

Prova di Programmazione Distribuita - 20/03/2024

Matricola 0512113808 Nome GIANNAPPO Cognome COGNOME RIVIELLO

Preferenze per orale (scrivere i soli giorni in cui non si è disponibili):

a) Scrivere un insieme di server Enterprise JavaBeans e un client che rappresentino un elenco degli immobili in vendita di un'agenzia immobiliare. Per ogni immobile, una entità dovrà raccogliere le informazioni su ID (int), indirizzo (stringa), valore immobile (int), data ultima modifica allo stato (data), data di caricamento dati (data), stato (enum o stringa contenente "in vendita", "in trattativa" oppure "venduto").

Tramite JPA, si deve gestire la persistenza degli immobili su DB (EsameDB), dove la chiave primaria è la ID

1. devono essere previste una query per INDIRIZZO e cognome, ed una query che restituisca tutte le entry del database;
2. deve essere previsto un bean Singleton che inizializzi l'archivio;
3. scrivere un client basato su invocazione di un bean stateless che prevede la stampa di: (a) Tutti gli immobili in affitto; (b) Tutte gli immobili in vendita da non più di due mesi.

Dati di esempio del DB:

ID	Indirizzo	Valore	Data modifica	Stato
1	Via Garibaldi, Salerno	600 000	01 03 2024	in vendita
2	Via Posidonia, Salerno	400 000	01 11 2024	n vendita
3	Via dei Greci, Salerno	200 000	01 11 2023	venduta
4	Via Porto, Salerno	300 000	01 11 2023	in trattativa

b) Scrivere un client basato su messaggi e la corrispondente parte lato server che invii un messaggio per inserire un nuovo immobile o aggiornare lo stato di uno esistente. Quando viene effettuato questo inserimento, viene stampato sulla console del server un messaggio di avviso "DATA_INSERTIMENTO: Inserimento immobile in INDIRIZZO di VALORE" (nota: sostituite le parole in maiuscolo con il valore corrispondente). Se viene inserita un immobile già in archivio, il cui stato cambia a "venduto", tramite un evento verrà stampato il messaggio "Immobile venduto!".

c) Rendendo i metodi del bean invocabili come Web services, scrivere un client che stampi tutti gli immobili venduti.

d) Descrivere brevemente i seguenti concetti: (1) Architettura RMI, (2) Modelli di delivery dei messaggi in JMS, (3) SOAP, (4) Ciclo di vita di un Entity.

Note: (1) Il DataSource deve chiamarsi: jdbc/EsameDS; (2) PersistentUnit e DB devono chiamarsi rispettivamente EsamePU ed EsameDB; (3) Il DatabasePopular deve prevedere la database definition; (4) La ConnectionFactory deve chiamarsi: jms/javaee7/ConnectionFactory; (5) Il topic deve chiamarsi: jms/javaee7/Topic. Si richiede di scrivere il codice tenendo presente le fondamentali richieste di ordine e buona strutturazione delle classi secondo i principi della programmazione e oggetti. Inoltre è necessario scrivere il codice seguendo le regole e le convenzioni di scrittura di programmi Java, non ultima la necessità di indentare correttamente e di commentare il codice. Procedere nella scrittura secondo il seguente ordine: (1) Entità, Interfacce, EJB, Client EJB; (2) JMS: MDB e Client; (3) Web Services: Server, Client.