Scrivere un insieme di Enterprise JavaBeans e client che rappresentano i giocatori di League, persistente sul DB, che contengano le informazioni sul id, livello, regione (EU, EUNE, NA, KR), lega (unranked, argento, oro, platino, diamante, challenger), essenza blu e riot points.

a) Tramite JPA, si deve gestire l'archivio persistete dei giocatori su DB (EsameDB), dove la chiave primaria è la id.
   1. Devono essere previste delle query per id, per regione, per lega e una query che restituisce tutti i giocatori di League of Legends.
   2. Deve essere previsto un bean Singleton che inizializzi l'archivio.
   3. Scrivere un client basato su invocazione di un bean stateless che preveda la stampa di:
      a. Tutti i giocatori.
      b. Tutti i giocatori di una determinata regione, argomento passato come parametro dall'utente (stdin)
      c. Tutti i giocatori con livello >= x, x passato dall'utente (stdin).
b) Scrivere un interceptor che provvede a mantenere una statistica del numero di volte che ogni metodo è chiamato, a esclusione del metodo per la stampa di tutti i giocatori.
c) Scrivere un client basato su messaggi che invia un messaggio con id e livello, aggiornando le informazioni sul database in maniera conseguente se e solo se il nuovo livello è > del precedente. Tramite evento viene stampato sul server un messaggio di successo o di fallimento dell'aggiornamento.
d) Utilizzare la tecnica delle Callback Annotations per verificare, ogni volta che un giocatore viene inserito, che si tratti di un inserimento valido (livello >= 1, lega unranked se livello < 30, essenza blu >= 0, riot points >= 0). In caso l'inserimento non è valido, viene lanciata una eccezione.
e) Rendendo i metodi del bean invocabili come Web Services, scrivere un client basato su invocazione WebServices che modifichi i riot points di un giocatore (chiesto all'utente). Quando viene modificato questo valore, allora tramite un evento viene stampato sulla console del server un messaggio di avviso.

ID	Livello	Regione	Lega	EssenzaBlu	RiotPoints
1	10	NA	unranked	2000	20
2	232	KR	challenger	120000	25000
3	80	KR	platino	34000	1600

NOTE:

1) Il DataSource deve chiamarsi: jdbc/EsameDS.
2) PersistentUnit e DB devono chiamarsi rispettivamente EsamePU ed EsameDB.
3) Il DatabasePopulator deve prevedere la database definition.
4) La ConnectionFactory deve chiamarsi: jms/javaee7/ConnectionFactory.
5) Il Topic deve chiamarsi: jms/javaee7/Topic.