

Programmazione Distribuita – Prova scritta del 11/01/2020**Cognome e Nome (stampatello):** _____ **Matricola:** _____**PREFERENZE PER ORALE (giorni esclusi!):**

Scrivere in stampatello Cognome e Nome su ogni foglio	<p>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 14/1/2020</p> <p>Gli orali inizieranno il 18/1/2021 e termineranno nella settimana.</p>
---	---

Riservato alla Commissione per la Correzione

a	b	c	TOT	Commenti
1	10	8		
2				

TRACCIA

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio dati relativo agli arresti effettuati dalle forze di Polizia degli Stati Uniti dopo i disordini del 6 gennaio al Campidoglio (Capitol) a Washington D.C. che contengono le informazioni su ID arrestato (int), Cognome, Nome, Forza di Polizia (FBI, CIA, Polizia Metropolitana Washington), organizzazione (QAnon, Proud Boys, Boogaloo), teoria del complotto preferita (Gates, Luna, Renzi), numero di reati contestati (intero), sciamano (booleano, true/false), numero tatuaggi (intero).

a) Tramite JPA, si deve gestire l'archivio persistente dei dati degli arresti su DB (EsameDB), dove la chiave primaria è la id. Devono essere previste delle query per Cognome, per Forza di Polizia, per organizzazione, per teoria del complotto preferita, ed una query che restituisce tutte le entry del database

1. Deve essere previsto un bean Singleton che inizializzi l'archivio
2. Scrivere un client basato su invocazione di un bean stateless che prevede la stampa di:
 - a. Tutti gli arrestati che ritengono di essere sciamani (sciamano= true)
 - b. Tutti gli arrestati di una organizzazione data in input
 - c. Tutti gli arrestati con meno di un numero di tatuaggi dato in input [12 punti]

b) Scrivere un client basato su messaggi e la corrispondente parte lato server che invia un messaggio per aggiornare, per un dato arrestato, il numero di reati contestati e il numero di tatuaggi. Il messaggio contiene gli aggiornamenti ai nuovi valori (id arrestato, variazione reati contestati (+1, +2, -1, etc), variazione numero di tatuaggi) e si deve prevedere l'aggiornamento dei valori. Quando viene effettuato questo aggiornamento, tramite un evento viene stampato sulla console del server un messaggio di avviso "Dati aggiornati" e se il numero di reati contestati scende a 0, allora si stampa anche "L'arrestato COGNOME NOME può essere rilasciato" (con COGNOME e NOME presi dal suo record). Inoltre, se il numero di tatuaggi è sopra 10, allora automaticamente si mette il valore "Sciamano" a True. [10 punti].

c) Descrivere l'Object Relational Mapping (ORM) e il ruolo dell'Entity Manager [8 punti]

Dati di esempio del DB

ID	Cognome	Nome	Polizia	Organizzazione	Complotto	Num. Reati	Sciamano	Num. Tatuaggi
1	Angeli	Jack	FBI	QAnon	Renzi	3	True	12
2	Hope	Bob	CIA	Proud Boys	Gates	2	False	4
3	Mack	John	FBI	Boogaloo	Luna	4	False	2

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) si risponde barrando le caselle apposite.

Programmazione Distribuita – Prova scritta del 11/01/2020

Cognome e Nome (stampatello): _____ Matricola: _____

PREFERENZE PER ORALE (giorni esclusi!):

- 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:
 - o Entità, Interfacce, EJB, Client EJB
 - o JMS: MDB e Client

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

- Il progetti funzionanti, entro le ore **21:00 del 14/1/2021**, vanno sottomessi tramite la piattaforma Teams (Istruzioni verranno inviate), 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
 - o I progetti da consegnare (progetti NetBeans) si devono chiamare "COGNOMENOMEXXX" nei rispettivi files COGNOMENOMEXXX.zip (devono essere esportati).
 - o L'archivio che li contiene TUTTI insieme deve chiamarsi COGNOMENOME_EE.zip
 - o 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.
 - o 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 21 del 14 Gennaio 2021

AMMISSIONI ed ORALI pubblicati sulla piattaforma