Programmazione Distribuita – Prova scritta del 14/01/2022

Cognome e Nome (stampatello): _____ ***Matricola:*** _____

PREFERENZE PER ORALE (giorni esclusi!):

Scrivere in stampatello Cognome e Nome su ogni foglio	I progetti funzionanti dal nome "COGNOMENOMEXXX" in un file COGNOMENOMEXXX.zip con file Readme.txt che illustri le modifiche) vanno sottomessi su Piattaforma e-learning, in un unico archivio COGNOMENOME_EE.zip entro le **ore 21:00 del 17/01/2022** Gli orali inizieranno orientativamente il **20/01/2022** ma attendete conferme sulla piattaforma di dipartimento e sul Teams della vostra classe.

Riservato alla Commissione per la Correzione

a	b	c	d	TOT	Commenti
12	5	5	8		

TRACCIA

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio dati relativo alle partite di un Torneo di Scacchi. Per ogni partita disputata durante il torneo, l'archivio mantiene le informazioni su ID (int), Tipo di Partita (Classic, Rapid, Blitz), nome del Giocatore 1 (G1), Rating giocatore 1 (valore intero), nome del Giocatore 2 (G2), Rating giocatore 2 (valore intero), Mosse (una stringa con le mosse), Risultato (G1, G2, Patta oppure In corso), Partita conclusa (boolean).

a) Tramite JPA, si deve gestire l'archivio persistente delle partite di scacchi su DB (EsameDB), dove la chiave primaria è la ID.

1. Devono essere previste query per Tipo di partita, Giocatori, Risultato, Mosse ed una query che restituisce 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. Tutte le partite finite dove entrambi i giocatori hanno rating maggiore di 2300.
    b. Tutte le partite finite in pareggio.

[12 punti]

b) Scrivere un client basato su messaggi e la corrispondente parte lato server che invia un messaggio per aggiornare le mosse della partita e per aggiornare il risultato (separatamente). Quando viene effettuato questo aggiornamento (sia delle mosse che del risultato), tramite un evento viene stampato sulla console del server un messaggio di avviso "Aggiornamento effettuato". Se la partita si conclude con un vincitore ed il suo rating è maggiore di 2800 allora verrà stampato sulla console (tramite eventi) anche il messaggio "NOME_GIOCATORE is a superplayer", alternativamente in caso di partita patta, verrà stampato sulla console il messaggio "Partita conclusa con un pareggio".

[5 punti]

c) Rendendo i metodi del bean invocabili come Web services, scrivere un client che stampi la lista di tutte le partite che sono state giocate (cioè che sono terminate).

[5 punti]

d) Descrivere brevemente i concetti di Object-Relational Mapping, Entity Manager, Persistence Unit e il ciclo di vita delle Entità.

[8 punti]

Dati di esempio del DB

ID	Tipo Partita	Giocatore 1	Giocatore 2	Rating1	Rating 2	Mosse	Risultato	Partita conclusa
1	Classic	Pap	Furman	2431	2114	e4 c5 Nf3 Nc6 Nc3 Nf6 e5 Nd5 Bc4 Nb6 Bb3 d5	G1	true
2	Rapid	Caruana	Nepomniachtchi	2770	2798	e4 e5 Nf3 Nf6 Nc3 Nc6 Bb5 Nd4 Nxd4 exd4 e5 dxc3	Patta	true
3	Classic	Carlsen	Maghsoodloo	2892	2655	e4 e6 d4 d5 e5 c5 c3 Qb6 Nf3	In corso	false
4	Blitz	Nepomniachtchi	Nakamura	2798	2836	e4 e5 Nf3 Nc6 Bb5 Nf6 O-O Nxe4 Re1 Nd6 Nxe5 Be7 Bf1 Nf5 Nf3 d5 d4 O-O Nc3 Bb4 h3 Nd6 a3 Bxc3 bxc3 Re8 Rxe8+	Patta	true

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

ATTENZIONE: si richiede di:

- Indicare a quali parti della prova (a, b, c, d) si risponde barrando le caselle apposite.
- 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:
    - Entità, Interfacce, EJB, Client EJB
    - JMS: MDB e Client
    - Web Services: Client

Cognome e Nome (stampatello): _____ ***Matricola:*** _____

PREFERENZE PER ORALE (giorni esclusi!):

PER L'INVIO DEL PROGETTO, LA CORREZIONE E GLI ORALI

- Il progetti funzionanti, entro le ore **21:00 del 17/01/2022**, vanno sottomessi come da istruzioni fornite sulla Piattaforma Teams, dove verranno anche fornite le ammissioni
- L'invio dei progetti entro la data stabilita è da considerarsi come richiesta di correzione (nel senso che se non viene inviato, il compito non viene corretto!).
- Regole per la consegna
    - I progetti da consegnare (progetti NetBeans) si devono chiamare "COGNOMENOMEXXX" nei rispettivi files COGNOMENOMEXXX.zip (devono essere esportati).
    - L'archivio che li contiene TUTTI insieme deve chiamarsi COGNOMENOME_EE.zip
    - Deve esserci un UNICO file **readme.txt** (non **Readme**, non **leggimi**, …) all'interno di COGNOMENOME_EE.zip che illustri le modifiche che si sono rese necessarie (per ogni file di ogni progetto) rispetto a quanto consegnato, secondo lo schema che viene pubblicato sulla piattaforma.
    - Le modifiche devono essere commentate nel codice dei progetti.
- Il calendario degli orali verrà comunicato sulla canale di "Annunci e Avvisi" dell'Anno Accademico in corso. Gli orali si terranno a partire dalle date indicate sulla piattaforma Teams, il link vi verrà comunicato con le ammissioni. Si specifica che è possibile che i risultati siano disponibili solo poco prima di questa data, e che gli orali (salvo diversa comunicazione su piattaforma) inizieranno comunque in questa data.

TERMINE PER L'INVIO DEL PROGETTO: ore 15:00 del 21/01/2022

AMMISSIONI ed ORALI pubblicati sulla piattaforma/Teams