** e rende la piattaforma più **manutenibile** e **aggiornabile**, poiché sarà sufficiente modificare l'implementazione dei metodi dell'interfaccia per introdurre le modifiche necessarie.

Digitronics adotta il design pattern Facade per organizzare tutta la sua logica di business, semplificando l'interazione con essa. In particolare, Digitronics utilizza il Facade per ciascuno dei suoi sottosistemi, implementandolo tramite interfacce che consentono di accedere ai metodi interni in modo strutturato e controllato. Di seguito un esempio di Facade nel sistema Digitronics:



3. Glossario

- **Adapter Pattern:** È un pattern strutturale che può essere basato sia su classi che su oggetti il cui fine è fornire una soluzione astratta al problema dell'interoperabilità tra interfacce differenti.
- **Facade Pattern:** Un oggetto che permette, attraverso un'interfaccia più semplice, l'accesso a sottosistemi che espongono interfacce complesse e molto diverse tra loro.