ESAME DI INGEGNERIA DEL SOFTWARE 2023/2024 - UNIVERSITA' DI FERRARA

NOME _____ MATRICOLA _____ DATA_____

Informazioni di base

Gestione di un Sistema di Prenotazione di Libri in una Biblioteca

Lo studente dovrà progettare un sistema per la gestione delle prenotazioni di libri in una biblioteca. Il sistema, utilizzato anche dallo Staff della Biblioteca, consente agli utenti di prenotare un libro, verificare la disponibilità e annullare una prenotazione

ESERCIZIO 1: UML e progettazione (10 punti)

Diagramma UML delle classi:

- E' richiesta la definizione delle classi principali del sistema, includendo almeno: `Utente`, `Libro`, `Prenotazione`, `StaffBiblioteca` e `Biblioteca`.
- Aggiungere una classe base `Persona` con attributi generali (`nome`, `email`).
- Le classi `Utente` e `StaffBiblioteca` devono **estendere** `Persona`.
- La classe `StaffBiblioteca` può aggiungere o rimuovere libri.
- Rappresentare le relazioni tra le classi (ad esempio, un utente può avere più prenotazioni).

Realizzare un Diagramma UML delle attività:

- Disegnare un diagramma delle attività che descriva il processo di prenotazione di un libro da parte di un utente attraverso lo Staff. (6 pt)
- Includere i passi principali come ricerca del libro, verifica della disponibilità e conferma della prenotazione. (4 pt)

ESERCIZIO 2: Java (10 punti)

Implementa il sistema in Java seguendo la progettazione UML realizzata nella parte 1.

Requisiti:

- Crea le classi `Utente`, `Libro`, `Prenotazione`, `StaffBiblioteca` e `Biblioteca`, con attributi e metodi appropriati. (2 pt)
- La classe `Utente` deve estendere `Persona` e rappresentare un utente registrato della biblioteca. (1 pt)
- La classe `StaffBiblioteca` deve anch'essa estendere `Persona` e avere metodi per aggiungere e rimuovere libri dalla biblioteca. (1 pt)
- La classe `Biblioteca` deve gestire un elenco di libri disponibili. (1 pt)

NOME _____ MATRICOLA _____ DATA_____

- Implementare un metodo `prenotaLibro(Utente utente, Libro libro)` nella classe `Biblioteca`, che consenta a un utente di prenotare un libro se disponibile. (2 pt)
- Implementare un metodo `annullaPrenotazione(Utente utente, Libro libro)`. (2 pt)
- Aggiungere una classe `Main` per testare il sistema con uno scenario di esempio. (1 pt)

ESERCIZIO 3: Domande a risposta chiusa (2 punto per ogni risposta esatta)

1. Quale principio stabilisce che le classi devono essere aperte per l'estensione, ma chiuse per la modifica?
    a. Single Responsibility Principle (SRP)
    b. Open/Closed Principle (OCP)
    c. Liskov Substitution Principle (LSP)
    d. Nessuna delle precedenti

2. Quale dei seguenti strumenti è utilizzato per il controllo delle versioni del codice sorgente in un progetto software?
    a. GIT
    b. Trello
    c. Docker
    d. Nessuna delle precedenti

3. Quale tra le seguenti affermazioni è vera riguardo al concetto di incapsulamento in programmazione orientata agli oggetti?
    a. L'incapsulamento consente a una classe di esporre solo ciò che è necessario all'esterno, nascondendo i dettagli interni.
    b. L'incapsulamento è un meccanismo per evitare l'ereditarietà.
    c. L'incapsulamento rende una classe completamente immutabile.
    d. Nessuna delle precedenti

NOME _____ MATRICOLA _____ DATA_____

4. Quale tra i seguenti concetti si riferisce all'abilità di una piattaforma software di essere utilizzata da persone con disabilità?

    a. Usabilità
    b. Accessibilità
    c. Compatibilità
    d. Nessuna delle precedenti

5. Cos'è il polimorfismo in Java?
    a. Un metodo può essere sovrascritto solo se è dichiarato come `public`.
    b. La classe derivata non può modificare i metodi della classe base.
    c. Un oggetto può essere istanziato in più forme o avere comportamenti diversi a seconda del tipo.
    d. Nessuna delle precedenti

6. Quale sarà l'output del seguente codice Java?

```
class A {
        void print() {
                System.out.println("Classe A");
         }
}

class B extends A {
        void print() {
                System.out.println("Classe B");
                 }
         }

        public class Test {
                public static void main(String[] args) {
                        A obj = new B();
                        obj.print();
                }
        }
```

    a. "Classe A"
    b. "Classe B"
    c. Errore di compilazione
    d. Nessuna delle precedenti