

1 – Considerare la seguente descrizione del processo di prestito dei libri di una biblioteca, per il quale si vuole progettare un software:

Un testo può essere composto da uno o più tomi. Ogni tomo ha una collocazione, mentre il testo ha un titolo e un anno di pubblicazione. Un testo può essere un libro, nel qual caso il testo ha uno o più autori, oppure una collezione di riviste, nel qual caso è ciascuno dei suoi tomi ad avere uno o più autori. Ogni autore ha un nome ed un cognome.

Un utente della libreria ha un nome, un cognome ed un numero di matricola. Un utente può essere o non essere ammesso al prestito. Se è ammesso al prestito può prendere in prestito non più di un libro alla volta. Il prestito può durare da 1 a 3 giorni. Tutti gli utenti possono prendere un numero illimitato di testi in consultazione. Le collezioni di riviste non possono essere prese in prestito. Un testo non può essere preso in prestito o in consultazione da più di un utente per volta.

Rappresentare il processo di prestito libri con un class diagram UML.

2 – Modificare il class diagram considerando la seguente aggiunta:

Un impiegato della libreria può registrare nuovi testi. Gli impiegati possono anche essere utenti della libreria, e quindi prendere testi in prestito o in consultazione. Un impiegato può prendere in prestito fino a tre libri contemporaneamente, ma per il resto si applicano agli impiegati tutte le regole degli utenti normali sul prestito e sulla consultazione.

3 – Le operazioni di prestito e consultazione libri sono così descritte:

Un utente chiede ad un impiegato il libro (uno solo) che vuole prendere in prestito o consultazione. Questi verifica che l'utente non stia chiedendo in prestito più libri di quanto gli sia consentito, che il libro non sia già in prestito o in consultazione, e se entrambi i controlli sono positivi gli assegna il libro in prestito.

Descrivere un corretto svolgimento della operazione di prestito (NON di quella di consultazione) con un sequence diagram UML. Se lo ritenete necessario, potete aggiungere classi, proprietà e metodi senza correggere i class diagram dei punti 1 e 2.

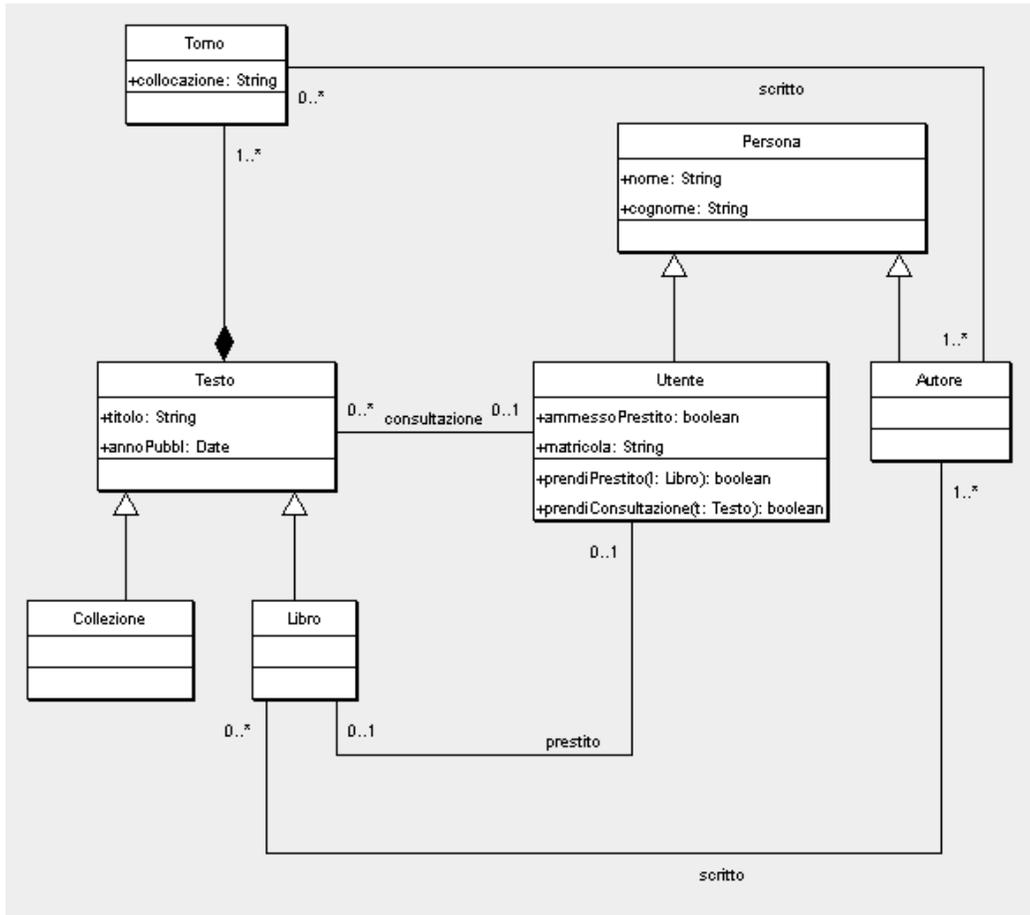
4 – Rappresentare con un sequence diagram UML l'operazione di registrazione di un nuovo libro così descritta:

Un impiegato può registrare presso la segreteria un nuovo testo registrando ogni tomo sotto una nuova collocazione.

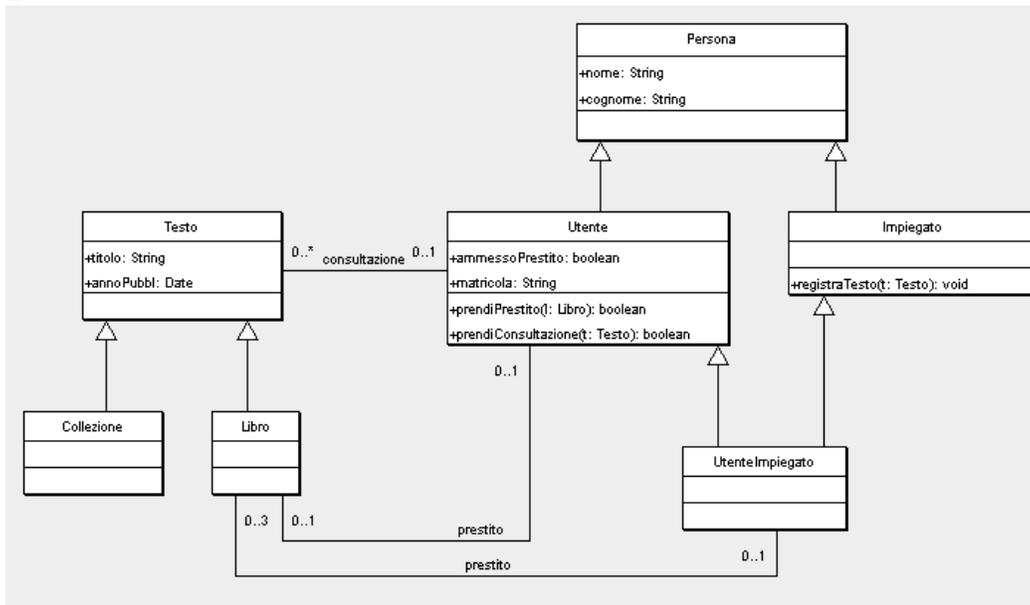
Anche qui, se lo ritenete necessario, potete arricchire le classi con nuove proprietà e metodi, oppure aggiungete nuove classi, senza necessità di correggere i class diagram dei punti 1 e 2.

Soluzione:

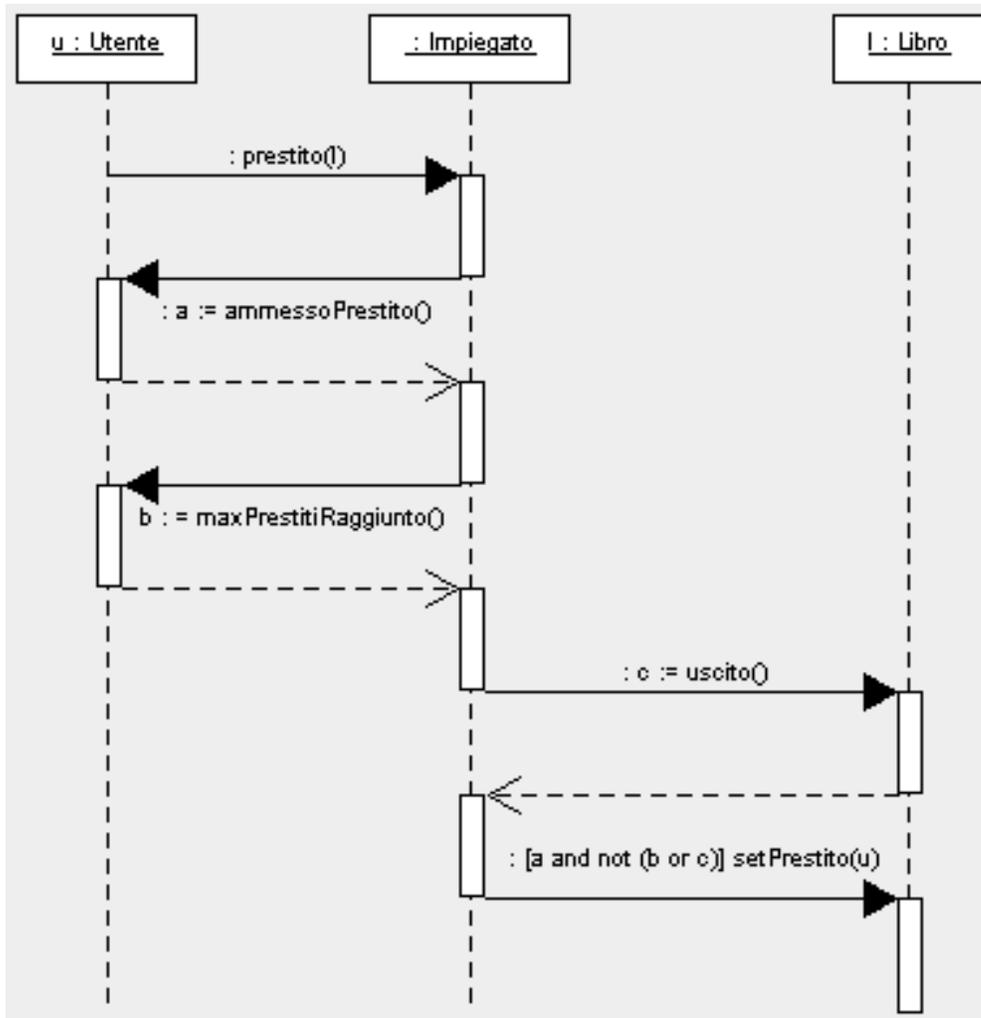
1 -



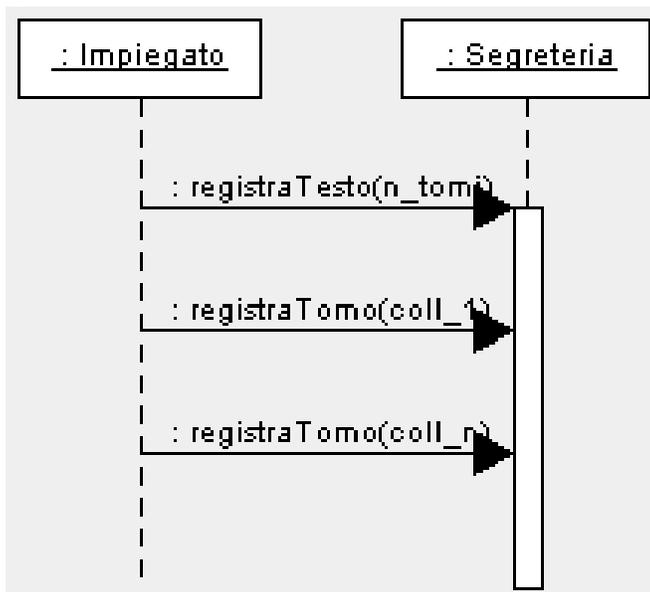
2 -



3-



4-



1 – Considerare la seguente descrizione:

Ogni programmatore è gestito da un manager, il suo superiore. Se il programmatore è in un progetto, può avere anche il project manager come suo superiore. I programmatori possono emettere note spese, le quali vanno approvate dal project manager e passate all'amministrazione. Gli analisti di un task del progetto lavorano ad un documento di analisi insieme al cliente. L'amministrazione distribuisce gli stipendi a programmatori e manager, paga le fatture ai fornitori ed emette fatture ai clienti.

Individuate attori ed use case, definendo un use case diagram per la situazione descritta.

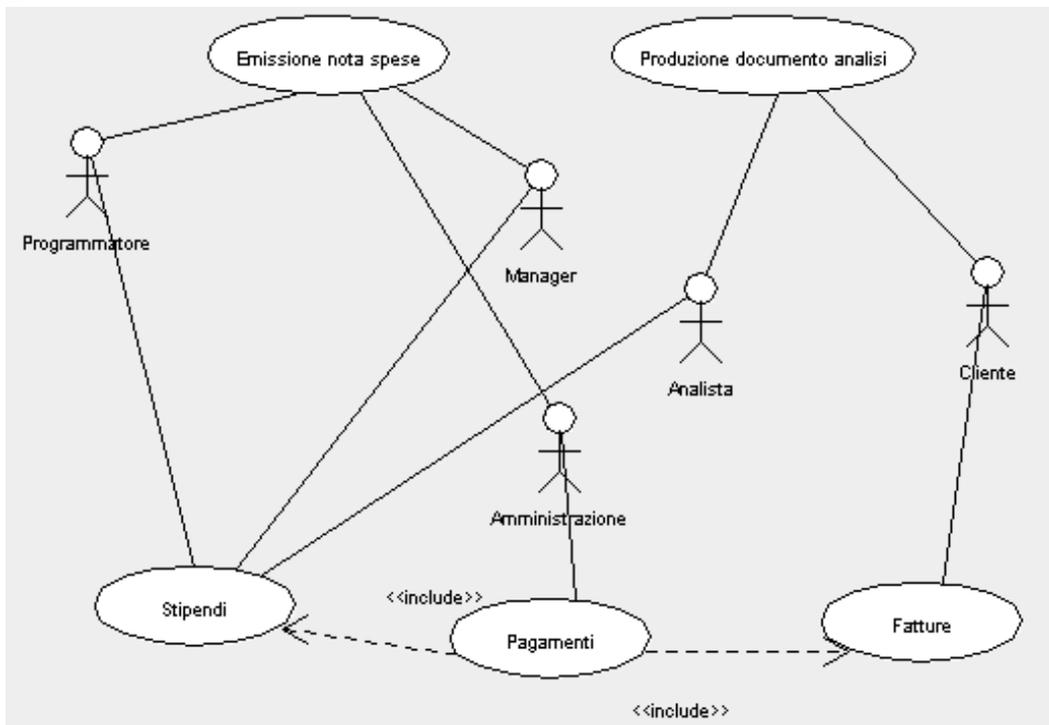
2 – Considerate la seguente descrizione del processo di ordinazione materiali:

L'amministrazione emette un ordine in base alle note di richiesta materiali, con la relativa quantità, che le arrivano. Ogni ordine è composto di un insieme di righe, ogni riga corrisponde al materiale richiesto in ciascuna nota. L'ordine viene inviato ad un fornitore se questo può soddisfare la richiesta di tutti i materiali nell'ordine.

Scrivete un sequence diagram che descriva la generazione di un ordine a partire da due note di richiesta materiale. Si supponga che vi siano due fornitori, uno che non può soddisfare la richiesta e uno che può.

Soluzione:

1 –



2 – ved. prossima pagina

