

Esercizio POO Classi Astratte e Interfacce

Roberto Borelli

09 Marzo 2023

Testo

Una Biblioteca contiene al suo interno vari Materiali Bibliotecari. Ogni materiale ha un nome e un numero di copie disponibili. I materiali sono di tipo DVD, Libro o Rivista. I libri e le riviste hanno un numero di pagine. I dvd hanno una durata (in minuti). I libri e i dvd possono essere prestati. Le riviste e i libri possono essere comprati.

- La classe `MaterialeBibliotecario` deve offrire un metodo `ottieniValutazione` che restituisce un numero con la virgola. La valutazione di un dvd si ottiene dividendo la sua durata per 3. La valutazione di un libro si ottiene moltiplicando il numero di pagine per 0.1. La valutazione di una rivista si ottiene moltiplicando il numero di pagine per 0.01.
- Si implementi nella classe `MaterialeBibliotecario` un metodo `verificaDisponibilita` che restituisce un booleano (vero se l'oggetto è nella biblioteca). Un oggetto è disponibile se il numero di copie presenti in biblioteca è positivo.
- Per gli oggetti comprabili si implementi il metodo `compra`
- Per gli oggetti prestabili si implementino i metodi `presta` e `restituisci`
- Si implenti una classe `Biblioteca` che abbia un metodo `ottieniOggettiDisponibili` che restituisce un `ArrayList` di Materiali Bibliotecari (quelli disponibili attualmente). Si aggiungano inoltre metodi per aggiungere e rimuovere materiali.
- Si crei una classe `Main` per testare tutte le funzionalità descritte.

Richieste

Per l'esercizio precedente, si disegni uno schema UML che rispetti tutte le convenzioni. Tutte le frecce devono essere del tipo giusto. Si abbia cura di mettere in corsivo i metodi astratti e le classi astratte qualora fossero necessarie. Si implenti il codice dell'esercizio in Java. Se si nota del codice duplicato, è il caso di ripensare la gerarchia delle classi. Si riparta quindi dall'uml e si riesegua l'implementazione. L'uml non può essere fatto su miro.com ma può essere fatto su carta o su draw.io. Il codice java deve come minimo compilare e non ci devono essere differenze rispetto a quanto mostrato nell'uml. Per ogni classe si implementino i costruttori e i metodi `toString`. Vietato utilizzare le implementazioni di default che offre l'editor di codice. Vietato usare chatGpt. Ammesso ispirarsi ad esercizi già eseguiti insieme.