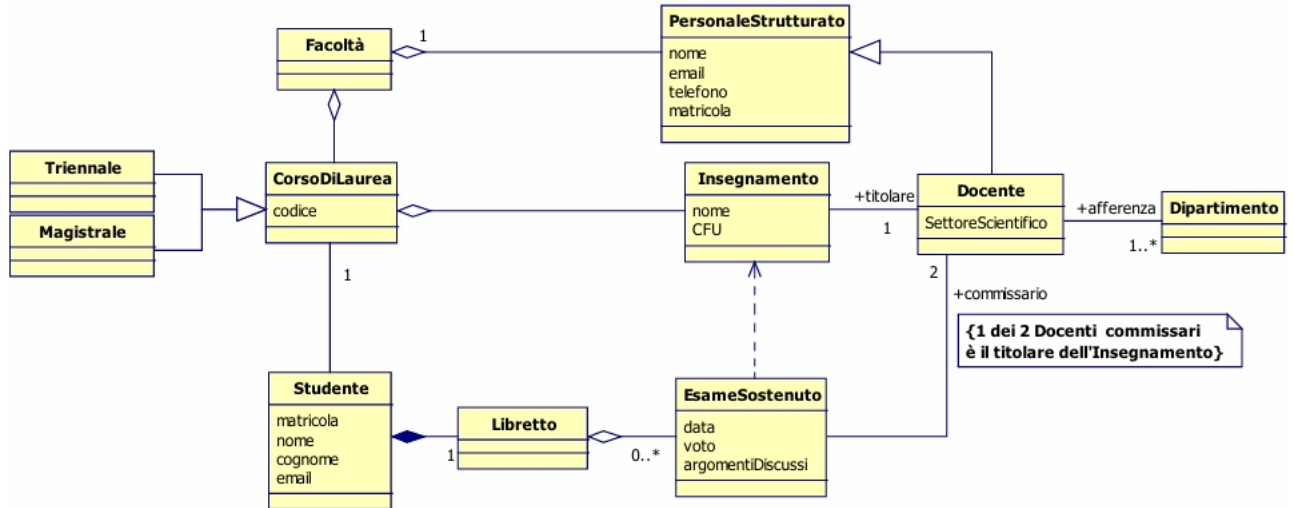


Verifica di Tecnologie e progettazione di sistemi informatici e di telecomunicazioni

Nome Cognome: _____ Classe: _____ Data: _____

Esercizio 1

Osserva il class diagram UML sotto e sviluppa la risposta in base all'elenco puntato:



- In base a quanto visto a lezione, commenta l'intero diagramma, presentando relazioni tra classi e spiegandone le diverse tipologie; **spiegare generalizzazione, associazione, aggregazione, composizione e dipendenza (1 pt. ciascuna)**
- 2 pt. • Se volessimo esplicitare la presenza di metodi per alcune classi, dove dovremmo inserirli?
- 3 pt. • Se volessimo distinguere attributi/metodi in base alla visibilità, cosa dovremmo aggiungere al diagramma? (public, private, package e protected)

Punteggio: ____/10

Esercizio 2

Disegna lo use-case diagram UML del seguente sistema:

La società di servizi VisitMalta365 permette di noleggiare automobili per visitare in tutta comodità le isole maltesi.

La sede fisica si trova vicino all'aeroporto internazionale di Malta, presso cui il cliente deve recarsi per ritirare il veicolo a inizio noleggio e riconsegnarlo al termine del periodo.

È obbligatorio procedere alla prenotazione del noleggio tramite il sito della società, a cui ci si deve però registrare e autenticare per visualizzare il catalogo di macchine. Una volta scelta l'automobile e pagato in anticipo l'importo, è possibile recarsi in sede e ritirare la vettura dall'operatore ivi presente.

Al termine del periodo di noleggio, il cliente riporta il veicolo e si emette una sanzione a suo carico se vengono rilevati dei danni dall'operatore.

Punteggio: ____/10

Esercizio 3

Disegna il sequence diagram UML di quanto avviene durante la fase di login di un sistema così codificato:

```
import java.util.Scanner;

// Questa classe simula l'interfaccia con cui l'utente reale interagisce
class InterfacciaUtente {

    // Metodo che avvia la procedura di login
    void login() {

        // Creiamo uno scanner per leggere da tastiera
        Scanner scanner = new Scanner(System.in);

        // Chiediamo all'utente di inserire username e password
        System.out.print("Inserisci username: ");
        String username = scanner.nextLine();

        System.out.print("Inserisci password: ");
        String password = scanner.nextLine();

        // Creiamo l'oggetto che rappresenta il server di autenticazione
        ServerAutenticazione server = new ServerAutenticazione();

        // Inviemo i dati al server per la verifica
        boolean success = server.verificaCredenziali(username, password);

        // Mostriamo il risultato a video
        if (success) {
            System.out.println("Accesso effettuato");
        } else {
            System.out.println("Accesso negato");
        }
    }
}

// Questa classe rappresenta il server che controlla nome utente e password
class ServerAutenticazione {

    // Metodo che verifica se le credenziali sono corrette
    boolean verificaCredenziali(String username, String password) {
        return "admin".equals(username) && "1234".equals(password);
    }
}
```

Suggerimento: disponi un attore "Utente", un oggetto classe "InterfacciaUtente" e un oggetto classe "ServerAutenticazione" nel diagramma

Punteggio: ____/10