Corso di Laurea in Ingegneria Informatica

Prova scritta di *Fondamenti di informatica II*

17 febbraio 2021

**Tempo a disposizione: 1 ora e 10 minuti**

**Esercizio 1**

Si implementi in Java una classe *Sistema* che fornisca metodi per l’analisi dei dati riguardanti articoli costruiti assemblando componenti. Si supponga che le classi *Componente* e *Articolo* siano già disponibili e forniscano i seguenti metodi:

Classe *Componente*:

* *public String getCodice()*, che restituisce il codice che identifica il componente.
* *public float getPrezzo()*, che restituisce il prezzo di acquisto del componente.
* *public boolean equals(Object o)*
* *public String toString()*

Classe *Articolo*:

* *public String getNome()*, che restituisce il nome che identifica l’articolo.
* *public float getPrezzo()*, che restituisce il prezzo di vendita dell’articolo.
* *public LinkedList<String> getComponenti()*, che restituisce la lista dei codici dei componenti utilizzati per la costruzione dell’articolo.
* *public boolean equals(Object o)*
* *public String toString()*

Il *profitto* derivante da un articolo è dato dalla differenza tra il prezzo di vendita dell’articolo stesso e la somma dei prezzi dei suoi componenti.

La classe *Sistema* contiene le liste dei componenti e degli articoli. Oltre ad eventuali metodi che si ritengano necessari, si includano almeno i seguenti metodi nella classe:

* *public String articoloTop().* Il metodo restituisce il nome dell’articolo dal quale deriva il massimo profitto. Qualora la proprietà fosse soddisfatta da più articoli, il metodo restituisce uno qualsiasi di essi.
* *public LinkedList<String> componentiUniversali().* Il metodo restituisce la lista dei codici dei componenti utilizzati in tutti gli articoli.
* *public LinkedList<String>**articoliComponentiCostosi(float p).* Il metodo restituisce la lista dei nomi degli articoli tra i cui componenti ne è presente almeno uno con prezzo maggiore di *p*.

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

* Componenti:
  + {“C1”, 100.0}
  + {“C2”, 200.0}
  + {“C3”, 50.0}
  + {“C4”, 100.0}
* Articoli:
  + {“Articolo A”, 210.0, [“C1”, “C4”] }
  + {“Articolo B”, 400.0, [“C1”, “C2”, “C3”] }
  + {“Articolo C”, 370.0, [“C1, “C2”,“C3”] }

Allora:

* *articoloTop()* restituisce “Articolo B” perchè da tale articolo deriva il massimo profitto (50.0).
* *componentiUniversali()* restituisce la lista [“C1”] perché il solo componente “C1” è utilizzato in tutti gli articoli.
* *articoliComponentiCostosi(150.0)* restituisce la lista [“Articolo B”, “Articolo C”] perché tra i componenti di tali articoli è presente “C2”, il cui prezzo è maggiore di 150.

**Esercizio 2**

Si arricchisca la classe *ListaConcatenataInt* sviluppata durante il corso con un metodo *verificaUgualiConsecutivi(int x)* che restituisca *true* se e solo se lista contiene almeno *x* volte 2 elementi uguali consecutivi. Si assuma che la lista non contenga 3 elementi uguali consecutivi. Il metodo *verificaUgualiConsecutivi* dovrà essere ricorsivo o invocare un opportuno metodo ricorsivo sulla classe *NodoInt*.