

Traccia di esempio per la seconda Prova di esonero

Creare un workspace avente nome *Esonero2Cognome*, e.g. *Esonero2Rossi*. In tale workspace si crei un progetto Java avente lo stesso nome del workspace.

Nel progetto creato si realizzino due package: *esonero2.store* ed *esonero2.test*.

Nel package *esonero2.store* si realizzino due classi che fungano da Magazzino di Prodotti.

In particolare:

- Magazzino di prodotti (di qualunque tipo essi siano);
- Magazzino di prodotti alimentari.

Classe Magazzino (valida per ogni classe Magazzino sopra elencata)

Rappresenta un contenitore di prodotti, può contenere istanze di prodotti ed è così definita:

- Attributi:

- prodotti: contenitore per memorizzare vari tipi di prodotti. **Non sono ammessi prodotti che abbiano lo stesso codice.**

- Operazioni:

1. boolean aggiungiProdotto(Prodotto prodotto): *true* se aggiunge un prodotto al contenitore, *false* altrimenti. Non c'è un limite al numero di prodotti.
2. boolean rimuoviProdottoByCodice(String codice): *true* se il prodotto, identificato mediante codice, è stato rimosso, *false* altrimenti.
3. Prodotti getLista(descrizione, filtro, operazione) dove:
 - descrizione: breve descrizione dell'operazione eseguita;
 - filtro: il filtro da applicare alla lista di elementi;
 - operazione: l'operazione da eseguire su ciascun elemento che soddisfa il filtro.
 - Prodotti: Collezione di prodotti contenente il risultato ottenuto a seguito dell'applicazione del *filtro* e dell'esecuzione dell'*operazione*.

Tramite l'applicazione di una specifica coppia (filtro, operazione) si dovranno ottenere:

- 3.1. lista di prodotti ordinata in ordine discendente per costo del prodotto (a partire dal più costoso);
- 3.2. lista di prodotti il cui costo sia inferiore ad un valore specificato;
- 3.3. lista di prodotti la cui descrizione inizi con un carattere specificato;
- 3.4. lista di prodotti che contenga il prodotto con il costo più elevato. Si precisa che tale lista conterrà al più un solo elemento.

Requisiti:

1. Per la realizzazione delle operazioni si utilizzino le API Streams e Lambdas.
2. Le operazioni di inserimento e rimozione di un prodotto saranno molto frequenti.

Qualunque tipo di prodotto sarà caratterizzato da:

- Attributi: codice (String), nome (String), costo (double)
- Operazioni:
 - Prodotto(String codice, String nome, double costo): costruttore per inizializzare gli attributi.
 - String getCodice(): restituisce il codice del prodotto.
 - String toString(): rappresentazione testuale di un prodotto.
 - double calcolaCosto(): calcola il costo del prodotto.

I prodotti alimentari si caratterizzano per avere:

- Attributi: energia (int), tempoConservazione (int).
- Metodi:
 - ProdottoAlimentare(String codice, String nome, double costo, int energia, int tempoConservazione): costruttore per inizializzare gli attributi.
 - double calcolaCosto(): il costo del prodotto è dimezzato se tempoConservazione è uguale o inferiore a 2.

Requisiti informativi dei prodotti

- Il *costo* non può mai essere negativo, se negativo il prodotto non può essere creato.
- *Energia* non può mai essere negativo, se negativo il prodotto alimentare non viene creato.
- Il *tempo di conservazione* è un valore almeno pari ad 1, se inferiore gli viene assegnato 1 (valore di default).
- L'attributo *codice*, obbligatorio per ogni prodotto, identifica univocamente un prodotto.

Esecuzione del TEST

Nel package *esonero2.test* si realizzi una classe di Test, denominata *MagazzinoTest*, che esegua le seguenti **ATTIVITÀ**:

1. Acquisizione, mediante inserimento da tastiera di tutti i prodotti i cui valori sono indicati nella sezione **INPUT** della presente traccia.
2. Aggiunta dei prodotti creati al punto 1. in un'istanza della classe *Magazzino* denominata *magazzino*. Rimozione dell'oggetto denominato *mela* da *magazzino* con stampa a video dell'esito. Aggiunta dell'oggetto denominato *mela* con stampa a video dell'esito.
3. Esecuzione delle operazioni 3.1, 3.2, 3.3 e 3.4. Per ogni esecuzione si producano le stampe dei risultati ottenuti dalle operazioni. Per le operazioni 3.2 e 3.3 i valori del costo e quello del carattere devono essere acquisiti da tastiera.

INPUT

1.	libro (nome della variabile - prodotto generico):	
	• Codice: G001; Nome: Libro; Costo: 20.0 euro	
2.	orologio (nome della variabile - prodotto generico):	
	• Codice: G002; Nome: Orologio; Costo: 100.0 euro	
3.	orologiodupl (nome della variabile - prodotto generico):	
	• Codice: G002 ; Nome: Orologio; Costo: 100.0 euro	
4.	mela (nome della variabile - prodotto alimentare):	Codice: A001; Nome: Mela; Costo: 1.0 euro; Energia: 40; Tempo di Conservazione: 7 giorni
5.	mela2 (nome della variabile - prodotto alimentare):	Codice: A001; Nome: Mela; Costo: 1.0 euro; Energia: -2 ; Tempo di Conservazione: 7 giorni
6.	pane (nome della variabile - prodotto alimentare):	Codice: A002; Nome: Pane; Costo: 2.0 euro; Energia: 80; Tempo di Conservazione: 2 giorni

VALUTAZIONE

Premessa

- Se il programma presenta errori di compilazione il punteggio finale sarà 0 a prescindere da quanto sia stato implementato.
- La valutazione di un'attività che solleva eccezione sarà 0.
- Se un'attività, pur non sollevando eccezione, non è soddisfatta completamente sarà valutata 0.

Punteggi

- Le attività 1 e 2 saranno valutate al massimo 1 punto ciascuna.
- L'attività 3 sarà valutata al massimo 4 punti. Ogni singola operazione testata da tale attività sarà valutata al massimo 1 punto.