

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

Scrivere in stampatello Cognome e Nome su ogni foglio	I progetti funzionanti esportati in un file COGNOMENOME.zip con file Readme.txt che illustri le modifiche, vanno sottomessi su Piattaforma e-learning entro le ore <b>15:00 del 03/04/2023</b> . Gli orali inizieranno orientativamente il <b>04/04/2023</b> ma attendete conferme sulla piattaforma di dipartimento e sul Teams della vostra classe.
---	---

**Riservato alla Commissione per la Correzione**

a	b	c	d	e	TOT	Commenti

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio di musei, persistente sul DB, che contengano le informazioni sul id, nome museo, direttore, numero visitatori per anno, città, provincia, regione.

a) Tramite JPA, si deve gestire l'archivio persistente dei circoli 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 musei 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 musei di una data regione, argomento passato come parametro dall'utente (stdin) (*PrintByRegion*) [12 punti]

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

c) Scrivere un client basato su messaggi che invia un messaggio per aggiornare il numero di visitatori per anno. Quando viene cambiato tale valore, tramite un evento viene stampato sul server il messaggio "Aggiornamento effettuato con successo". Se il numero di visitatori supera il valore 5 Milioni, un evento verrà generato con messaggio "Complimenti al Direttore X", dove X è il Direttore del Museo per il quale è stato effettuato l'aggiornamento [5].

d) 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 musei presenti su DB [3 punti].

e) Dopo aver descritto cosa è una Service Oriented Architecture descrivere dettagliatamente:

- le 3 caratteristiche che devono avere i suoi servizi
- la definizione di Web Service [7 punti]

Nome Museo	Direttore	Numero visitatori/anno	Città	Provincia	Regione
Museo di Paestum	Gabriel Zuchriegel	430.000	Capaccio	Salerno	Campania
Musei Vaticani	Barbara Jatta	5.978.804	Roma	Roma	Lazio
Galleria degli Uffizi	Eike Dieter Schmidt	1.935.901	Firenze	Firenze	Toscana

**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, 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:
  - o Entità, Interfacce, EJB, Client EJB
  - o Database populatore
  - o JMS: MDB e Client

**INFORMAZIONI SUL PROGRAMMA:**

sono state svolte le parti: ☐ a ☐ b ☐ c ☐ d ☐ e

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

## FOGLIO DA CONSERVARE

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio di musei, persistente sul DB, che contengano le informazioni sul id, nome museo, direttore, numero visitatori per anno, città, provincia, regione.

- a) Tramite JPA, si deve gestire l'archivio persistente dei circoli 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 musei 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 musei di una data regione, argomento passato come parametro dall'utente (stdin) (*PrintByRegion*) [12 punti]
- b) Scrivere un interceptor che provvede a stampare il numero di volte che il metodo *PrintByRegion* viene invocato [3 punti]
- c) Scrivere un client basato su messaggi che invia un messaggio per aggiornare il numero di visitatori per anno. Quando viene cambiato tale valore, tramite un evento viene stampato sul server il messaggio "Aggiornamento effettuato con successo". Se il numero di visitatori supera il valore 5 Milioni, un evento verrà generato con messaggio "Complimenti al Direttore X", dove X è il Direttore del Museo per il quale è stato effettuato l'aggiornamento [5].
- d) 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 musei presenti su DB [3 punti].
- e) Dopo aver descritto cosa è una Service Oriented Architecture descrivere dettagliatamente:
- le 3 caratteristiche che devono avere i suoi servizi
  - la definizione di Web Service [7 punti]

Nome Museo	Direttore	Numero visitatori/anno	Città	Provincia	Regione
Museo di Paestum	Gabriel Zuchriegel	430.000	Capaccio	Salerno	Campania
Musei Vaticani	Barbara Jatta	5.978.804	Roma	Roma	Lazio
Galleria degli Uffizi	Eike Dieter Schmidt	1.935.901	Firenze	Firenze	Toscana

### 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

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

- Il progetti funzionanti, entro le **ore 15:00 del 03/04/2023**, vanno sottomessi sulla Piattaforma E-Learning al link "PD\_29Marzo2023", 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
  - L'archivio che contiene tutti i progetti deve chiamarsi **COGNOMENOME\_EE.zip**
  - Deve esserci un **UNICO file readme.txt** (non **Readme**, non **leggi**, ...) all'interno di **COGNOMENOME.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 data indicata sulla traccia

**TERMINE PER L'INVIO DEL PROGETTO: 03/04/2023**