

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 del 12/06/2019
---	---

**Riservato alla Commissione per la Correzione**

a	b	c	d	e	TOT	Commenti
/15	/3	/5	/3	/4	/30	

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio di negozi di elettronica, persistente sul DB, che contengano le informazioni su id negozio, nome negozio, direttore, vendite anno precedente, vendite anno corrente, città, provincia, regione.

- a) [15 punti] Tramite JPA, si deve gestire l'archivio persistente dei negozi su DB (EsameDB), dove la chiave primaria è la id.
- Devono essere previste delle query per id, per regione, ed una query che restituisce tutti i negozi d'Italia
  - 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 negozi di una data regione, argomento passato come parametro dall'utente (stdin) (*PrintByRegion*)
  - Scrivere un client basato su invocazione di un bean stateless che prevede la stampa di tutti i negozi che hanno vendite dell'anno corrente maggiori di quelle dell'anno precedente (*PrintActiveShops*)
- b) [3 punti] Scrivere un interceptor che provvede a stampare il numero di volte che il metodo *PrintByRegion* viene invocato [3 punti]
- c) [5 punti] Scrivere un client basato su messaggi che invia un messaggio per aggiornare la quantità di vendite per l'anno corrente. Il messaggio contiene il numero di nuove vendite effettuate per un negozio, e si deve prevedere l'aggiornamento del valore delle vendite dell'anno corrente (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) [3 punti] Rendendo i metodi del bean invocabili come Web Services, scrivere un client basato su invocazione di Web Services che stampi la lista di tutti i negozi presenti su DB
- e) [4 punti] Descrivere e confrontare Interceptors e Decorators

Nome Negozio	Direttore	Vendite anno precedente	Vendite anno corrente	Città	Provincia	Regione
Elettroshop	Stefano Lavori	430.015	132.210	Napoli	Napoli	Campania
BuildYourPC	Marco Montagna Zuccherò	640.000	212.133	Roma	Roma	Lazio
ElettronicaOggi	Guglielmo Cancelli	345.941	615.231	Cernusco	Milano	Lombardia

**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, 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 ☐ e

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**

Indicare quali sono le classi che compongono il programma, su che facciata dei fogli si trovano e che ruolo svolgono

Cognome e Nome (stampatello): \_\_\_\_\_ Matricola: \_\_\_\_\_

## FOGLIO DA CONSERVARE

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio di negozi di elettronica, persistente sul DB, che contengano le informazioni su id negozio, nome negozio, direttore, vendite anno precedente, vendite anno corrente, città, provincia, regione.

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 negozi d'Italia
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 negozi di una data regione, argomento passato come parametro dall'utente (stdin) (*PrintByRegion*)
4. Scrivere un client basato su invocazione di un bean stateless che prevede la stampa di tutti i negozi che hanno vendite dell'anno corrente maggiori di quelle dell'anno precedente (*PrintActiveShops*)

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

c) [5 punti] Scrivere un client basato su messaggi che invia un messaggio per aggiornare la quantità di vendite per l'anno corrente. Il messaggio contiene il numero di nuove vendite effettuate per un negozio, e si deve prevedere l'aggiornamento del valore delle vendite dell'anno corrente (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) [3 punti] Rendendo i metodi del bean invocabili come Web Services, scrivere un client basato su invocazione di Web Services che stampi la lista di tutti i negozi presenti su DB

e) [4 punti] Descrivere e confrontare Interceptors e Decorators

Nome Negozio	Direttore	Vendite anno precedente	Vendite anno corrente	Città	Provincia	Regione
Elettroshop	Stefano Lavori	430.015	132.210	Napoli	Napoli	Campania
BuildYourPC	Marco Montagna Zuccherò	640.000	212.133	Roma	Roma	Lazio
ElettronicaOggi	Guglielmo Cancelli	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) Il topic deve chiamarsi: jms/javaee7/Topic

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

- Il progetti funzionanti, entro le **ore 9:00 del 12/06/2019**, vanno sottomessi sulla Piattaforma E-Learning al link "PD-10-Giugno-2019", dove verranno anche fornite le ammissioni, il giorno 12 giugno in tarda sera. Gli orali si svolgeranno nelle giornate del **13 Giugno** (tutta la giornata, iniziando alle 9) e il **14 Giugno** (a partire dalle 12).
- 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 **leggi**, ...) 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 del Prof. Scarano (Dip. di Informatica, IV piano, Stecca 7, studio n. 1) 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: 12 Giugno 2019**

**ORALI PREVISTI: 13 e 14 Giugno 2019**