Esercizi Java

Esercizio 1

Scrivere una classe Java per gestire un conto corrente, su cui sono possibili le seguenti operazioni:

- Aprire conto corrente vuoto (con 0 euro)
- Aprire conto corrente con X euro
- Versare X euro nel conto
- Prelevare X euro dal conto
- Stampare un messaggio con il saldo attuale (nel metodo `toString()`)

Estendere la classe `ContoCorrente` per poter gestire conti con valute diverse, aggiungendo un campo valuta che può assumere valori "euro", "dollar", "pound" (usare Enumeration).

Definire il costruttore che prende in input il saldo iniziale e la valuta e sovrascrivere il metodo `toString()` per mostrare i dati appropriati.

Implementare una classe test in cui sono istanziati tre conti di valute diverse e mostrato il saldo di ognuna.

Esercizio 2

Creare una classe `Persona` con le variabili di istanza: nome (stringa) e data di nascita (vedi `java.time` e `java.util.Date`).

Creare una sottoclasse `Impiegato` che ha le seguenti variabili di istanza: nome e data di nascita (ereditati) e stipendio.

Definire costruttori e metodi set e get per le variabili di istanza, per entrambe le classi `Persona` e `Impiegato`.

Costruire una sottoclasse di `Impiegato`, chiamata `Stagista`, che contiene due variabili di istanza entrambe di tipo intero: `numeroPresenze`, che registra il numero di ore di presenza, e `numeroIdentificativoStage`

Definire costruttore e metodi set e get per le variabili di istanza.

Implementare una classe `ImpiegatiDemo` e nel metodo `main()` creare tre oggetti di tipo `Impiegato`, di cui almeno uno stagista, e memorizzarli in un array.

Aggiungere un metodo `cercaGiovane` che prende in input un vettore di impiegati e ritorna l'impiegato più giovane. Invocare il metodo sul vettore precedente e stampare le informazioni relative all'impiegato trovato.

Esercizio 3

Implementare le classi Java utili per descrivere il motore di un'automobile. Ogni motore è caratterizzato da:

1. cilindrata (intero)
2. numero_cilindri (intero)

Da queste informazioni è possibile derivare la potenza (espressa in cavalli, valore intero) in base al tipo di motore. Esistono tre tipi di motore:

- benzina – potenza: (cilindrata/numero_cilindri) * 0.1
- diesel – potenza: (cilindrata/numero_cilindri) * 0.2
- metano – potenza: ((cilindrata * 0.8)/numero_cilindri) * 0.25

Definire la classe astratta `Motore` e le opportune classi concrete. Implementare una classe test per verificare il funzionamento delle classi e dei metodi.

Esercizio 4

Implementare le classi Java necessarie a modellare questa situazione:

- *Ogni animale fa un verso e ha un certo numero di zampe:*
  - *Il gatto ha 4 zampe e miagola*
  - *Il cane ha 4 zampe e abbaia*
  - *Il tacchino ha 2 zampe e goglotta*
- *Ogni esemplare di animale ha un nome e un anno di nascita. Per semplicità si tiene traccia solo degli anni e l'età si calcola come il numero di anni trascorsi dalla data corrente*

Definire una classe astratta `Animale` e le sottoclassi concrete, implementando opportunamente i metodi per recuperare: verso dell'animale (stringa), numero di zampe (intero), ed età (intero). Sovrascrivere il metodo `toString()` per combinare queste informazioni in una stringa.

Nota: vedi `java.time` e `java.util.Date` per gestire le date.

Implementare inoltre il metodo `confronta(Animale a)` che permette di confrontare un animale con un altro (anche di specie diversa) in base alla loro età in anni.

Creare infine una classe test nel cui metodo `main()` sono istanziati alcuni animali e verificato il funzionamento dei metodi precedenti.

Esercizio 5

Definire il comportamento di un animale (verso e numero di zampe) tramite un'interfaccia `IAnimale`.

Modificare le classi precedenti per usare l'interfaccia `IAnimale` piuttosto che la classe astratta `Animale`.

Nota: la modellazione con classe astratta è più corretta e da preferire in casi come questo.