

## Prodotti e Venditori (esercitazione)

Definire una classe **Prodotto** che permetta di rappresentare un generico prodotto da vendere. Tale classe deve contenere almeno: un intero rappresentante il codice del prodotto, ed un double rappresentante il prezzo (in euro).

Dotare, quindi, la classe dei seguenti metodi:

- Un costruttore senza parametri (per default, impostare tutti i dati a zero);
- Un costruttore con due parametri (il codice ed il prezzo);
- Metodi get e set per il codice ed il prezzo;
- L'overloading dell'operatore `operator<<`, che permetta di stampare su standard output le informazioni su un prodotto.

Può essere utile definire un metodo void `leggi()` che permetta di leggere da standard input le informazioni riguardanti un prodotto, in modo tale da creare a runtime delle istanze di Prodotto.

Estendere la classe prodotto definendo due sottoclassi:

- La classe `ProdottoAlimentare` aggiunge una data di scadenza ed un **metodo `verifica()` che restituisce true solo se il prodotto è ancora buono, altrimenti genera false se è scaduto.**
- La classe `ProdottoPerBambini` aggiunge la **fascia d'età** come attributo. Le fasce disponibili sono 0-3, 3-6, 6-9, 9-12.

Definire, quindi, una classe **Venditore**, che abbia come dati:

- Un intero, rappresentante il codice del venditore;
- Un array di lunghezza variabile contenente i prodotti, rappresentante l'elenco dei prodotti venduti;
- Un intero, rappresentante il numero di prodotti nell'array;

Dotare la classe dei seguenti metodi:

- Un costruttore senza parametri (per default, impostare tutti i dati a zero);
- Un costruttore con un parametro (il codice);
- Il costruttore di copia (facoltativo);
- L'overloading dell'operatore `operator<<`, che permetta di stampare su standard input le informazioni sul venditore: il codice e l'elenco dei prodotti venduti;
- Un metodo **double `calcolaPrezzoTotale()`** che restituisca la somma dei prezzi di tutti i prodotti del venditore;
- Un metodo **double `calcolaValoreMagazzino()`** che restituisca la somma dei prezzi dei prodotti del venditore escludendo quelli scaduti;
- Un metodo **void `aggiungiProdotto()`** che permetta di aggiungere un prodotto nell'elenco dei prodotti.

Anche in questo caso può essere utile un metodo `leggi()` per agevolare l'inserimento di un nuovo venditore.

Si noti che per poter aggiungere un prodotto è necessario che ci sia spazio sufficiente nell'array. Si noti, inoltre, che la classe `Venditore` si fa uso della composizione: al suo interno, infatti, utilizza la classe `Prodotto`. Per cui, dove necessario, è consigliato fare uso dei metodi della classe `Prodotto` nei metodi della classe `Venditore`. Ad esempio, nel metodo per la stampa della classe `Venditore` è bene utilizzare il metodo per la stampa della classe `Prodotto` per stampare su standard output i singoli prodotti.

Infine, realizzare un `main()` che crea un certo numero di venditori (almeno 3) ciascuno con almeno 5 prodotti, e permetta mediante un menu di stampare i venditori ed i loro prezzi totali in ordine crescente o decrescente a scelta dell'utente.

Alcuni suggerimenti:

- gestire la scadenza del prodotto in maniera semplice;
- organizzare i valori dei venditori e dei prodotti in dei file di testo, da leggere via codice mediante il metodo `leggi()`;
- per assicurarsi che i codici di venditori e prodotti siano univoci sfruttare le variabili statiche.