

Cognome e Nome (stampatello): _____ Matricola: _____

Scrivere in stampatello Cognome e Nome su ogni foglio	I progetti funzionanti con un file Readme.txt che illustri le modifiche vanno sottomessi su Piattaforma e-learning, in un unico archivio COGNOMENOME.zip entro le ore 9 del 18/12/2018
---	--

Riservato alla Commissione per la Correzione

a	b	c	TOT	Commenti

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio di task persistenti sul DB, che contengono le informazioni su ID (int), nome del task, tipo del task (comunicazione, innovazione, amministrazione), ore di lavoro previste (intero tra 1 e 8), nome dell'assegnatario del task (stringa), Percentuale di completamento (intero tra 0 (=non iniziato) e 100 (=completato))

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

- Devono essere previste delle query per id, per tipo del task, per ore di lavoro, per nome di assegnatario, per ore di lavoro minore di, per task completati, per task non completati ed una query che restituisce tutti i task `getAllTasks()`
- 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 task (`getAllTasks()`)
 - Tutti i prodotti che non hanno assegnatario (stringa "nessuno") [15 punti]

b) Scrivere un interceptor che provvede a stampare a video il numero totale di volte che il metodo `getAllTasks()` è stato chiamato [5 punti].

c) Scrivere un client basato su messaggi che invia un messaggio con un nuovo task da inserire con l'aggiornamento all'interno del DB. Utilizzare la tecnica delle Callback Annotations per verificare, ogni volta che un nuovo task viene inserito, che si tratti di un task valido (ad esempio, ore di lavoro = 12 non è valido, così come completamento <0 o > 100). In caso il task non sia valido viene lanciata una eccezione [10 punti].

Dati di esempio del DB

ID	Nome	Tipo	Ore	Assegnatario	Completamento
1	Completare pagina home del sito web	comunicazione	4	Maria	0
2	Presentazione novità	innovazione	2	nessuno	75
3	Emettere fattura cliente 2	amministrazione	7	Pasquale	100

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

INFORMAZIONI SUL PROGRAMMA:

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

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: _____

FOGLIO DA CONSERVARE

Scrivere un insieme di Enterprise JavaBeans e client che rappresentino un archivio di task persistenti sul DB, che contengono le informazioni su ID (int), nome del task, tipo del task (comunicazione, innovazione, amministrazione), ore di lavoro previste (intero tra 1 e 8), nome dell'assegnatario del task (stringa), Percentuale di completamento (intero tra 0 (=non iniziato) e 100 (=completato))

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

1. Devono essere previste delle query per id, per tipo del task, per ore di lavoro, per nome di assegnatario, per ore di lavoro minore di, per task completati, per task non completati ed una query che restituisce tutti i task `getAllTasks()`
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:
 - a. Tutti i task (`getAllTasks()`)
 - b. Tutti i prodotti che non hanno assegnatario (stringa "nessuno") [15 punti]

b) Scrivere un interceptor che provvede a stampare a video il numero totale di volte che il metodo `getAllTasks()` è stato chiamato [5 punti].

c) Scrivere un client basato su messaggi che invia un messaggio con un nuovo task da inserire con l'aggiornamento all'interno del DB. Utilizzare la tecnica delle Callback Annotations per verificare, ogni volta che un nuovo task viene inserito, che si tratti di un task valido (ad esempio, ore di lavoro = 12 non è valido, così come completamento <0 o > 100). In caso il task non sia valido viene lanciata una eccezione [10 punti].

Dati di esempio del DB

ID	Nome	Tipo	Ore	Assegnatario	Completamento
1	Completare pagina home del sito web	comunicazione	4	Maria	0
2	Presentazione novità	innovazione	2	nessuno	75
3	Emettere fattura cliente 2	amministrazione	7	Pasquale	100

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 18/12/2018**, vanno sottomessi sulla Piattaforma E-Learning al link "PD_14Dicembre2018", 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 **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 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: 18 Dicembre 2018 ore 9