



## Programmazione ad Oggetti Traccia Laboratorio 8 novembre 2019

### Esercizio 1.

Utilizzando la classe Prodotto allegata alla traccia, realizzare una classe Supermercato che permetta di gestire i prodotti all'interno di un supermercato. In particolare, la classe deve contenere un campo privato `list<Prodotto*>` e i seguenti metodi:

- `void aggiungiProdotto(Prodotto*)`; aggiunge un prodotto all'interno del supermercato.
- `void rimuoviProdotto(Prodotto*)`; rimuove un prodotto inserito.
- `void stampaInventario() const`; stampa tutti i prodotti presenti nel supermercato.
- `void prodottiEconomici(int n) const`; stampa i primi `n` prodotti più economici.
- `void svuota()`; rimuove tutti i prodotti presenti nel supermercato.

Realizzare, inoltre, una classe GestoreProdotti che permetta di gestire i prodotti. In particolare, la classe deve contenere un campo privato `list<Prodotto*>` e i seguenti metodi:

- `bool aggiungiProdotto(int id, string descrizione, float prezzo)`; crea e aggiunge un prodotto. L'id del prodotto è univoco, quindi restituisce `false` se l'id è già stato inserito, `false` altrimenti.
- `bool modificaProdotto(int id, string descrizione, float prezzo)`; modifica il prodotto con un determinato id. Restituisce `false` se il prodotto non è presente, `true` altrimenti.
- `Prodotto* getProdotto(int id) const`; restituisce il prodotto con un determinato id, `nullptr` se il prodotto non è presente.

Infine, realizzare un main di prova che permetta l'interazione con un utente utilizzando un'interfaccia testuale. La schermata iniziale deve contenere la seguente interfaccia:

==== Menu ====

- Premi 1 per creare un prodotto
- Premi 2 per modificare un prodotto
- Premi 3 per aggiungere un prodotto all'interno del supermercato
- Premi 4 per rimuovere un prodotto dal supermercato
- Premi 5 per rimuovere tutti i prodotti dal supermercato
- Premi 6 per stampare tutti i prodotti nel supermercato
- Premi 7 per stampare i prodotti economici
- Premi 9 per uscire

CONTINUA A PAGINA 2.



### Esercizio 2.

Realizzare una classe che gestisca una rubrica telefonica usando i seguenti metodi:

- a) void aggiungiNumero(string nome, string numero); aggiunge un nuovo numero alla rubrica.
- b) string getNumero (string nome) const; restituisce il numero della persona. Se la persona non è presente nella rubrica, restituire "".
- c) void modificaNumero (string nome, string numero); modifica il numero di telefono di una persona.
- d) void stampa() const; stampa i numeri in rubrica.

### Esercizio 3.

Utilizzando la classe Esame allegata alla traccia, realizzare una classe Carriera che permetta di gestire la carriera di uno studente. La classe deve contenere un campo privato list<Esame\*> e i seguenti metodi:

- a) void aggiungi(Esame\*); aggiunge l'esame alla lista degli esami. L'esame è aggiunto solo se il voto è  $\geq 18$  e  $\leq 31$ , dove 31 rappresenta la lode.
- b) float mediaVoti() const; restituisce la media pesata sui crediti dei voti degli esami. Gli esami con lode vanno conteggiati con voto 31.
- c) void stampa() const; stampa tutti gli esami sostenuti dallo studente.
- d) void stampa(string professore) const; stampa tutti gli esami sostenuti da uno studente con un determinato professore.
- e) void stampaEsamiMigliori() const; stampa tutti gli esami con il voto più alto.
- f) void ordinaPerVoto(); ordina gli esami in base al voto decrescente.
- g) void ordinaPerNome(); ordina gli esami in base al nome dell'esame.
- h) bool borsaDiStudio() const; calcola se lo studente ha diritto alla borsa di studio. Uno studente ha diritto alla borsa di studio se ha sostenuto almeno 60 crediti e la sua media è maggiore o uguale di 25.