

Analisi Architetturale: Coesione e Accoppiamento

Gruppo 12

15 dicembre 2025

1 Introduzione

Il presente documento analizza le metriche di qualità software del sistema *LibraryManager*, focalizzandosi sui principi di **Coesione** (il grado di responsabilità interna di una classe) e **Accoppiamento** (il grado di dipendenza tra classi diverse).

2 Analisi delle Componenti

2.1 Classi Entità: Utente / Libro

Coesione: Funzionale

Le classi rappresentano fedelmente le entità del dominio; ogni metodo è strettamente correlato alla gestione dei dati intrinseci dell'entità (es. getter, setter, stato attivo).

Accoppiamento: Per Dati (Data Coupling)

L'interazione con l'esterno avviene scambiando esclusivamente tipi primitivi o stringhe, garantendo la massima indipendenza e riusabilità.

2.2 Classe Entità: Prestito

Coesione: Funzionale

Tutti i metodi concorrono a definire l'entità *Prestito* e le relative operazioni di gestione temporale (date) e di stato.

Accoppiamento: Di Timbro (Stamp Coupling)

I metodi richiedono il passaggio di intere strutture dati composite (riferimenti agli oggetti *Libro* o *Utente*) per mantenere i vincoli di integrità referenziale, piuttosto che passare i singoli campi primitivi.

2.3 Modello Dati: ArchivioDati

Coesione: Comunicazionale

La classe raggruppa liste eterogenee (*Libri*, *Utenti*, *Prestiti*) che vengono utilizzate insieme durante le operazioni di I/O.

Accoppiamento: Di Timbro

La classe funge da contenitore per il passaggio dell'intero stato del sistema al gestore della persistenza.

2.4 Gestori Logica: GestoreLibro / Utente / Prestito

Coesione: Funzionale

Ogni metodo persegue un unico obiettivo specifico relativo alla manipolazione della lista in memoria, senza occuparsi della persistenza fisica o della visualizzazione.

Accoppiamento: Di Controllo (Control Coupling)

Alcuni metodi interni o ereditati possono accettare flag o oggetti Comparator che alterano la logica interna di esecuzione (es. ordinamento), creando una dipendenza dal chiamante che decide "come" eseguire l'operazione.

2.5 Gestore I/O: GestoreArchivio

Coesione: Funzionale

La classe ha l'unica responsabilità di serializzare e deserializzare i dati. Tutte le funzioni interne servono a questo unico scopo.

Accoppiamento: Di Timbro

Dipende dalla struttura della classe ArchivioDati e dalle classi del modello per poter effettuare la scrittura su disco.

2.6 Handler: *Handler (Utente/Libro/Prestito)

Coesione: Sequenziale

I metodi (es. onAdd, onEdit) orchestrano una serie di operazioni che devono essere eseguite in un ordine rigoroso: caricamento loader, inizializzazione controller, passaggio dati, visualizzazione stage.

Accoppiamento: Di Controllo

I metodi di ordinamento accettano stringhe di controllo (es. criteri di filtro) che influenzano il flusso algoritmico interno.

2.7 Controller Principale: PrincipaleController

Coesione: Temporale / Procedurale

Il metodo initialize raggruppa operazioni eterogenee legate solo dal momento in cui avvengono (avvio app). Il controller agisce come coordinatore procedurale tra gli Handler e i Manager.

Accoppiamento: Di Timbro

Scambia frequentemente intere collezioni (ObservableList) o oggetti complessi (ArchivioDati) con i sottosistemi.

2.8 Controller Form: Form*Controller

Coesione: Comunicazionale

I metodi operano sullo stesso insieme di dati (i campi del form e l'oggetto in modifica) per eseguire azioni distinte ma correlate (validazione, popolamento, salvataggio).

Accoppiamento: Di Timbro (Elevato in Prestito)

Ricevono le intere istanze degli oggetti da modificare. Nel caso di FormPrestitoController, l'accoppiamento è maggiore dovendo gestire la dipendenza simultanea da Libro, Utente e Prestito.

3 Tabella Riassuntiva

Di seguito un riepilogo delle metriche per una consultazione rapida.

Classe/Componente	Coesione	Accoppiamento
Modello (Utente, Libro)	Funzionale	Dati (Basso - Ottimo)
Modello (Prestito)	Funzionale	Timbro
ArchivioDati	Comunicazionale	Timbro
Gestori (Logic Managers)	Funzionale	Controllo
GestoreArchivio	Funzionale	Timbro
Handlers (AreaHandler)	Sequenziale	Controllo
PrincipaleController	Temporale	Timbro
Form Controllers	Comunicazionale	Timbro
LibraryManagerApp	Funzionale	Timbro (JavaFX Stage)

Tabella 1: Sintesi dell'analisi architetturale