

Esercizio 1

Si implementi in Java una classe *Sistema* che fornisca metodi per l'analisi delle vendite effettuate nell'ambito di aste di opere d'arte. Si supponga che siano definite le classi *Opera* e *Vendita*, che forniscono i seguenti metodi:

Classe *Opera*:

- *public String getNome()*, che restituisce il nome che identifica l'opera d'arte;
- *public String getAutore()*, che restituisce l'autore dell'opera;
- *public int getAnno()*, che restituisce l'anno in cui è stata creata l'opera;
- *public boolean equals(Object o)*;
- *public String toString()*.

Classe *Vendita*:

- *public String getID()*, che restituisce il codice identificativo della vendita;
- *public String getOpera()*, che restituisce il nome dell'opera venduta tramite l'asta;
- *public LinkedList<Integer> getOfferte()*, che restituisce la sequenza delle offerte registrate durante l'asta per l'opera venduta;
- *public LinkedList<String> getOfferenti()*, che restituisce la lista dei nomi delle persone che hanno effettuato le offerte (ossia: la posizione *i* di *getOfferenti()* contiene il nome della persona che ha effettuato l'offerta il cui importo è nella posizione *i* della lista restituita da *getOfferte()*);
- *public boolean equals(Object o)*;
- *public String toString()*.

SI NOTI CHE, IN OGNI VENDITA, L'ACQUIRENTE CHE SI È AGGIUDICATO L'OPERA È QUELLO IL CUI NOME È IN ULTIMA POSIZIONE NELLA LISTA RESTITUITA DAL METODO *getOfferenti()*.

La classe *Sistema* contiene le liste ***listaOpere*** delle opere d'arte e ***listaVendite*** delle vendite effettuate tramite asta. Oltre ad eventuali metodi che si ritengano necessari, si includano almeno i seguenti metodi nella classe:

- *public boolean consistente(int maxOfferteIndividuali)*, che restituisce *true* se e solo se, in ogni vendita, sono soddisfatte le seguenti condizioni:
 - a. le liste di offerte e di offerenti hanno la stessa lunghezza;
 - b. le offerte sono in ordine strettamente crescente;
 - c. non esistono offerte consecutive della stessa persona;
 - d. nessuna persona ha fatto un numero di offerte superiore a *maxOfferteIndividuali*.
- *public LinkedList<String> acquirentiSimili(String nomeAcquirente)*. Il metodo restituisce la lista dei nomi distinti degli acquirenti che si sono aggiudicati almeno un'opera di un autore che ha prodotto un'opera che è stata aggiudicata a *nomeAcquirente* (gli acquirenti restituiti non devono comprendere *nomeAcquirente*).
- *public int maxRilancio(String autore)*. Il metodo restituisce l'importo del massimo rilancio che è avvenuto tra tutte le vendite che hanno coinvolto opere prodotte da *autore*. Si noti che un rilancio è la differenza tra due offerte consecutive registrate nella stessa vendita. In assenza di rilanci, il metodo restituisce 0.

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

- *listaOpere*:
 - {"Il sole", "Mutandari", 2000}
 - {"La luna", "Fragolari", 2001}
 - {"Inquietudine", "Fragolari", 2001}

- listaVendite:
 - {"V1", "Il sole", [1000, 1100, 1200], ["Lino", "Anna", "Lino"]}
 - {"V2", "La luna", [600, 1100, 1300, 1350, 1400], ["Lino", "Anna", "Lino", "Giorgia", "Lino"]}
 - {"V3", "Inquietudine", [800], ["Giorgia"]}

Allora:

- *consistente(4)* restituisce *true* (in quanto sono verificate le condizioni *a*, *b*, *c*, *d* su tutte le vendite), mentre *consistente(2)* restituisce *false* (in quanto non è verificata la condizione *d* sulla vendita "V2", nella quale Lino ha effettuato 3 offerte);
- *acquirentiSimili("Lino")* restituisce la lista ["Giorgia"], in quanto Giorgia si è aggiudicata l'opera "Inquietudine" di Fragolari, che è l'autore di una delle opere che si è aggiudicato Lino (in particolare, l'opera "La luna");
- *maxRilancio("Fragolari")* restituisce 500, che è il massimo rilancio registrato per le opere di Fragolari, in particolare in occasione della vendita "V2".

Esercizio 2

Si arricchisca la classe *ListaConcatenataInt* sviluppata durante il corso con un metodo *verificaCoppie* che verifica se la somma dei valori in posizione *i-1* e *i* nella lista è sempre la metà della somma dei valori in posizione *i* e *i+1*. Il metodo *verificaCoppie* dovrà essere ricorsivo o invocare un opportuno metodo ricorsivo sulla classe *NodoInt*.

Esempio. La lista [2, 3, 7, 13, 27, 53] soddisfa la condizione, in quanto:

- $2+3 = (3+7) / 2$
- $3+7 = (7+13) / 2$
- $7+13 = (13+27) / 2$
- $13+27 = (27+53) / 2$

La lista [2, 3, 5, 10, 28] non soddisfa invece la condizione, in quanto $2+3 > (3+5) / 2$.