**Ecco alcuni esercizi java advanced, puoi evolvere l’esercizio come vuoi una volta terminata la traccia( ad esempio, eccezioni, test ecce cc)**

**Traccia 1:**

* **Classe FormaGeometrica:**
* Attributi: colore (String), metodi di accesso.
* Metodo calcolaArea() da implementare nelle sottoclassi.
* Metodo toString() per rappresentare le informazioni della forma.
* **Sottoclassi Cerchio e Quadrato:**
* Estendono la classe FormaGeometrica.
* Implementano il metodo calcolaArea() specifico per la forma geometrica.
* **Classe GestoreForme:**
* Lista di oggetti FormaGeometrica.
* Metodi per aggiungere forme, visualizzare l'elenco e calcolare l'area totale.
* Creare un main dove testare tutte le funzionalità

**Traccia 1a**

* **Classe Film:**
* Attributi: titolo (String), regista (String), annoUscita (int), valutazione (double), metodi di accesso.
* Costruttore che accetta titolo, regista, annoUscita e valutazione.
* Metodo toString() per rappresentare le informazioni del film.
* **Classe CatalogoFilm:**
* Lista di oggetti Film.
* Metodi per aggiungere nuovi film, rimuovere film, visualizzare l'elenco completo dei film e cercare film per regista o anno di uscita.
* Creare un main dove testare tutte le funzionalità

**Traccia 2:**

* **Classe Dipendente:**
* Attributi: nome (String), salario (double), metodi di accesso.
* Metodo calcolaSalarioAnnuo() da implementare nelle sottoclassi.
* Metodo toString() per rappresentare le informazioni del dipendente.
* **Sottoclassi Manager e Sviluppatore:**
* Estendono la classe Dipendente.
* Implementano il metodo calcolaSalarioAnnuo() specifico per il tipo di dipendente.
* **Classe GestoreDipendenti:**
* Lista di oggetti Dipendente.
* Metodi per aggiungere dipendenti, visualizzare l'elenco e calcolare la somma totale dei salari.
* Creare un main dove testare tutte le funzionalità

**Traccia 2a:**

* **Classe Contatto:**
* Attributi: nome (String), cognome (String), numeroTelefono (String), email (String), metodi di accesso.
* Costruttore che accetta nome, cognome, numeroTelefono e email.
* Metodo toString() per rappresentare le informazioni del contatto.
* **Classe RubricaTelefonica:**
* Lista di oggetti Contatto.
* Metodi per aggiungere nuovi contatti, rimuovere contatti, visualizzare l'elenco completo dei contatti e cercare contatti per nome o numero di telefono
* Creare un main dove testare tutte le funzionalità

**Traccia 3 :**

* **Classe Animale:**
* Attributi: nome (String), età (int), metodo di accesso.
* Metodo emettiSuono() da implementare nelle sottoclassi.
* Metodo toString() per rappresentare le informazioni dell'animale.
* **Sottoclassi Leone e Elefante:**
* Estendono la classe Animale.
* Implementano il metodo emettiSuono() specifico per il tipo di animale.
* **Classe GestoreAnimali:**
* Lista di oggetti Animale.
* Metodi per aggiungere animali, visualizzare l'elenco e far emettere suoni a tutti gli animali.
* Creare un main dove testare tutte le funzionalità

**Traccia 4:**

* **Classe Libro:**
* Attributi: titolo (String), autore (String), genere (String), disponibilità (boolean), metodi di accesso.
* Costruttore che accetta titolo, autore, genere e impostazione di disponibilità.
* Metodo toString() per rappresentare le informazioni del libro.
* **Classe Biblioteca:**
* Lista di oggetti Libro.
* Metodi per aggiungere libri, rimuovere libri, visualizzare l'elenco e cercare libri per autore o genere.
* Creare un main dove testare tutte le funzionalità

**Traccia 5:**

* **Classe Ordine:**
* Attributi: numeroOrdine (int), dataCreazione (Date o String), prodottiOrdinati (ArrayList di oggetti Prodotto), statoElaborazione (String), metodi di accesso.
* Costruttore che accetta numeroOrdine, dataCreazione, prodottiOrdinati e statoElaborazione.
* Metodo toString() per rappresentare le informazioni dell'ordine.
* **Classe Prodotto:**
* Attributi: nome (String), quantità (int), prezzo (double), metodi di accesso.
* Costruttore che accetta nome, quantità e prezzo.
* Metodo toString() per rappresentare le informazioni del prodotto.
* **Classe SistemaOrdini:**
* Lista di oggetti Ordine.
* Metodi per aggiungere nuovi ordini, rimuovere ordini, visualizzare l'elenco completo degli ordini e aggiornare lo stato di elaborazione di un ordine.
* Creare un main dove testare tutte le funzionalità

**Implementazione di Ereditarietà e Polimorfismo:**

* **Ereditarietà:**
* Usa la parola chiave extends per estendere le classi base nelle sottoclassi.
* Le sottoclassi ereditano attributi e metodi della classe base.
* **Polimorfismo:**
* Le sottoclassi possono implementare in modo diverso i metodi ereditati dalla classe base.
* Il polimorfismo consente di trattare oggetti di diverse classi derivate come oggetti della classe base.
* **Override:**
* Utilizza l'annotazione @Override quando implementi metodi nelle sottoclassi per garantire che stai effettivamente facendo l'override di un metodo della classe base