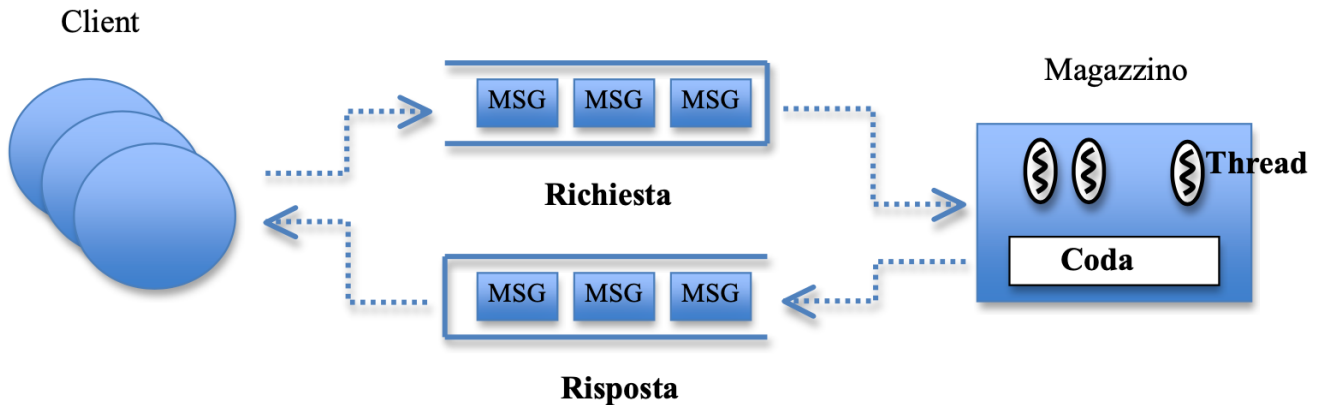


Università degli Studi di Napoli Federico II

Advanced Computer Programming

Esercitazione JMS 01

Si realizzi in linguaggio JAVA un server **multithread** basato su **code di messaggi JMS**. Il sistema implementa un deposito di articoli elettronici, e si compone di 2 entità mostrate in figura.



Client. Ciascun client può inviare N messaggi sulla coda **Richiesta**. Ogni messaggio contiene 2 informazioni: (i) *tipo di richiesta* (deposita o preleva) e (ii) *id_articolo* (rappresentato da un intero). Il messaggio deve essere di tipo `MapMessage` e deve contenere le coppie (“operazione”, “deposita”) e (“valore”, *id_articolo*), nel caso di deposito (e.g., [{“operazione” : “deposita”}, {“valore”: 10}]) e la coppia (“operazione”, “preleva”) nel caso di richiesta di tipo preleva. La ricezione dei messaggi da parte del client deve essere asincrona.

Magazzino gestisce una coda a gestione circolare. La dimensione della coda è pari a 10. All’arrivo di un nuovo messaggio client, il Magazzino crea un nuovo thread, che estrae le informazioni dal messaggio per ottenere il *tipo richiesta* ed eventualmente *id_articolo*. Se il messaggio è di tipo “deposita”, *id_articolo* è inserito nella coda. Nel caso di un messaggio “preleva”, viene prelevato l’*id_articolo* in testa alla coda e restituito al client tramite la coda **Risposta**. Quando la coda è piena, il thread che effettua un *deposito* è messo in attesa; analogamente, se la coda è vuota, il thread che effettua una *preleva* è messo in attesa. E’ possibile gestire l’accesso alla coda secondo i meccanismi di sincronizzazione ritenuti più opportuni. La ricezione dei messaggi da parte del Magazzino deve essere asincrona.

Si creino 3 Client. Ogni Client genera 10 messaggi, 5 di tipo deposito, con *id_articolo* generato in maniera casuale, e 5 di tipo preleva. Assumere che i Client possano rimanere in attesa indefinita di consumazione.