

Esercizio 1

Si implementi in Java una classe *Sistema* che fornisca metodi per l'analisi di dati relativi ad acquisti e noleggi di film su una piattaforma di streaming. Si supponga che le classi *Film* e *Cliente* siano già disponibili e forniscano i seguenti metodi:

Classe *Film*:

- *public String getTitolo()*, che restituisce il titolo che identifica il film.
- *public String getRegista()*, che restituisce il nome del regista del film.
- *public int getPrezzoAcquisto()*, che restituisce il prezzo di acquisto del film.
- *public int getPrezzoNoleggio()*, che restituisce il prezzo del noleggio del film.
- *public boolean equals(Object o)*.
- *public String toString()*.

Classe *Cliente*:

- *public String getNome()*, che restituisce il nome che identifica il cliente.
- *public LinkedList<String> getTitoli()*, che restituisce la lista dei titoli dei film che il cliente ha acquistato o noleggiato.
- *public LinkedList<String> getModalita()*, che restituisce la lista delle modalità utilizzate dal cliente: l'*i*-esimo elemento della lista contiene
 - "acquisto" se l'*i*-esimo film della lista restituita dal metodo *getTitoli* è stato acquistato;
 - "noleggio" se l'*i*-esimo film della lista restituita dal metodo *getTitoli* è stato noleggiato.
- *public LinkedList<Integer> getDate()*, che restituisce la lista delle date (rappresentate mediante interi) in cui il cliente ha effettuato acquisti o noleggi: l'*i*-esimo elemento della lista contiene la data in cui il cliente ha effettuato l'operazione di acquisto o noleggio dell'*i*-esimo film della lista restituita dal metodo *getTitoli*.
- *public boolean equals(Object o)*.
- *public String toString()*.

La classe *Sistema* contiene le liste dei film e dei clienti. Oltre ad eventuali metodi che si ritengano utili, si includano nella classe almeno i seguenti metodi:

- *public String clienteMax()*. Il metodo restituisce il nome del cliente che ha speso la massima cifra totale. La cifra totale spesa da un cliente deve ovviamente essere calcolata considerando, per ogni film che il cliente ha acquistato o noleggiato, il relativo prezzo. Se più di un cliente soddisfa la condizione, il metodo restituisce uno qualsiasi tra di essi.
- *public boolean verificaDati(String c)*. Il metodo restituisce *true* se e solo se l'insieme delle operazioni fatte dal cliente con nome *c* soddisfa le seguenti condizioni:
 - *c* non ha acquistato più di una volta uno stesso film;
 - per ogni film *f* noleggiato da *c* più di una volta, le date in cui *c* ha noleggiato *f* sono tutte diverse;
- *public String registaApprezzato(int d1, int d2)*. Il metodo restituisce il nome del regista i cui film, tra la data *d1* e la data *d2* (incluse), sono stati acquistati o noleggiati dal maggior numero di clienti diversi. Se più di un regista soddisfa la condizione, il metodo restituisce uno qualsiasi di essi.

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

- Film:
 - {Titolo "Film A", regista "Rossi", prezzo acquisto 10, prezzo noleggio 5}
 - {Titolo "Film B", regista "Verdi", prezzo acquisto 8, prezzo noleggio 4}
 - {Titolo "Film C", regista "Rossi", prezzo acquisto 10, prezzo noleggio 5}
 - {Titolo "Film D", regista "Verdi", prezzo acquisto 12, prezzo noleggio 6}

- Clienti:
 - {Nome "Mario", titoli ["Film B", "Film D"], modalità ["acquisto", "acquisto"], date [10,20]}
 - {Nome "Luigi", titoli ["Film B", "Film A"], modalità ["noleggio", "acquisto"], date [10,20]}
 - {Nome "Anna", titoli ["Film B", "Film A"], modalità ["acquisto", "noleggio"], date [15,30]}
 - {Nome "Lucia", titoli ["Film C", "Film C"], modalità ["noleggio", "noleggio"], date [5,25]}

Allora:

- *clienteMax()* restituisce "Mario" perché il cliente "Mario" ha speso la massima cifra totale (20).
- *verificaDati("Lucia")* restituisce *true* perché il cliente "Lucia" non ha acquistato più di una volta uno stesso film e per ogni film noleggiato da "Lucia" più di una volta ("Film C"), le date in cui "Lucia" ha noleggiato il film sono tutte diverse.
- *registraApprezzato(10, 20)* restituisce "Verdi" perché i film del regista "Verdi", tra la data 10 e la data 20 (incluse), sono stati acquistati o noleggiati dal maggior numero di clienti diversi (3).

Esercizio 2

Si arricchisca la classe *ListaConcatenataInt* sviluppata durante il corso con un metodo *boolean listaEquilibrata()* che restituisce *true* se e solo se sono vere entrambe le seguenti condizioni:

- il numero di elementi maggiori di zero è uguale al numero di elementi minori di zero;
- la somma di tutti gli elementi è pari a zero.

Ad esempio, se la lista contiene i valori [5,-3,-4,2,0,-4,4], allora il metodo restituisce *true* perché:

- la lista contiene 3 elementi maggiori di zero e 3 elementi minori di zero;
- la somma di tutti gli elementi è pari a zero.

Il metodo *listaEquilibrata* dovrà essere ricorsivo o invocare un opportuno metodo ricorsivo sulla classe *NodoInt*.