

Esercizio 1

Si implementi in Java una classe *Sistema* che fornisca metodi per l'analisi dei dati riguardanti corsi e docenti. Si supponga che siano definite le classi *Corso* e *Docente*, che forniscono i seguenti metodi:

Classe *Corso*:

- *public String getCodice()*, che restituisce il codice che identifica il corso.
- *public String getTitolo()*, che restituisce il titolo del corso.
- *public int getDurata()*, che restituisce la durata (in ore) del corso.
- *public int getNumStudenti()*, che restituisce il numero di studenti che seguono il corso.
- *public boolean equals(Object o)*.
- *public String toString()*.

Classe *Docente*:

- *public String getNome()*, che restituisce il nome che identifica il docente.
- *public LinkedList<String> getCorsi()*, che restituisce la lista dei codici dei corsi tenuti dal docente.
- *public boolean equals(Object o)*.
- *public String toString()*.

La classe *Sistema* contiene le liste *listaCorsi* dei corsi e *listaDocenti* dei docenti. Oltre ad eventuali metodi che si ritengano necessari, si includano almeno i seguenti metodi nella classe:

- *public boolean verifica(int caricoMin, int studentiMax)*, che restituisce *true* se e solo se sono soddisfatte tutte e tre le seguenti condizioni:
 - a. nessun docente ha un carico inferiore a *caricoMin* (il carico di un docente è la somma delle durate dei corsi tenuti dal docente stesso);
 - b. nessun corso ha un numero di studenti maggiore di *studentiMax*;
 - c. nessun corso è tenuto da più di un docente.
- *public LinkedList<String> docentiSimili(String docente)*. Il metodo restituisce la lista dei nomi distinti dei docenti che tengono almeno un corso avente titolo uguale ad uno dei corsi tenuti dal docente con nome *docente*.
- *public LinkedList<String> corsiCritici()*. Il metodo restituisce la lista dei codici dei corsi "critici"; un corso *c* è critico se soddisfa le seguenti condizioni:
 - a. esistono almeno altri 2 corsi con lo stesso titolo di *c*;
 - b. il numero totale di studenti che seguono corsi aventi lo stesso titolo di *c* (incluso *c* stesso) è maggiore di 300.

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

- Corsi:
 - {"C1", "Fondamenti di informatica 1", 57, 140}
 - {"C2", "Fondamenti di informatica 1", 57, 140}
 - {"C3", "Fondamenti di informatica 2", 38, 140}
 - {"C4", "Fondamenti di informatica 2", 38, 140}
 - {"C5", "Analisi 1", 30, 100}
 - {"C6", "Fondamenti di informatica 1", 30, 100}
- Docenti:
 - {"Rossi", ["C1", "C3", "C6"]}

- {"Bianchi", ["C2"]}
- {"Verdi", ["C4", "C5"]}

Allora:

- *verifica(50, 140)* restituisce *true* perché:
 - nessun docente ha un carico inferiore a 50 (il carico di Rossi è 125, quello di Bianchi è 57 e quello di Verdi è 68);
 - nessun corso ha un numero di studenti maggiore di 140;
 - nessun corso è tenuto da due docenti diversi.
- *docentiSimili("Verdi")* restituisce la lista ["Rossi"], in quanto Rossi tiene il corso con codice "C3" avente titolo uguale al corso con codice "C4" tenuto da Verdi.
- *corsiCritici()* restituisce la lista ["C1", "C2", "C6"] perché ognuno dei corsi con codici "C1", "C2" e "C6" ha:
 - altri 2 corsi con lo stesso titolo;
 - numero totale di studenti che seguono corsi aventi lo stesso titolo maggiore di 300 (ossia 380).

Esercizio 2

Si arricchisca la classe *ListaConcatenataInt* sviluppata durante il corso con un metodo *verificaCoppie* che restituisce *true* se e solo se i valori in posizione *i* e *i+1* (con *i* pari) sono entrambi minori o uguali ai valori in posizione *i+2* e *i+3* (se esistono). Si assuma che la lista non sia vuota e che la sua lunghezza sia pari. Il metodo *verificaCoppie* dovrà essere ricorsivo o invocare un opportuno metodo ricorsivo sulla classe *NodoInt*.

Esempio. Se la lista contiene i valori [1, 2, 5, 4, 5, 5, 6, 5] allora la funzione restituisce *true*, perché:

- 1 e 2 sono minori o uguali a 5 e 4;
- 5 e 4 sono minori o uguali a 5 e 5;
- 5 e 5 sono minori o uguali a 6 e 5.

Se la lista contiene i valori [1, 2, 5, 4, 3, 7] allora la funzione restituisce *false*, perché 5 è maggiore di 3.