

Esercizio 1

Si implementi in Java una classe *Sistema* che fornisca metodi per l'analisi di piatti preparati presso un ristorante e degli ingredienti in essi utilizzati. Si supponga che siano definite le classi *Piatto* e *Ingrediente*, che forniscono i seguenti metodi:

Classe *Piatto*:

- *public String getCodice()*, che restituisce il codice identificativo del piatto.
- *public String getNome()*, che restituisce il nome del piatto.
- *public LinkedList<String> getIngredienti()*, che restituisce la lista dei codici degli ingredienti utilizzati nella preparazione del piatto.
- *public LinkedList<Integer> getDosi()*, che restituisce la lista delle dosi (esprese in grammi) degli ingredienti. In particolare, l'intero in posizione *i* della lista restituita da *getDosi()* è il numero di grammi utilizzati dell'ingrediente il cui codice è nella posizione *i* della lista restituita da *getIngredienti()*.
- *public boolean equals(Object o)*.
- *public String toString()*.

Classe *Ingrediente*:

- *public String getCodice()*, che restituisce il codice identificativo dell'ingrediente.
- *public String getNome()*, che restituisce il nome dell'ingrediente.
- *public float getCostoUnitario()*, che restituisce il costo (al grammo) dell'ingrediente.
- *public boolean adattoVegetariani()*, che restituisce *true* se e solo se l'uso dell'ingrediente è consentito nella dieta vegetariana.
- *public boolean equals(Object o)*.
- *public String toString()*.

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

- *public boolean verificaPiattiVegetariani(int n)*. Il metodo restituisce *true* se e solo se il numero di piatti adatti ai vegetariani è almeno *n*. Un piatto è adatto ai vegetariani se tutti i suoi ingredienti lo sono.
- *public LinkedList<String> piattiConIngrediente(String nomeIng, int k)*. Il metodo restituisce la lista dei nomi dei piatti che contengono una dose dell'ingrediente di nome *nomeIng* pari ad almeno *k* grammi.
- *public LinkedList<String> piattiCostosi(float n)*. Il metodo restituisce la lista dei nomi dei piatti il cui costo è almeno *n*. Il costo di un piatto è pari alla somma dei costi pagati per ciascun ingrediente, dove il costo per ciascun ingrediente è uguale al prodotto tra la dose nel piatto e il costo unitario dell'ingrediente.

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

- *listaPiatti*:
 - {"P1", "Carbonara", ["I1", "I2"], [20, 30]}
 - {"P2", "Fagiolata", ["I3", "I4"], [60, 50]}
- *listaIngredienti*:
 - {"I1", "Uovo", 0.5, *false*}
 - {"I2", "Pancetta", 0.8, *false*}
 - {"I3", "Fagioli", 0.3, *true*}
 - {"I4", "Pomodoro", 0.1, *true*}

Allora:

- *verificaPiattiVegetariani(2)* restituisce *false* perché solo il piatto “Fagiolata” è adatto ai vegetariani, mentre “Carbonara” non lo è (in quanto contiene almeno un ingrediente non adatto ai vegetariani).
- *piattiConIngrediente(“Pancetta”, 10)* restituisce la lista [“Carbonara”] in quanto “Carbonara” è il nome dell’unico piatto che contiene almeno 10 grammi di pancetta.
- *piattiCostosi(30.0)* restituisce la lista [“Carbonara”] in quanto il solo piatto che ha un costo superiore a 30.0 è il piatto di nome “Carbonara” (il cui costo è 34.0, ottenuto come $20 \times 0.5 + 30 \times 0.8$, che è la somma dei contributi al costo dovuti alle dosi di uova e pancetta). Il piatto di nome “Fagiolata” ha invece un costo inferiore a 30, poiché tale costo è $60 \times 0.3 + 50 \times 0.1 = 23$.

Esercizio 2

Si arricchisca la classe *ListaConcatenataInt* sviluppata durante il corso con un metodo *contaTerne* che restituisce quante volte nella lista accade che l’elemento in posizione *i* sia pari alla somma algebrica degli elementi nelle posizioni *i-1* e *i+1*. Si noti che, se la lista non ha almeno 3 elementi, allora il metodo *contaTerne* restituisce 0. Il metodo *contaTerne* dovrà essere ricorsivo o invocare un opportuno metodo ricorsivo sulla classe *NodoInt*.

Esempio. Se la lista è [12, 20, 8, 3, 2, 4, 2, -2], allora il metodo restituisce 3, in quanto:

- l’elemento in posizione *i=1* (20) è pari alla somma algebrica degli elementi nelle posizioni *i-1* e *i+1* (12+8);
- l’elemento in posizione *i=5* (4) è pari alla somma algebrica degli elementi nelle posizioni *i-1* e *i+1* (2+2);
- l’elemento in posizione *i=6* (2) è pari alla somma algebrica degli elementi nelle posizioni *i-1* e *i+1* (4-2);
- nessun altro elemento soddisfa la condizione.