

Esercitazione:

A – Rispondi alle seguenti domande argomentando la risposta:

- 1) Descrivi l'architettura monolitica.
- 2) Che cos'è un microservizio?
- 3) Definisci le differenze tra architettura monolitica e microservizi.
- 4) Cosa si intende per xml? E per Json?
- 5) Cosa si intende per scalabilità?
- 6) Che ruolo ha il team nella realizzazione di un microservizio?
- 7) Cosa si intende per inconsistenza dei dati?
- 8) Che differenza c'è tra la comunicazione sincrona e quella asincrona?
- 9) Definisci il Circuit Breaker.
- 10) Cosa si intende per continuous deployment?
- 11) Definisci un ORM.
- 12) Come può essere classificato un sistema ORM? Argomenta.
- 13) Cos'è Hibernate?
- 14) Cosa si intende per persistenza dei dati?
- 15) Quali sono le caratteristiche degli stati di persistenza?
- 16) Cos'è una session? E una Session Factory?
- 17) Cosa si intende per jdbc?
- 18) Che cos'è la serializzazione? E la deserializzazione?
- 19) Cos'è Unified Modeling Language?
- 20) Definisci il diagramma delle classi.
- 21) Spiega le differenze tra associazioni, aggregazioni e composizione.
- 22) Si elencano le caratteristiche di un Java Bean e quelle di un Entity Bean.
- 23) Quali sono le differenze tra jdbc e hibernate?
- 24) Descrivere il ciclo di vita di un'entità.
- 25) Quale libreria viene implementata ed estesa da Hibernate?
- 26) Descrivere Statement, Prepared Statement e Result Set.
- 27) A cosa serve il file hibernate.cfg.xml?
- 28) A cosa serve il file pom.xml?
- 29) Quali sono le opzioni possibili per hibernate.hbm2ddl.auto?
- 30) Come si possono configurare le relazioni tra Entity in Hibernate?

B – Si crei un progetto jdbc che gestisca il database di studenti.

Ogni Studente che dovrà avere i seguenti attributi: matricola, nome, cognome, età, data di nascita, nome del corso e esami sostenuti.

All'interno di una classe main implementare i metodi per:

- Aggiungere uno studente
- Stampare uno studente
- Stampare tutti gli studenti
- Cancellare uno studente
- Cancellare tutti gli studenti
- Aggiornare uno studente
- Aggiornare uno studente in base alla matricola

C – Definire l'entità RichiestaFerie con le seguenti proprietà:

- id
- dataCreazione
- dataModifica
- emailAziendale
- giornoInizio
- giornoFine
- motivazione
- stato
- motivazioneRifiuto

Implementare la persistenza dell'entity tramite l'ORM Hibernate e l'implementazione del design pattern DAO.

NB: creare le istanze e salvarle nel DB.

D – Creare un progetto Java che usi Maven per gestire le dipendenze e che usi Hibernate per gestire la connessione con il database.

Il database dovrà contenere tre tabelle Studente, Matricola e Corso.

Studente: id come chiave primaria, nome e cognome.

Matricola: id come chiave primaria, e matricola.

Corso: id come chiave primaria, descrizione e denominazione.

Inserire per ogni tabella 5 righe ed individuare le relazioni tra le tabelle

Il progetto avrà:

- Le classi o i file di configurazione
- Tre entityBean
- Una interfacciaDao con i metodi di crud generici
- Tre classi Dao che implementano interfacciaDao
- Una classe di test con un metodo main