Corso di Laurea in Ingegneria Informatica
Prova scritta di *Fondamenti di informatica II*
16 febbraio 2024
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 viaggi in autobus e conducenti. Si supponga che siano definite le classi *Viaggio* e *Conducente*, che forniscono i seguenti metodi:

Classe *Viaggio*:

- *public String getCodice()*, che restituisce il codice che identifica il viaggio;
- *public String getDestinazione()*, che restituisce la destinazione del viaggio;
- *public int getDurata(),* che restituisce la durata (in ore) del viaggio;
- *public int getNumPasseggeri(),* che restituisce il numero di passeggeri che partecipano al viaggio;
- *public boolean equals(Object o)*;
- *public String toString().*

Classe *Conducente*:

- *public String getNome()*, che restituisce il nome che identifica il conducente;
- *public LinkedList<String> getViaggi()*, che restituisce la lista dei codici dei viaggi gestiti dal conducente;
- *public boolean equals(Object o);*
- *public String toString().*

La classe *Sistema* contiene le liste *listaViaggi* dei viaggi e *listaConducenti* dei conducenti. Oltre ad eventuali metodi che si ritengano necessari, si includano almeno i seguenti metodi nella classe:

- *public boolean verifica(int numMax, String nomeConducente),* che restituisce *true* se e solo se per ogni destinazione esistono al più *numMax* viaggi il cui conducente si chiama *nomeConducente*;
- *public LinkedList<String> conducentiSovraccarichi().* Il metodo restituisce la lista dei nomi dei conducenti che risultano sovraccarichi, ossia che hanno effettuato un numero di viaggi uguale almeno al doppio del numero di viaggi medio per conducente.
- *public LinkedList<String> stesseCaratteristiche (String nomeConducente).* Il metodo restituisce la lista dei nomi distinti dei conducenti tali che la durata complessiva e il numero di passeggeri complessivo dei loro viaggi sono gli stessi che caratterizzano il conducente di nome *nomeConducente*.

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

- Viaggi:
  - {"V1", "Roma", 2, 5}
  - {"V2", "Roma", 8, 20}
  - {"V3", "Salerno", 2, 5}
  - {"V4", "Salerno", 4, 70}
  - {"V5", "Reggio Calabria", 3, 5}
  - {"V6", "Roma", 1, 5}

- Conducenti:
  - {"Rossi", ["V1", "V3", "V5", "V6"]}
  - {"Bianchi", ["V2"]}
  - {"Verdi", ["V4 "]}

Allora:

- *verifica(1, "Rossi")* restituisce *false* perché per la destinazione Roma esistono 2 viaggi il cui conducente è "Rossi";
- *conducentiSovraccarichi( )* restituisce la lista ["Rossi"] perché "Rossi" ha compiuto 4 viaggi, che è il doppio del valore medio di viaggi per conducente (che è dato da (4+1+1)/3=2);

- *stesseCaratteristiche("Bianchi")* restituisce la lista *["Rossi"]* perché:
    - Bianchi ha effettuato il solo viaggio "V2", per cui la durata complessiva e il numero di passeggeri complessivo coincidono con quelli del viaggio "V2", ossia:
        i. durataComplessiva=8; numPasseggeriComplessivo=20;
    - Rossi ha effettuato i viaggi i viaggi "V1", "V3", "V5", "V6", per cui la durata complessiva e il numero di passeggeri complessivo coincidono con quelli del viaggio "V2" di Bianchi, ossia:
        i. durataComplessiva=2+2+3+1=8; numPasseggeriComplessivo=5+5+5+5= 20;
    - Verdi ha effettuato il solo viaggio "V4", che ha le caratteristiche in esame diverse dal viaggio "V2" di Bianchi.

Esercizio 2

Si arricchisca la classe *ListaConcatenataInt* sviluppata durante il corso con un metodo *verifica(int lungMin, int n)* che restituisce *true* se è solo se la lista **non** contiene il valore *n* ed è composta da sequenze di lunghezza **strettamente maggiore** di *lungMin* composte da numeri **tutti** minori di *n* o **tutti** maggiori di *n*.
Il metodo *verifica* dovrà essere ricorsivo o invocare un opportuno metodo ricorsivo sulla classe *NodoInt*.

Esempio. Se la lista contiene i valori [6, 7,6, 2, 1, 0, 1, 9,9, 9] allora *verifica (2, 5)* restituisce *true*, perché la lista è la concatenazione delle sottoliste [6,7,6] (che ha lunghezza maggiore di 2 e contiene solo numeri maggiori di 5), [2,1,0,1] (che ha lunghezza maggiore di 2 e contiene solo numeri minori di 5), e [9,9,9] (che ha lunghezza maggiore di 2 e contiene solo numeri maggiori di 5).
Al contrario, se la lista contiene i valori [6,7,6,2,1,9,9,9], allora *verifica (2, 5)* restituisce *false,* perché la sottosequenza [2,1] di numeri non maggiori di 5 non ha lunghezza superiore a 2.