

Cognome e Nome (stampatello): _____ Matricola: _____

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 9:00 del 26/02/2022
Gli orali inizieranno orientativamente la settimana

Riservato alla Commissione per la Correzione

a	b	c	d	TOT	Commenti
10	7	5	8		

TRACCIA

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio dati relativo alla **prossima** elezione del Presidente della Repubblica, per la gestione dei voti ricevuti dai candidati. Per ogni candidato, l'archivio contiene le informazioni su ID (int), Cognome, Nome, Carica (Senatore, Deputato, altro), Genere (Uomo, Donna), GiaElettoPresidente (booleano), Voti ricevuti (intero).

- a) Tramite JPA, si deve gestire l'archivio persistente dei dati delle votazioni su DB (EsameDB), dove la chiave primaria è la ID.
- Devono essere previste query per Cognome, per Genere, per Carica ed una query che restituisce tutti candidati del database
 - Deve essere previsto un bean Singleton che inizializzi l'archivio
 - Scrivere un client basato su invocazione di un bean stateless che prevede la stampa di:
 - Tutti i candidati Donna che hanno ricevuto più di un certo numero (passato in input) di voti
 - Tutti i candidati che sono in carica come Deputato [10 punti].

b) Scrivere un client basato su messaggi e la corrispondente parte lato server che invia un messaggio che serve per aggiornare i voti. I voti passati rappresentano l'aggiornamento al voto presente nell'archivio, e vanno quindi sommati a quelli precedentemente memorizzati. Si deve controllare se i voti si riferiscono ad un candidato Donna e in tal caso si stampa, tramite un evento, sulla console del server un messaggio di avviso "Finalmente! Voti per un candidato donna!" [7 punti]. Si deve prevedere un interceptor che conti le invocazioni del metodo di ricerca per Genere, stampando su console il numero di invocazioni finora rilevato.

c) Rendendo i metodi del bean invocabili come Web services, scrivere un client che stampi tutti i voti per ogni Carica (la somma dei voti ricevuti dai candidati con Carica Senatore, Deputato o Altro). Si deve tenere presente la efficienza della soluzione dal punto di vista della banda di rete utilizzata. [5 punti].

d) Descrivere per quanto riguarda JMS, (1) la architettura e le componenti coinvolte, (2) il meccanismo di filtering (3) priorità di messaggi e (4) la durable subscription [8 punti]

Dati di esempio del DB

ID	Cognome	Nome	Carica	Genere	GiaElettoPresidente	Voti ricevuti
1	Bonino	Emma	Deputato	Donna	False	125
2	Alberti Casellati	Maria Elisabetta	Senatore	Donna	False	132
3	Mattarella	Sergio	Altro	Uomo	True	82

NOTE:

- Il DataSource deve chiamarsi: jdbc/EsameDS
- PersistentUnit e DB devono chiamarsi rispettivamente EsamePU ed EsameDB
- Il DatabasePopular deve prevedere la database definition
- La ConnectionFactory deve chiamarsi: jms/javaee7/ConnectionFactory
- Il topic deve chiamarsi: jms/javaee7/Topic

ATTENZIONE: si richiede di:

- Indicare a quali parti della prova (a, b, c) 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: _____

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

- Il progetti funzionanti, entro le ore **09:00 del 26/02/2022**, vanno sottomessi tramite la piattaforma Teams, seguendo le iscrizioni e il link, 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 sul Teams del corso 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 09 del 26/02/2021

AMMISSIONI ed ORALI pubblicati sulla piattaforma/Teams