

Esercizio 1

Si implementi in Java una classe *Sistema* che fornisca metodi per l'analisi di dati relativi alle merci in vendita in un esercizio commerciale e agli ordini di tali merci effettuate presso i loro fornitori. Si supponga che le classi *Merce* e *Fornitore* siano già disponibili e forniscano i seguenti metodi:

Classe *Merce*:

- *public String getNome()*, che restituisce il nome della merce (che si suppone essere identificativo della merce stessa).
- *public String getMarca()*, che restituisce il nome della marca produttrice della merce.
- *public LinkedList<String> getListaFornitori()*, che restituisce la lista dei nomi dei fornitori della merce.
- *public LinkedList<Float> getListaPrezzi()*, che restituisce la lista dei prezzi a cui viene fornita la merce (il prezzo presente nella posizione *i*-esima di tale lista è il prezzo UNITARIO a cui la merce viene fornita dal fornitore che si trova nella stessa posizione nella lista restituita da *getListaFornitori()*).
- *public boolean equals(Object o)*.
- *public String toString()*.

Classe *Ordine*:

- *public String getID()*, che restituisce il codice identificativo dell'ordine.
- *public String getMerce()*, che restituisce il nome della merce ordinata.
- *public int getQuantita()*, che restituisce il numero di unità ordinate della merce.
- *public String getFornitore()*, che restituisce il nome del fornitore al quale è stato indirizzato l'ordine.
- *public int getData()*, che restituisce un intero che codifica la data di effettuazione dell'ordine.
- *public boolean equals(Object o)*.
- *public String toString()*.

La classe *Sistema* contiene le liste delle merci e degli ordini effettuati. Oltre ad eventuali metodi che si ritengano utili, si includano nella classe almeno i seguenti metodi:

- *public boolean verificaMerciEOrdini()*. Il metodo restituisce *true* se e solo se:
 - a. per ogni merce, il numero di fornitori è uguale al numero di prezzi di fornitura e la marca non fa parte della lista dei suoi fornitori;
 - b. per ogni ordine, la merce indicata fa parte della lista delle merci e il fornitore al quale l'ordine è indirizzato è riportato nella lista dei fornitori della merce.
- *public LinkedList<String> merciSopraSoglia(float sogliaMinimo, float sogliaMedio)*. Il metodo restituisce la lista dei nomi delle merci fornite da almeno un fornitore e il cui prezzo minimo di fornitura supera strettamente il valore *sogliaMinimo* e il prezzo medio di fornitura supera strettamente il valore *sogliaMedio* (si restituisca un valore null se *verificaMerciEOrdini()* restituisce *false*).
- *public LinkedList<String> marcheOrdiniMassimi(int d1, int d2)*. Il metodo restituisce la lista delle marche delle merci presenti negli ordini che, tra gli ordini effettuati tra la data *d1* e la data *d2* (incluse), hanno importo massimo. L'importo di un ordine è dato dal prodotto della quantità di merce ordinata per il prezzo al quale il fornitore indicato fornisce la merce (si restituisca un valore null se *verificaMerciEOrdini()* restituisce *false*). La lista restituita deve contenere una stessa marca al più una volta.

Esempio. Si assuma che i dati a disposizione siano i seguenti:

- *Merci*:
 - {Nome "M1", Marca "X", ListaFornitori ["F1", "F2"], ListaPrezzi [1.1, 1.3] }
 - {Nome "M2", Marca "Y", ListaFornitori ["F1"], ListaPrezzi [2] }
 - {Nome "M3", Marca "X", ListaFornitori ["F1", "F2"], ListaPrezzi [1.4, 2] }

- Ordini:
 - {ID "ORD1", Merce "M1", Quantità 10, Fornitore "F1", Data 66 }
 - {ID "ORD2", Merce "M3", Quantità 5, Fornitore "F1", Data 100 }
 - {ID "ORD3", Merce "M3", Quantità 8, Fornitore "F2", Data 70 }

Allora:

- *verificaMerciEOrdini()* restituisce *true*. Avrebbe restituito *false*, ad esempio, se la lista di fornitori di M1 fosse stata ["F1","F2","F3"] e quella dei prezzi [1.1, 1.3], o se la lista di fornitori di M2 avesse contenuto Y (ossia, la marca di M2), o se l'ordine ORD1 avesse riguardato la merce M4 (non presente nella lista delle merci), o se nell'ordine ORD1 fosse stato specificato il fornitore F3, che non fa parte dei fornitori di M1.
- *merciSopraSoglia(1.3, 1.5)* restituisce la lista ["M2", "M3"] perchè M2 ha come prezzi minimo e medio di fornitura il valore 2 (che è strettamente maggiore sia di 1.3 che di 1.5), ed M3 ha come prezzo minimo di fornitura il valore 1.4, che è maggiore di 1.3, e prezzo medio 1.7 (dato da $(1.4+2)/2$), che è maggiore di 1.5. La merce M1 non soddisfa invece nè il requisito sul prezzo minimo che sul prezzo medio.
- *marcheOrdiniMassimi(50,80)* restituisce la lista ["X"] perché è la marca della merce associata all'unico ordine di importo massimo. Gli unici ordini che ricadono nella finestra temporale [50..80] sono ORD1 e ORD3, e l'importo di ORD1 è $1.1 \times 10 = 11$ mentre quello di ORD3 è $8 \times 2 = 16$.

Esercizio 2

Si arricchisca la classe *ListaConcatenataInt* sviluppata durante il corso con un metodo *boolean verificaOrdinamento()* che verifica se la lista è formata da sequenze di valori positivi e negativi che soddisfano le seguenti condizioni:

- tutte le sequenze di valori positivi sono ordinate in modo crescente;
- tutte le sequenze di valori negativi sono ordinate in modo decrescente.

Ad esempio, la lista [2, 3, 7, -1, -3, -4, -5, 8, 11, 20] soddisfa la condizione, mentre la lista [2, 3, 7, -1, -3, -2] non la soddisfa. Il metodo *verificaOrdinamento* dovrà essere ricorsivo o invocare un opportuno metodo ricorsivo sulla classe *NodoInt*.