

FONDAMENTI DI INFORMATICA T-1 (L-Z)

Ing. Gestionale – Prof. Penzo
Preparazione alla prova d'esame

ESERCIZIO 1

Si scriva un programma Java che definisca opportuni classi, costruttori e metodi per la distribuzione automatica di biglietti per il trasporto. I dati gestiti dal programma sono relativi a:

- Percorsi, costituiti dalle informazioni: codice di percorso (stringa univoca), elenco di fermate (lista di informazioni: nome fermata e distanza in km dalla fermata di partenza).
- Mezzi, caratterizzati dal codice del mezzo e dal codice del percorso coperto dal mezzo.

I mezzi si suddividono in mezzi urbani ed extraurbani (treni). I codici dei mezzi vengono generati automaticamente dal sistema secondo il formato: U-#progressivo per gli urbani, E-#progressivo per gli extraurbani, con numerazione progressiva indipendente per U e per E. Per i mezzi urbani si mantiene l'informazione se siano autobus o convogli metro, per i treni il tipo di treno (ETR, IC e Regionale).

Si supponga che tali informazioni siano contenute in una classe *Biglietteria* nelle strutture dati:

- un vettore *percorsi*, contenente i percorsi disponibili sulla rete (massimo 10000).
- una lista *mezzi*, contenente tutti i mezzi a disposizione sul territorio.
- una mappa *listino*, che elenca il costo espresso in euro differenziato per i treni ETR, IC e Regionale in base ai km di percorrenza (mappa km → costi).

Si assuma che le strutture dati siano già inizializzate con un numero imprecisato di elementi. Si definiscano i soli metodi getter e setter necessari all'esecuzione del programma. Si utilizzi la programmazione generica. Si definiscano i seguenti metodi, all'interno dei quali *non è consentita l'immissione di valori da standard input (escluso il programma principale o se diversamente specificato)* (N.B.: **svolgere tutti i punti nel modo più efficiente**):

1) metodo *trovaFermata* che, dato il nome di una fermata in un percorso, restituisca la fermata corrispondente. Gestire il caso di assenza della fermata riportando a video un messaggio di errore.

2) metodo *codiciPercorsiRilevanti* che, dato il nome di una fermata di partenza e il nome di una fermata di arrivo, restituisca un set contenente i codici dei percorsi che contengono entrambe le fermate indicate e in quell'ordine. Il metodo deve utilizzare il metodo del punto 1) per individuare le fermate nei percorsi. Attenzione al caso in cui le fermate non vengono trovate entrambe nello stesso percorso.

3) metodo *calcolaResto* che, dato un mezzo extraurbano, un chilometraggio da percorrere e un importo espresso in euro, determini e restituisca il resto da dare all'utente quantificando tale resto in numero di monete da 50c, da 20c e da 10c (per semplicità non si considerino casi di granularità inferiore). Se l'importo risulta insufficiente, il metodo deve stampare un messaggio di errore.

4) metodo *percorsiPiuLunghi* che restituisca un set contenente i percorsi con il numero maggiore di fermate.

Il programma principale deve, infine, svolgere il seguente punto:

- a) richiedere all'utente i dati per istanziare un mezzo urbano e i nomi di due fermate. Istanziare il mezzo, richiamare il metodo del punto 2) e stampare a video un messaggio che indichi se il mezzo istanziato è un mezzo che copre un percorso rilevante per le due fermate. Si assuma la correttezza dell'input.