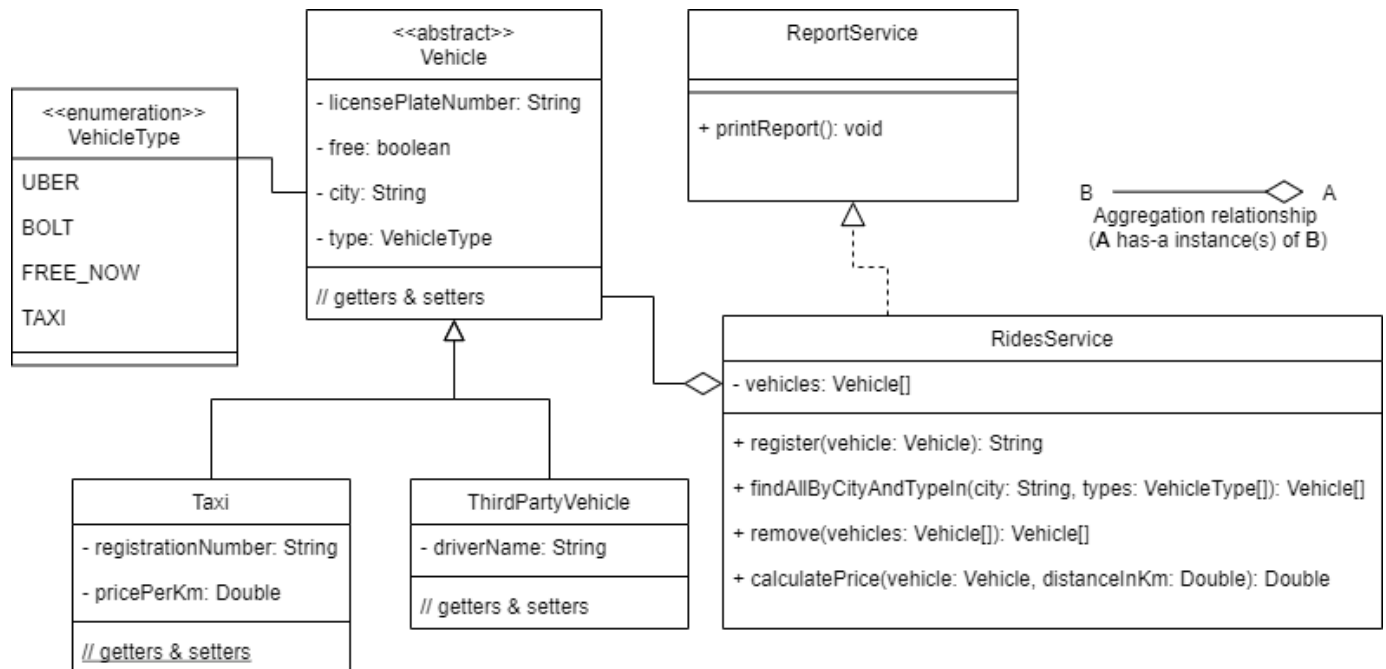


D. Rides

Aplicatia pe care trebuie sa o construiti astazi reprezinta un software de management al unor vehicule pentru transport de persoane. Aplicatia va lucra cu doua tipuri de vehicule: `Taxi` si `ThirdPartyVehicle`. Printre functionalitatile pe care trebuie sa le implementati se numara: inregistrarea unui nou vehicul, cautarea tuturor vehiculelor dupa orasul in care activeaza si tip, stergerea unor vehicule, calcularea pretului dupa tipul vehiculului si distanta de parcurs sau afisarea unui raport despre vehiculele din sistem.

Figure 1: Digrama UML



Cerinte

1. Cititi si intelegeti diagrama UML mai sus prezentata, iar apoi creati structura de clase dupa cum va este prezentat in diagrama.

2. Implementati metodele din clasa `RidesService` dupa cum urmeaza:

- Metoda `register` trebuie sa salveze obiectul de tipul `Vehicle` primit ca si parametru in vectorul `vehicles`. Valoarea intoarsa de metoda va fi `'Vehicle successfully registered'` daca operatiunea a avut loc cu succes;
- Daca obiectul primit ca si parametru care urmeaza a fi salvat nu are oricare dintre attributele `licensePlateNumber`, `city` sau `type` setate atunci se va intoarce mesajul `'Invalid data'`;
- Daca nu mai este loc pentru o noua intrare in vectorul `vehicles` atunci se va intoarce mesajul `'No more registrations allowed'`;
- Metoda `remove` trebuie sa stearga toate vehiculele primite ca si parametru din vectorul `vehicles` doar daca vehiculul ce urmeaza a fi sters are valoarea `true` pentru atributul `free` (faceti o cautare dupa `licensePlateNumber` aici);
- Metoda `remove` trebuie sa intoarca o lista de vehicule care nu au putut fi sterse sau o lista goala daca toate (cele primite ca si parametru) au fost sterse cu succes.

3. Creati o clasa `Main` unde sa se regaseasca metoda statica `main`. Creati o instanta a clasei `RidesService` si niste instance ale celorlalte clase dupa cum urmeaza:

- Creati o instanta a clasei `Taxi` si doua instance a clasei `ThirdPartyVehicle`;
- Inregistrati instancele create anterior folosind metoda `RidesService register`;
- Folositi metoda `RidesService remove` pentru a sterge una din instancele clasei `ThirdPartyVehicle`.

4. Implementati metoda `findAllByCityAndTypeIn` astfel incat aceasta sa intoarca o lista de obiecte de tipul `Vehicle` care au acelasi `city` ca si cel primit ca si parametru si `type` setat la orice valoare din valorile parametrului `types`. Implementati metoda `calculatePrice` astfel incat pentru obiectele de tipul `Taxi` sa se intoarca valoarea `pricePerKm * distanceInKm`, iar pentru restul tipurilor de vehicule folositi valoarea 2.34 pe post de `pricePerKm`.

5. Implementati metoda `printReport` astfel incat sa se afiseze datele din sistem similar cu ceea ce vedeti in imaginea de mai jos. Informatiile pe care trebuie sa le afisati sunt: `type`, `licensePlateNumber`, `pricePerKm` si `free` pentru obiectele de tip `Taxi` si `type`, `licensePlateNumber`, `driverName` si `free` pentru restul tipurilor de vehicule.

Figure 2: Example

```
1 Rides report:
2 Cluj-Napoca: 2 TAXI, 1 UBER
3 1. TAXI - CJ12ABC - 2.55 - liber
4 2. TAXI - AB22DEF - 3.21 - ocupat
5 3. UBER - B123TTT - Marcel - liber
6 Bucuresti: 1 UBER, 1 BOLT
7 1. UBER - B55TEF - Dan - liber
8 2. BOLT - B66GHE - Gheorghe - ocupat
```