

Cognome e Nome (stampatello): _____ Matricola: _____

Scrivere in stampatello Cognome e Nome su ogni foglio	I progetti funzionanti, 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 del 16/07/2019
---	--

Riservato alla Commissione per la Correzione

a	b	c	d	TOT	Commenti
/15	/5	/5	/5	/30	

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio di ventilatori, persistente sul DB, che contengano le informazioni su id ventilatore, modello, marca, giri/al minuto (int), rumore (da 0 a 10), elementi venduti, nazione produttore.

a) [15 punti] Tramite JPA, si deve gestire l'archivio persistente dei negozi su DB (EsameDB), dove la chiave primaria è la id.

1. Devono essere previste delle query per id, per regione, ed una query che restituisce tutti i ventilatori
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 tutti i ventilatori prodotti in una specifica nazione, argomento passato come parametro dall'utente (stdin) (*PrintByCountry*)
4. Scrivere un client basato su invocazione di un bean stateless che prevede la stampa di tutti i ventilatori con velocità superiore a 2000 giri al minuto (*PrintQuickFan*)

b) [5 punti] Scrivere un interceptor che provvede a stampare il numero di volte che il metodo *PrintByCountry* viene invocato [3 punti]

c) [5 punti] Scrivere un client basato su messaggi che invia un messaggio per aggiornare la quantità di elementi venduti per un ventilatore. Il messaggio contiene il numero di nuove vendite effettuate per un dato ventilatore (e la sua id), e si deve prevedere l'aggiornamento del valore delle vendite (cioè va sommato al precedente il valore passato nel messaggio). Quando viene cambiato tale valore, tramite un evento viene stampato sulla console del server un messaggio di avviso [5]

d) [5 punti] Rendendo i metodi del bean invocabili come Web Services, scrivere un client basato su invocazione di Web Services che modifichi la velocità in giri al minuto (chiesto all'utente). Quando viene modificato questo valore, allora tramite un evento viene stampato sulla console del server un messaggio di avviso.

Modello	Marca	Giri al minuto	Rumore	Elementi venduti	Nazione
SoffioFresco	VentilatoriPerTutti	1500	4	10	Italia
Wind 3000	Sturm	2200	6	12	Germania
BrezzaLeggera	Frescura	1000	5	11	Francia

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 si deve chiamare: jms/javaee7/Topic

ATTENZIONE: si richiede di:

- Indicare a quali parti della prova (a, b, c, d, e) 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

INFORMAZIONI SUL PROGRAMMA: sono state svolte le parti: ☐ a ☐ b ☐ c ☐ d

Indicare **sul retro di questo foglio** quali sono le classi che compongono il programma, su che facciata dei fogli si trovano e che ruolo svolgono

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

DESCRIZIONE DEL PROGRAMMA

Cognome e Nome (stampatello): _____ Matricola: _____

FOGLIO DA CONSERVARE

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio di ventilatori, persistente sul DB, che contengano le informazioni su id ventilatore, modello, marca, giri/al minuto (int), rumore (da 0 a 10), elementi venduti, nazione produttore.

- a) [15 punti] Tramite JPA, si deve gestire l'archivio persistente dei negozi su DB (EsameDB), dove la chiave primaria è la id.
1. Devono essere previste delle query per id, per regione, ed una query che restituisce tutti i ventilatori
 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 tutti i ventilatori prodotti in una specifica nazione, argomento passato come parametro dall'utente (stdin) (*PrintByCountry*)
 4. Scrivere un client basato su invocazione di un bean stateless che prevede la stampa di tutti i ventilatori con velocità superiore a 2000 giri al minuto (*PrintQuickFan*)
- b) [5 punti] Scrivere un interceptor che provvede a stampare il numero di volte che il metodo *PrintByCountry* viene invocato [3 punti]
- c) [5 punti] Scrivere un client basato su messaggi che invia un messaggio per aggiornare la quantità di elementi venduti per un ventilatore. Il messaggio contiene il numero di nuove vendite effettuate per un dato ventilatore (e la sua id), e si deve prevedere l'aggiornamento del valore delle vendite (cioè va sommato al precedente il valore passato nel messaggio). Quando viene cambiato tale valore, tramite un evento viene stampato sulla console del server un messaggio di avviso [5]
- d) [5 punti] Rendendo i metodi del bean invocabili come Web Services, scrivere un client basato su invocazione di Web Services che modifichi la velocità in giri al minuto (chiesto all'utente). Quando viene modificato questo valore, allora tramite un evento viene stampato sulla console del server un messaggio di avviso.

Nome Negozio	Direttore	Vendite birre alcoliche	Vendite birre analcoliche	Città	Provincia	Regione
BeviBene	John Budweiser	430.015	132.210	Napoli	Napoli	Campania
BirraESaiCosaBevi	Nanni Moretti	640.000	212.133	Roma	Roma	Lazio
BirreOggi	Pasquale Poretti	345.941	615.231	Cernusco	Milano	Lombardia

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) I topic si devono chiamare: jms/javaee7/Topic1 e jms/javaee7/Topic2

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

- Il progetti funzionanti, entro le **ore 9:00 del 16/07/2019**, vanno sottomessi sulla Piattaforma E-Learning al link “PD-SimulazioneEsame_4”, 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 piattaforma Piattaforma E-Learning dell'Anno Accademico in corso
- Gli orali si terranno a partire dalle date indicate nello studio della Prof. Delfina Malandrino (Dip. di Informatica, IV piano, Stecca 7, studio n. 40). 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 9 del 16 Dicembre 2019