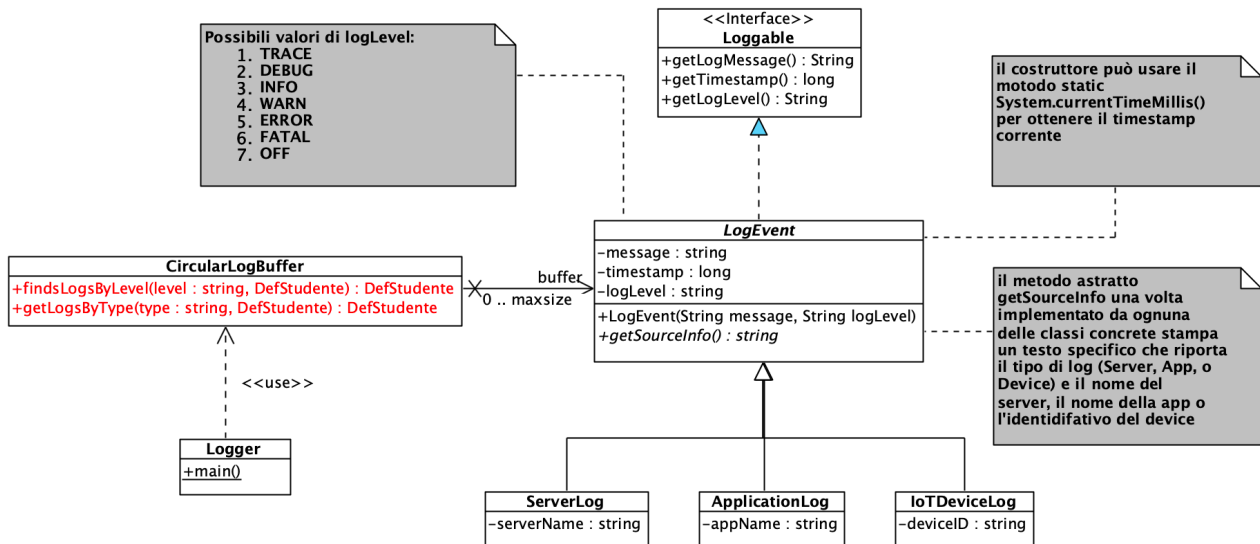


Corso di Programmazione A.A. 2023/2024

Esame del 22 Ottobre 2024

Prof. Domenico Amalfitano

Si vuole implementare un'applicazione software per la gestione di un logger di eventi. Il logger utilizza una coda circolare `CircularLogBuffer` che memorizza oggetti di tipo `LogEvent` in un buffer di al più `maxsize` oggetti. Lo studente, per fini di testing, può scegliere liberamente il valore numerico della variabile `maxsize`. Il software è modellato dal diagramma UML riportato di seguito.



Requisiti del software.

1. L'interfaccia `Loggable` fornisce i metodi di `get` delle caratteristiche degli attributi degli oggetti di tipo `LogEvent`.
2. Lo studente analizzi le tre note UML del progetto che specificano ulteriormente le caratteristiche delle classi.
3. Lo studente implementi nella classe `CircularLogBuffer` tutti i metodi necessari per gestire una coda circolare memorizzata in un array.
 - a. In particolare, se si prova ad inserire un evento di log in una coda già piena allora questo nuovo evento va a sostituire uno (nel caso in cui ce ne fossero più di uno) dei log più vecchi, ovvero a TimeStamp minimo.
 - b. Se si prova ad accedere in lettura ad una coda vuota viene generata un'eccezione.
4. Nel diagramma UML mancano alcuni costruttori e i metodi `toString` della classe astratta e delle tre classi concrete. Lo studente implementi i metodi che ritiene necessari al completo svolgimento della prova.
5. Il metodo `findsLogsByLevel` ha tra i parametri una stringa relativa a un valore di livello di log cercato. Il metodo restituisce, per ogni tipologia di eventi di log, l'insieme di tutti i log il cui livello è uguale al valore cercato.
6. Il metodo `getLogsByType` restituisce un elenco di tutti gli eventi di log di un dato tipo. Il tipo di log cercato è passato come parametro di tipo stringa del metodo.
7. Lo studente definisca i valori di ritorno, nonché ulteriori parametri dei metodi `findsLogsByLevel` e `getLogsByType`.
8. È ammesso l'uso dei container Java a supporto dell'implementazione dei metodi `findsLogsByLevel` e `getLogsByType`.
9. Si implementi la classe utente, `Logger`, per testare le funzionalità del sistema tenendo conto anche di diversi scenari d'uso.