

# Relazione progetto Programmazione ad Oggetti QtCafe

Leonardo Lago n°matricola 2034297

January 2023

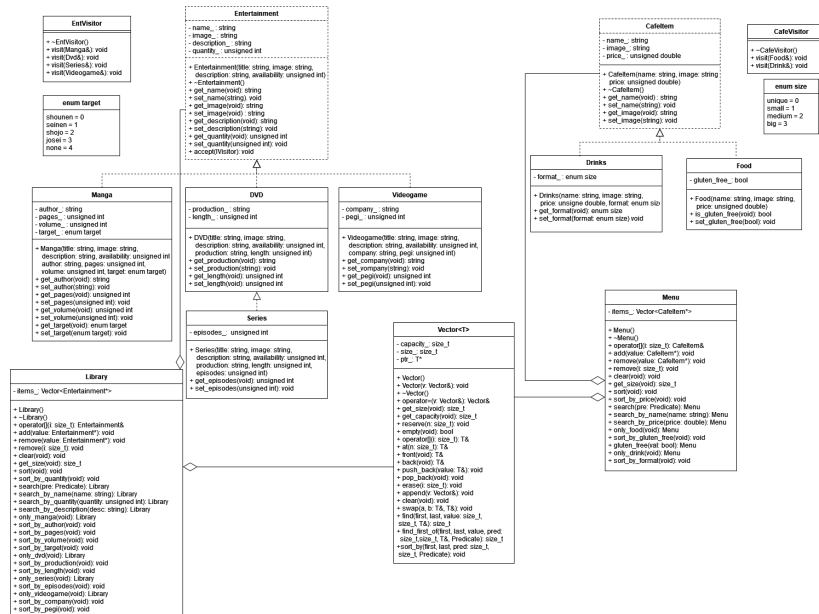
## 1 Introduzione

QtCafe è un software gestionale per un manga cafe che permette di aggiungere, eliminare, modificare e visualizzare i propri articoli. Questi sono di diversa tipologia, raggruppati in due principali categorie: gli articoli del cafe, che sono quindi cibi e bevande e quelli della libreria, nella quale rientrano manga, DVD, serie e videogiochi. Ho scelto di sviluppare questa tematica per il mio progetto poiché sono un amante della cultura nipponica e appassionato di manga. Inoltre, ho valutato che la varietà degli articoli da gestire potesse offrire uno spunto per l'utilizzo del polimorfismo in modo non banale.

## 2 Descrizione del modello

Il modello logico è principalmente strutturato in due parti molto simili tra loro: quella che si occupa di gestire gli articoli del cafe e quella degli articoli di intrattenimento. Entrambe contengono una classe astratta per la rappresentazione di un oggetto generico, le sottoclassi concrete e un contenitore che utilizza la classe template Vector. Quest'ultima rappresenta appunto un vettore dinamico di oggetti di tipo T ed è stata implementata avendo come riferimento la classe vector dello standard. Presenta quindi metodi per l'inserimento, cancellazione e accesso degli oggetti oltre che a funzionalità come find e sort\_by che attraverso l'uso di predicati permettono appunto la ricerca e l'ordinamento degli oggetti all'interno del vettore. Per permettere ciò sono state create numerose classi predicato che implementano l'operatore () e restituiscono un valore booleano a seconda che si sia trovato o meno un oggetto contenente il campo dati specificato nel caso dei predicati Search oppure se il primo valore è minore o uguale al secondo nel caso dei predicati Compare.

La classe astratta CafeItem, rappresenta un generico articolo del cafe contenente le informazioni comuni: nome, prezzo ed immagine rappresentativa. Per



ciascun campo sono implementati getter e setter. La classe concreta Food, rappresenta un cibo/pietanza che aggiunge il campo dati booleano senza glutine. La classe concreta Drink, a sua volta, rappresenta una bevanda caratterizzata dalla presenza del formato che può essere unico, piccolo, medio o grande. La classe concreta Menu rappresenta il contenitore per un generico prodotto del cafe, implementando quindi metodi per l'inserimento, cancellazione e accesso agli oggetti, oltre che alle precise funzioni di ricerca e ordinamento per i vari attributi degli oggetti. Sono di rilevanza anche i metodi `only_food` e `only_drink` che restituiscono una copia di Menu contenente, rispettivamente, solo oggetti Food e Drink.// Analogamente la classe astratta Entertainment, rappresenta un generico articolo di intrattenimento, contiene le informazioni riguardo nome, descrizione, immagine e quantità disponibile e implementa quindi getter e setter per ciascuno di essi. La classe concreta Manga, rappresenta un volume manga arricchendo l'oggetto generico di nome dell'autore, numero di pagine, numero del volume e del target che rientra in 5 categorie: nessuno, shounen (per ragazzi), seinen(per adulti maschi), shojo (per ragazze) e josei(per adulti femmine). La classe concreta Dvd, descrive un dvd con nome della produzione e lunghezza in minuti. Series costituisce, invece, il particolare dvd di una serie televisiva e mantiene quindi l'informazione sul numero di episodi contenuti nel disco. La classe Videogame, rappresenta un videogioco del quale sono specificate l'azienda produttrice e l'età minima sotto la quale non si può usufruire del contenuto(pegi). Infine la classe Library è il contenitore per un generico articolo di intrattenimento e implementa metodi analoghi alla classe Menu.

Le classi `CafeItem`, `Entertainment` e le loro rispettive sottoclassi, precedentemente illustrate, sono dei DTO (Data Transfer Object) e non espongono funzionalità di particolare interesse. Si è quindi deciso di utilizzare il design pattern Visitor e a tal scopo sono state create le classi `CafeVisitor` e `EntVisitor` con l'aggiunta nelle rispettive classi astratte del metodo virtuale `pure accept` per accettare il proprio visitor.

### 3 Polimorfismo

L'utilizzo del polimorfismo è riscontrabile nell'utilizzo del design pattern Visitor nella gerarchia `CafeItem` e `Entertainment`. Esso viene utilizzato all'interno delle classi `Menu` e `Library` nei metodi `only_type` per estrarre solo gli oggetti che sono sottotipo del tipo desiderato. I due visitor coinvolti sono `CafeAddVisitor` e `EntAddVisitor`. All'interno delle classi predicatori sono impiegati i visitor `CafeInfoVisitor` e `EntInfoVisitor` che permettono l'accesso ai campi dati specifici dell'oggetto che si sta considerando. Infine i visitor `CafeWidgetVisitor` e `EntWidgetVisitor`, usati all'interno del codice della GUI, sono i responsabili della creazione di un widget specifico a seconda del tipo dinamico dell'oggetto.

### 4 Persistenza dei dati

Per la persistenza dei dati è stato utilizzato il formato JSON. Il file contiene due vettori di oggetti, rispettivamente per gli articoli del cafe e della libreria. Gli oggetti sono semplici associazioni chiave-valore, e la serializzazione delle sottoclassi viene gestita aggiungendo un attributo "type". Un esempio della struttura è dato dal file `example.json`, fornito assieme al codice, che contiene un almeno un paio di oggetti per tipo, consentendo quindi di poter verificare le funzionalità dell'applicazione.

### 5 Funzionalità implementate

Le funzionalità implementate sono, per semplicità, suddivise in due categorie: funzionali ed estetiche. Le prime comprendono:

- gestione di diverse tipologie di articoli
- conversione e salvataggio in formato JSON
- funzionalità di ordinamento
- funzionalità di ricerca rispetto a varie opzioni

Le funzionalità grafiche:

- barra dei menù in alto
- scorciatoie da tastiera (mostrate anche nelle voci del menù)

- controllo della presenza di modifiche non salvate prima di uscire
- ogni tipologia di prodotto ha una propria visualizzazione
- possibilità di navigare attraverso diverse schermate tramite interfaccia
- utilizzo di icone nei pulsanti
- utilizzo di immagini e pulsanti nella visualizzazione degli oggetti
- utilizzo di colori e stili grafici
- effetti grafici come cambio del colore al passaggio del mouse

Le funzionalità elencate sono intese in aggiunta a quanto richiesto dalle specifiche del progetto.

## 6 Rendicontazione ore

Attività	Ore previste	Ore effettive
Studio e progettazione	8	9
Sviluppo del codice del modello	10	13
Studio del framework Qt	12	10
Sviluppo del codice della GUI	12	14
Test e debug	5	7
Stesura relazione	3	2
Totale	50	56

Il monte ore è stato leggermente superato in quanto lo sviluppo del modello e della GUI ha richiesto più tempo del previsto in particolare per la presenza di diversi bug e distrazioni.