

Si consideri il modulo `moduloVeicoli.py` fornito dalla docente. Le classi presenti nel file `moduloVeicoli.py` hanno dei metodi che quando invocati richiedono che venga svolta una certa azione. Questi metodi invocano il metodo `azione()` della classe `Manovratore` sulla variabile di istanza `manovratore` perché venga svolta l'azione richiesta.

Scrivere nel file **`esercizio.py`** la classe `Manovratore` e la classe `Strada` che usa un oggetto `Manovratore` per fare in modo che

- i. quando l'auto è a riserva, venga fatto rifornimento, cioè `spia_benzina` venga settata a `false` ed effettuata la stampa di "L'auto {} sta facendo benzina"
- ii. quando un autobus effettua una fermata, deve essere aggiornata `ora_ultima_fermata` con l'ora corrente ed effettuata la stampa di "L'autobus {} sta effettuando una fermata alle ore {}".

Si vedano stampe del test per capire cosa deve comparire al posto delle parentesi graffe.

I valori delle variabili il cui nome inizia con `'_'` non possono essere letti o modificati direttamente. I metodi per aggiornare queste variabili sono forniti nelle classi.

Il metodo `azione()` NON deve contenere statement condizionali.

Il metodo `__init__` della classe `Strada` prende in input due tuple: la prima contiene i nomi `id_veicolo` delle auto e la seconda contiene le tuple (`id_veicolo`, `tratta`), dove `tratta` è una tupla di due stringhe.

`Strada` deve fornire il metodo di istanza `getVeicolo(id_veicolo)` che prende in input il nome del veicolo e restituisce il veicolo con quel nome.

La versione "base" dell'esercizio prevede che nella strada ci siano due auto e due autobus.

La versione con bonus prevede che `Strada` funzioni per un numero arbitrario di auto e autobus (cioè vuol dire che deve funzionare NON soltanto per gli insiemi di veicoli usati nel test!)

La versione con superbonus prevede che la versione con bonus funzioni in modo che, oltre a quanto già specificato da I e II, se un autobus arriva al capolinea, venga effettuata la stampa di "L'autobus {} è arrivato al capolinea: ora il suo tragitto è {}." e venga modificato il tragitto con quello in direzione opposta. **Non** usate un unico metodo per gestire sia la fermata che l'arrivo al capolinea.

Per acquisire di punti aggiuntivi alle prove in itinere occorre scrivere il codice per la versione con superbonus. Nel file di test `esercizio.py` ci sono i test per tutte e tre le versioni (basta togliere gli apicetti): potrebbero essere utili per progredire verso la versione finale.

NON MODIFICATE IL CODICE GIÀ SCRITTO NEI FILE.