---------- Raccolta Terze Prove

Si consideri la seguente situazione che si vorrebbe modellare con un opportuno insieme di classi JAVA:

Un'azienda siciliana commercializza prodotti agricoli caratterizzati dal peso in gr. Si vogliono considerare i seguenti tipi di prodotti: *vini bianchi, vini rossi,* ed *oli*. I vini, sono caratterizzati dalla percentuale di conservante e dalle informazioni inerenti la provenienza (località vigneto, codice vigna e data di raccolta dell'uva). I vini rossi possono essere colorati mediante un procedimento chimico. Tra i vini bianchi si distinguono gli *spumanti*, caratterizzati dalla percentuale di gas. Gli oli, caratterizzati dalla viscosità, possono subire in qualunque momento l'operazione di aromatizzazione. Ogni vino bianco possiede un'etichetta in cui è riportato il codice a barre del prodotto, la data di imbottigliamento e quella di scadenza.

**A: analisi e disegno.**
Riportare il diagramma UML di tutte le classi che si ritengono necessarie alla corretta descrizione e rappresentazione della situazione sopra descritta. Nella gerarchia ereditaria si considerino anche i seguenti metodi eventualmente polimorfi e se ne dia un'implementazione:

- *getTotale*; restituisce il numero di prodotti esistenti fino all'istante di invocazione del metodo;
- *getPeso*; restituisce il peso del prodotto;
- *getEtichetta*; restituisce l'etichetta del prodotto;
- *getPrezzo*; restituisce il prezzo in euro per ogni prodotto calcolato mediante la formula:
  - $c \cdot Peso$ per gli oli,
  - $(c+Peso)^5 \cdot Conservante / 0.3$ per i vini rossi,
  - $52 \cdot c + 3 \cdot m$ per i vini bianchi,

  dove $c$ è un coefficiente costante per tutti i prodotti, ed $m$ è il numero di giorni compresi tra la data di raccolta dell'uva e la data di scadenza del prodotto.

**B: utilizzo delle classi.**
Si fornisca un frammento di programma che descriva la creazione e l'inizializzazione casuale di una collezione di 30 prodotti agricoli e

- si visualizzino le informazioni per ogni prodotto,
- si trovi il vino bianco più pesante,
- si ordinino (in base al prezzo) in un array indipendente tutti i vini della collezione (gli oggetti non devono essere clonati).

---------- Raccolta Terze Prove

Si consideri la seguente situazione che si vorrebbe modellare con un opportuno insieme di classi JAVA:

> Si vuole creare un database di oggetti geometrici 2D e 3D. Una figura geometrica è caratterizzata da un contenitore (char), ed è assegnata mediante le coordinate cartesiane dei suoi vertici. Ogni figura piana è caratterizzata da un nome (float), e quelle solide da un colore (byte). Si vogliono considerare triangoli, quadrati e pentagoni. Sui triangoli si può eseguire l'operazione di ricongiunzione, mentre sui pentagoni si può eseguire l'appiattimento. Per le figure piane considerate ha senso calcolare l'area, il perimetro, il numero di lati ed un coefficiente di variazione. Ogni figura solida contiene al proprio interno un dispositivo di equilibrio, caratterizzato dal numero seriale (20 caratteri), da un codice numerico (long) e da una memoria da 1KB.

**A: analisi e disegno.**
Riportare il diagramma UML di tutte le classi che si ritengono necessarie alla corretta descrizione e rappresentazione della situazione sopra descritta. Nella gerarchia ereditaria si considerino anche i seguenti metodi eventualmente polimorfi e se ne dia un'implementazione:

- *getTotale*; restituisce il numero di figure create fino all'istante di invocazione del metodo;
- *getContenitore*; restituisce il contenitore della figura;
- *getDispositivo*; restituisce il dispositivo di equilibrio;
- *getCoefficiente*; restituisce il coefficiente di variazione per ogni figura calcolato mediante la formula:
  - $c \cdot L$ per i triangoli,
  - $c\text{^}5 \cdot Area / 0.9$ per i quadrati,
  - $7 \cdot c + 3 \cdot Perimetro$ per i pentagoni,

  dove $c$ è un coefficiente costante per tutte le figure piane ed $L$ è il lato maggiore del triangolo.

**B: utilizzo delle classi.**
Si fornisca un frammento di programma che descriva la creazione e l'inizializzazione casuale di una collezione di 100 figure geometriche e

- si visualizzino le informazioni per ogni figura,
- si trovi la figura con il minimo(secondo l'ordinamento alfabetico) carattere contenitore,
- si ordinino (in base al coefficiente di variazione) in un array indipendente tutti i quadrati della collezione (gli oggetti non devono essere clonati).

---------- Raccolta Terze Prove

Si consideri la seguente situazione che si vorrebbe modellare con un opportuno insieme di classi JAVA:

> Si devono gestire informazioni inerenti auto, camion e barche. Le auto ed i camion sono caratterizzati dalla velocità in km/h. Camion e barche sono invece caratterizzati dal peso del carico. Auto, camion e barche hanno una data di immatricolazione; le prime caratterizzate dal numero di posti. Le barche possono essere riverniciate. Tra i camion si distinguono i furgoni, che contengono al proprio interno un dispositivo satellitare di sicurezza, caratterizzato dal numero seriale (20 caratteri), da un codice numerico (long) e da una memoria da 1KB.

**A: analisi e disegno.**
Riportare il diagramma UML di tutte le classi che si ritengono necessarie alla corretta descrizione e rappresentazione della situazione sopra descritta. Nella gerarchia ereditaria si considerino anche i seguenti metodi eventualmente polimorfi e se ne dia un'implementazione:

- *getTotale*; restituisce il numero di oggetti creati fino all'istante di invocazione del metodo;
- *getCarico*; restituisce il peso del carico;
- *getDispositivo*; restituisce il dispositivo satelllitare;
- *getPrezzo*; restituisce il prezzo in euro per ogni oggetto calcolato mediante la formula:
  - $c \cdot NumPosti$ per le auto,
  - $c\text{^}7 \cdot \Delta / 0.9$ per i camion,
  - $7 \cdot c + 3000$ per le barche,

  dove $c$ è un coefficiente costante per tutti e $\Delta$ è il numero di giorni trascorsi dalla immatricolazione ad oggi.

**B: utilizzo delle classi.**
Si fornisca un frammento di programma che descriva la creazione e l'inizializzazione casuale di una collezione di 100 oggetti e

- si visualizzino le informazioni per ogni oggetto,
- dire se esiste un auto a 5 posti che si muova ad una velocità di almeno 120km/h,
- aumentare di una percentuale variabile (3÷15%) il carico delle barche, e ordinare (mediante SelectionSort) in base al peso del carico, in un array indipendente, tutte le barche della collezione (gli oggetti non devono essere clonati).

---------- Raccolta Terze Prove

Si consideri la seguente situazione che si vorrebbe modellare con un opportuno insieme di classi JAVA:

Un'azienda produce serbatoi per acqua potabile: orizzontali (caratterizzati dalla larghezza) , verticali (caratterizzati dalla altezza) ed esterni (caratterizzati dal diametro). Tutti i serbatoi hanno una capacità totale ed una quantità di liquido contenuto.

I serbatoi orizzontali e quelli verticali possono essere interrati ed entrambe contengono una valvola di sfiato caratterizzata dal peso, da un codice alfanumerico a 5 cifre e da un coefficiente. Tra i serbatoi verticali si distinguono quelli termoisolati, caratterizzati dalla data di produzione, che subiscono periodicamente l'operazione di disinfezione.

**A: analisi e disegno.**

Riportare il diagramma UML di tutte le classi che si ritengono necessarie alla corretta descrizione e rappresentazione della situazione sopra descritta. Nella gerarchia ereditaria si considerino anche i seguenti metodi eventualmente polimorfi e se ne dia un'implementazione:

- *getTotale*; restituisce il numero di serbatoi creati fino all'istante di invocazione del metodo;
- *getCapacità*; restituisce la capacità totale del serbatoio;
- *getValvola*; restituisce la valvola di sfiato;
- *getPrezzo*; restituisce il prezzo in euro per ogni serbatoio calcolato mediante la formula:
  - $c \cdot Capacità$ per quelli da interro,
  - $c\text{^}7 \cdot \Delta / 0.6$ per quelli termoisolati,
  - $7 \cdot c + Diametro$ per quelli esterni,

  dove $c$ è un coefficiente costante per tutti i serbatoi e $\Delta$ è il numero di giorni di vita del serbatoio.

**B: utilizzo delle classi.**

Si fornisca un frammento di programma che descriva la creazione e l'inizializzazione casuale di una collezione di 1000 serbatoi e

- si visualizzino le informazioni per ogni serbatoio,
- dire se è possibile travasare tutto il liquido contenuto nei serbatoi da interro all'interno dei serbatoi esterni,
- diminuire di una percentuale variabile (3÷15%) il liquido presente nei serbatoi, e ordinare in base alla quantità di liquido, in un array indipendente, tutti i serbatoi della collezione (gli oggetti non devono essere clonati).

---------- Raccolta Terze Prove

Si consideri la seguente situazione che si vorrebbe modellare con un opportuno insieme di classi JAVA:

Una fabbrica produce camini in ghisa (caratterizzati dal numero di vetri), in muratura (caratterizzati dalla altezza della bocca) ed ornamentali (caratterizzati dal tipo di marmo). Tutti i camini hanno un volume espresso in $cm^3$.

I camini in ghisa e quelli in muratura sono usati per il riscaldamento e sono caratterizzati dalla quantità di calore fornito (in cal); entrambe presentano una canna fumaria caratterizzata dal diametro, dalla lunghezza e da un codice CEE ad 8 caratteri. Tra i camini in ghisa si distinguono quelli autopulenti, caratterizzati dalla data dell'ultimo controllo, che subiscono periodicamente l'operazione di svuotamento.

**A: analisi e disegno.**
Riportare il diagramma UML di tutte le classi che si ritengono necessarie alla corretta descrizione e rappresentazione della situazione sopra descritta. Nella gerarchia ereditaria si considerino anche i seguenti metodi eventualmente polimorfi e se ne dia un'implementazione:

- *getTotale*; restituisce il numero di camini creati fino all'istante di invocazione del metodo;
- *getVolume*; restituisce il volume del camino;
- *getCannaFumaria*; restituisce la canna fumaria;
- *getPrezzo*; restituisce il prezzo in euro per ogni camino calcolato mediante la formula:
  - $c \cdot DiametroCannaFumaria$ per quelli da riscaldamento,
  - $c^4 \cdot \Delta / 8.67$ per quelli autopulenti,
  - $5.2 \cdot c + Volume$ per quelli ornamentali,

  dove $c$ è un coefficiente costante per tutti i camini e $\Delta$ è il numero di giorni trascorsi dall'ultimo controllo.

**B: utilizzo delle classi.**
Si fornisca un frammento di programma che descriva la creazione e l'inizializzazione casuale di una collezione di 28 camini e

- si visualizzino le informazioni per ogni camino,
- dire se la quantità totale di calore fornito da tutti i camini in ghisa è maggiore o uguale di 13,8kcal,
- aumentare di una percentuale (variabile per ogni camino: 3÷15%) il volume dei camini, e ordinare in base ad esso tutti i camini della collezione (gli oggetti non devono essere clonati).

---------- Raccolta Terze Prove

Si consideri la seguente situazione che si vorrebbe modellare con un opportuno insieme di classi JAVA:

Un rivenditore ha in magazzino delle cinghie di trasmissione (caratterizzate dalla lunghezza in cm) di vario tipo: elastiche (caratterizzate dal coefficiente di stress), in fibra (caratterizzate dalla larghezza) e catene (caratterizzate dal numero di anelli).

Tutte le cinghie non elastiche sono rigide ed hanno un fattore di resistenza ed una etichetta in cui è indicata la data di produzione, il codice dello stabilimento di produzione ed il colore (5 char). Tra le cinghie in fibra si distinguono quelle rinforzate, caratterizzate dallo spessore dell'anima interna di acciaio, che possono subire periodicamente l'operazione di verifica di compattezza.

**A: analisi e disegno.**

Riportare il diagramma UML di tutte le classi che si ritengono necessarie alla corretta descrizione e rappresentazione della situazione sopra descritta. Nella gerarchia ereditaria si considerino anche i seguenti metodi eventualmente polimorfi e se ne dia un'implementazione:

- *getTotale*; restituisce il numero di cinghie create fino all'istante di invocazione del metodo;
- *getLunghezza*; restituisce la lunghezza della cinghia;
- *getEtichetta*; restituisce l'etichetta completa;
- *getPrezzo*; restituisce il prezzo in euro per ogni cinghia calcolato mediante la formula:
  - $c \cdot Spessore$ per quelle rinforzate,
  - $c^7 \cdot \Delta / 3.7$ per quelle rigide,
  - $2.4 \cdot c + CoeffStress$ per quelle elastiche,

  dove $c$ è un coefficiente costante per tutte le cinghie e $\Delta$ è il numero di giorni di vita della cinghia.

**B: utilizzo delle classi.**

Si fornisca un frammento di programma che descriva la creazione e l'inizializzazione casuale di una collezione di 33 cinghie e

- si visualizzino le informazioni per ogni cinghia,
- dire se disponendo su una linea retta tutte le cinghie rigide la lunghezza totale è non maggiore di 1000cm,
- aumentare di una percentuale (variabile per ogni cinghia: 2÷10%) il coefficiente di stress delle cinghie elastiche, e ordinare in base al prezzo tutte le cinghie della collezione (gli oggetti non devono essere clonati).

---------- Raccolta Terze Prove