

PaginaIniziale

1. Singleton Pattern

- **Dove si applica:** gestione della connessione al database
- **Motivazione:** si utilizza per garantire che ci sia solo una singola istanza di `DatabaseManager` in tutta l'applicazione. Questo evita problemi legati alla creazione multipla di connessioni e centralizza la gestione della connessione al database. Nell'attuale implementazione, la connessione viene stabilita solo una volta e la variabile `isDatabaseConnected` aiuta a mantenere questo stato
- **Codice:**

```
private static DatabaseManager databaseManager;
```

2. Command Pattern

- **Dove si applica:** gestione delle azioni dei bottoni
- **Motivazione:** si utilizza per incapsulare le azioni dei bottoni (come `registrazioneButton` o `loginButton`) in oggetti di comando separati. Questo approccio facilita la gestione delle azioni e consente di estendere facilmente le funzionalità aggiungendo nuovi comandi senza modificare il codice esistente
- **Codice:**

```
registrazioneButton.addActionListener(this);
```

```
loginButton.addActionListener(this);
```

3. Factory Method Pattern

- **Dove si applica:** creazione delle finestre
- **Motivazione:** si utilizza per centralizzare la creazione di queste finestre (come `Reistrazione` o `LoginFrame`).
- **Codice**

```
new RegistrazioneFrame(databaseManager, null);  
new LoginFrame(databaseManager);
```