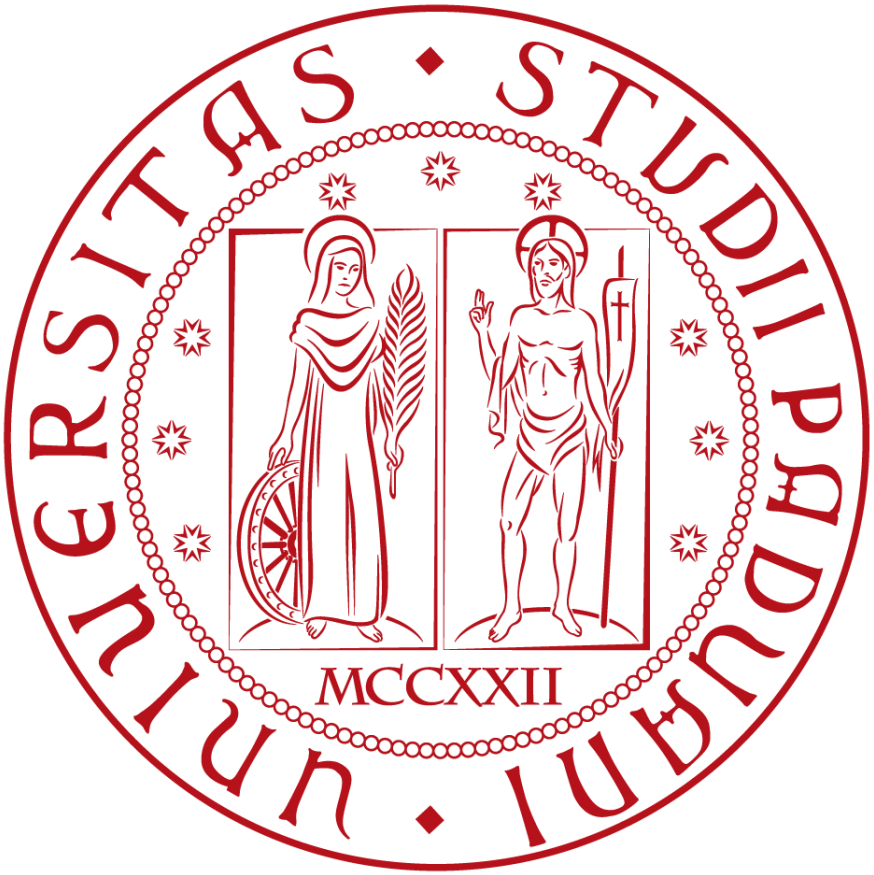
06 febbraio 2016

Università degli studi di Padova
Corso di Laurea in Informatica
AA. 2015/16

Relazione Progetto di Programmazione ad oggetti



Email: pietro.gabelli@studenti.unipd.it

Pietro Gabelli | 1008028

Indice

pietro.gabelli@studenti.unipd.it

Elenco delle tabelle

Elenco delle figure

pietro.gabelli@studenti.unipd.it

1 Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è di presentare in maniera chiara le principali scelte architetturali effettuate.

1.2 Scopo del progetto

Il progetto ha come scopo lo sviluppo di un'applicazione per la gestione di una biblioteca. Gli oggetti che compongono la biblioteca sono inseriti in un database che viene utilizzato attraverso una interfaccia grafica. Lo sviluppo è stato fatto utilizzando C++ e Qt.

1.3 Specifiche progettuali

Il progetto è destinato ad immagazzinare i titoli di una biblioteca domestica. Gli oggetti che si possono salvare sono:

- Libri;
- Film, siano questi VHS o DVD.
- CD;

Le funzionalità principali che si intende modellare sono:

- Aggiunta di nuovi elementi;
- Ricerca tra i titoli presenti;
- Cancellazione di titoli, tra quelli presenti.

I vincoli di progetto sono i seguenti:

1. Definizione ed utilizzo di una gerarchia G di tipi di altezza >= 1 e larghezza >= 1.
2. Definizione di un opportuno contenitore C, con relativi iteratori, che permetta inserimenti, rimozioni, modifiche.
3. Utilizzo del contenitore C per memorizzare oggetti polimorfi della gerarchia G.
4. Il front-end dell'applicazione deve essere una GUI sviluppata nel framework Qt.

Il progetto è stato sviluppato in un sistema XUbuntu; è stato utilizzato l'IDE Qt Creator nella versione 3.0.1, con le librerie alla versione 5.2.1.
Si è provato il progetto sui computer del Laboratorio Informatico Plesso Paolotti (LabP140 - labP036) dove compila ed esegue correttamente.
Il database che compone l'applicazione viene aperto e salvato usando il formato XML: entrambe le operazioni avvengono in maniera automatica grazie ad un path impostato di default.

Nello sviluppo è stata curata la separazione tra la parte logica e la parte grafica, senza utilizzare il design pattern MVC. Si e' preferita invece un'architettura Model/View, che rende possibile la separazione del modo in cui sono immagazzinati i dati e la loro presentazione all'utente.

pietro.gabelli@studenti.unipd.it

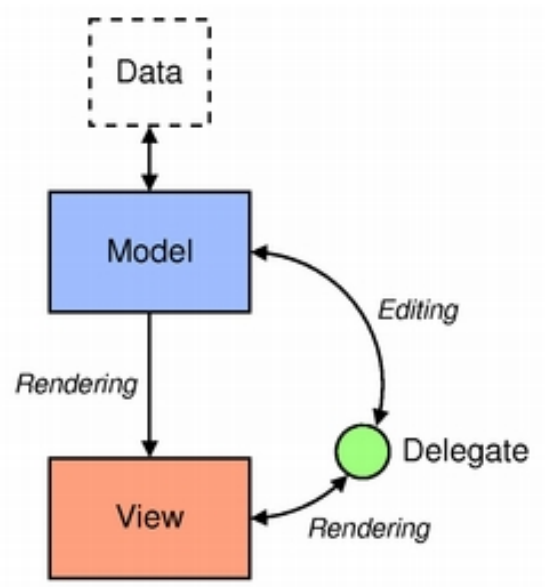


Figura 1: Model/View architecture

2 Classi logiche

La gerarchia sviluppata è composta da 5 classi:

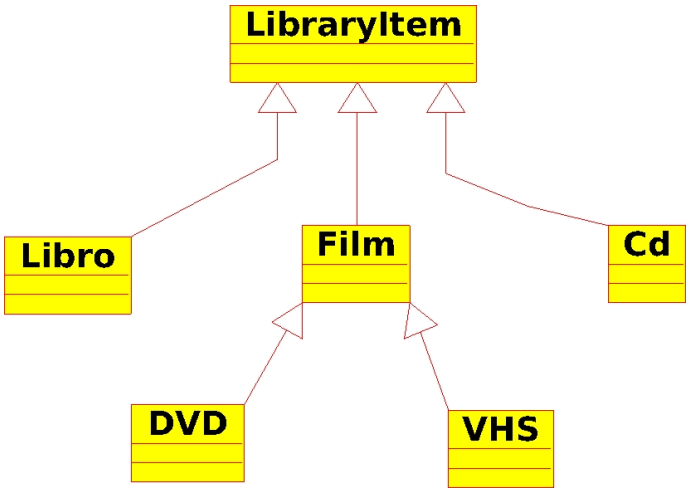


Figura 2: Gerarchia delle classi

La classe LibraryItem rappresenta un oggetto generico inserito nella libreria, per questo motivo ho scelto di renderla polimorfa ed astratta: non è possibile dichiarare oggetti di questa classe. All'interno di questa classe sono memorizzati gli attributi comuni a tutti gli oggetti della gerarchia: *titolo,genere*, entrambi di tipo QString.
Per questa classe e per le altre classi nella gerarchia verranno resi disponibili:

- Un costruttore di defalut ed un costruttore per i campi dati della classe;
- Un distruttore virtuale;
- Un metodo clone che restituisce il puntatore ad un copia all'oggetto su cui viene invocato;
- Un operatore di uguaglianza ed uno di disuguaglianza;

pietro.gabelli@studenti.unipd.it

- Metodi **get** che restituiscono una copia dei campi dati, per ogni campo dati richiesto.

Le classi **Libro, CD, Film** sono classi derivate direttamente dalla classe base **Utente**; sono tutte concrete, a differenza di **Film**, che e' anch'essa polimorfa ed astratta.
Le classi **DVD, VHS** sono classi concrete derivate da **Film**.
Per quanto riguarda gli attributi:

- La classe **Libro** ha:
  - Un *autore*, memorizzato tramite un QString;
  - Un *anno di uscita*, memorizzato tramite un intero;
  - Un *editore*, memorizzato tramite un QString.
- La classe **Film** ha:
  - Un *regista*, memorizzato tramite un QString;
  - Una *durata* in minuti, memorizzata tramite un intero;
  - Una *data d'uscita*, memorizzata tramite un QDate.
- La classe **CD** ha:
  - Un *artista*, memorizzato tramite un QString;
  - Un *anno di uscita*, memorizzato tramite un intero;
  - Un *numero di dischi*, memorizzato tramite un intero.
- Le classi **CD, DVD** sono implementazioni di **Film**; non contengono ulteriori campi dati. Vengono ridefiniti i metodi.

Per memorizzare gli oggetti della collezione e' stato creata una classe **Contenitore**. All'interno di questa classe.

3 Classi Grafiche

pietro.gabelli@studenti.unipd.it