

Esame di Programmazione Orientata agli Oggetti del 15 luglio 2022

nome	progetto <i>cognome+nome+n°_matricola</i>	
Matricola	Aula	Posizione

Eredità, interfacce e Container

Occorre rappresentare i prodotti di una compagnia noleggio veicoli:

Ciascun veicolo è caratterizzato dai seguenti dati: marca, modello (es modelli moto: scooter, city, enduro, custom... es modelli auto: suv, citycar, medium, large, lusso...), codice telaio (tutti di tipo stringa) e un costo base per unità minima di tempo [vedi oltre].

Esistono diversi tipi di veicoli: noi consideriamo bici, moto, auto, furgoni.

Ci sono veicoli che sono targati e altri no: le bici non lo sono.

Per guidare alcuni tipi veicoli, e dunque per il noleggio, occorre che l'utente sia **Patentato**, cioè che si possa rispondere alla richiesta `getPatente()` restituendo una patente caratterizzata da

- `tipoPatente` e `codicePatente` (verificare se adeguata al tipo di vettura)

NB: la patente c'è SOLO per le tipologie di veicoli che lo richiedono: tutte eccetto le biciclette.

A ogni veicolo noleggiato viene assegnato un *noleggio*.

Noleggio è una classe caratterizzata da

- `durata noleggio` (es. per una bici, 3 ore, ma anche 3 giorni, per un'auto, 10 giorni)
- `utente noleggiatore`, con i seguenti attributi:
 - nome, recapito telefonico e mail, codice fiscale
- `costo noleggio`

Il noleggio di auto e furgoni è **con cauzione** che occorre calcolare in modo specifico [vedi oltre].

Tutti i veicoli della compagnia noleggi sono caratterizzati da funzioni per:

- conoscere il *costo* del noleggio di un veicolo dato il periodo di noleggio desiderato e il modello desiderato che varia in base al tipo di veicolo
- *noleggiare* il veicolo per un determinato periodo se possibile
semplificazione importante: non si chiede qui di prenotare il noleggio ma di noleggiare veicoli attualmente non ancora noleggiati per un periodo voluto
- conoscere lo *stato* di noleggio (restituzione del noleggio o null)

Alcune tipologie di veicoli sono caratterizzati dal **poter avere passeggeri a bordo** (fare una scelta coerente per le tipologie di veicolo delineate):

in questo caso implementeranno il metodo `capienza_passeggeri()`.

Sia le auto che le moto si noleggianno per intere giornate e il costo base di noleggio per dì è un attributo specifico assegnato in fase di costruzione e modificabile.

I *furgoni* possono essere per trasporto *merci* (*camion*), in questo caso sono caratterizzati da una cubatura, o *persone* (*pullman*), in questo caso sono caratterizzati da un numero posti.

I furgoni si noleggianno a giorni e, qualunque sia marca e modello:

- il costo noleggio di un camion si calcola come cubatura*10€/giorno

- il costo noleggio di un pulmino si calcola come $n \cdot \text{posti} \cdot 15\text{€}/\text{giorno}$

Le *biciclette* noleggiate possono essere: bike o e-bike e si noleggianno a ore o a giorni con un costo:

- bike: 7 €/ora per la 1^a ora poi 4 €/ora, oppure 20€/giorno
- electric bike: 10 €/ora oppure 30 €/giorno

La *cauzione* di un'auto è calcolata come il costo di una settimana, la cauzione di un furgone è maggiorata del 50% rispetto al costo di una settimana.

Implementare per ogni attributo delle classi sin qui descritte gli opportuni metodi *getter* e/o *setter*.

Definire classi ed interfacce per rappresentare lo scenario menzionato.

I veicoli della compagnia sono registrati in una hashmap con chiave il codice telaio, prevedere le operazioni fondamentali; gestire le seguenti operazioni:

1. inserimento di un nuovo veicolo, gestendo eventuali eccezioni per valori non corretti degli attributi
2. cancellazione di un elemento con dato codice telaio
3. ricerca per codice telaio
4. restituire la lista di *auto*, dati il modello, e il range di costo noleggio al di
5. visualizzazione di tutti i veicoli, stampando le informazioni in base alle diverse tipologie identificate (utilizzare in un main dove inserire almeno un veicolo per tipo)
6. noleggio/restituzione di un veicolo, caratterizzato da un codice telaio, controllando che la selezione sia disponibile (in caso contrario, sollevare un'eccezione) e visualizzando il costo totale richiesto per il noleggio. Gestire le opportune eccezioni (es. rilascio di un veicolo non noleggiato)
7. visualizzazione di tutti i veicoli attualmente noleggiati
8. scrivere su file testo tutti i veicoli attualmente noleggiati con le relative informazioni sul noleggio.

Thread

Il parco auto prevede al suo interno un'officina revisione per le macchine che vengono restituite. L'officina prende in carico non più di 20 veicoli.

Supponiamo che i 4 tipi di veicoli (bici, moto, auto e furgoni) afferiscano da 4 flussi accettazione diversi identificati da 4 differenti thread (produttori) e che ogni thread associ a ogni veicolo sottoposto un ticket (un numero progressivo) possiamo ad es avere che il thread flussoAuto inserisca ... (auto 3) (auto 4)... mentre flusso Bici inserisca (bici1)... etc.

Gli operai dell'officina che prendono in carico i veicoli sono 2 thread (consumatori) e man mano revisionano i veicoli liberando l'officina per nuovi ingressi.

Con periodo molto più lento un altro thread (cancellatore) che segna il termine del turno di lavoro dell'officina svuota l'officina; dopo l'officina riprenderà con le comuni attività di accettazione e revisione.

Eccezione

sollevare, gestire eccezioni e definire almeno 2 eccezioni.