

# UBER

Progettare un sistema di organizzazione viaggi autostop: un utente con la macchina organizza un viaggio, pubblicando la richiesta sul sistema cerca altri utenti che devono fare lo stesso tragitto.

Le eccezioni sono lanciate mediante la classe `AutoStopException`

La classe principale è **Autostop**

## Utenti

La classe **addUsers**(String,String) registra nuovi utenti all'applicazione, tramite nome e email. Genera un'eccezione se la email è presente.

La classe **addDrivers**(String, String, int, String) aggiunge utenti che hanno la macchina al sistema e definisce il numero di posti disponibili sulla sua vettura, insieme alla targa. Genera un'eccezione se la targa è già presente.

Un utente registrato senza auto potrebbe aggiungerla successivamente (**addCar**); **getDrivers**() fornisce una lista degli utenti con l'auto.

## Viaggi

L'organizzazione dei viaggi segue vari *stati*, Gli utenti con la macchina possono aprire una sessione di viaggio tramite **openTravelSession**() con status *OPEN*, in cui forniscono un tragitto, gli orari. Un utente può aprire o partecipare (prenotare) a una sola sessione alla volta. Il metodo lancia un'eccezione se non esiste il driver o ha già una sessione aperta.

Il metodo **getTravel**() restituisce tutte le sessioni aperte, con una lista di stringhe con il seguente formato "email - posti liberi - tappa 1 - tappa 2 eccetera"

Il metodo **bookTravel**() registra un utente a un viaggio (aperto), impostando tappa di ingresso. Lancia un'eccezione se il driver non ha sessioni aperte, se la tappa non è presente, se la tappa è l'ultima.

Il metodo **travelReady**() chiude le registrazioni, imposta lo stato a *READY*, il raggiungimento della capienza massima sposta lo status automaticamente, se il driver non ha sessioni aperte lancia un'eccezione.

Una volta avvenuti i viaggi, lo status cambia a *DONE* tramite **travelDone**(), se non ci sono sessioni Ready lancia un'eccezione.

**getUsersByTravel**() dato un driver, restituisce gli user registrati a una sua sessione aperta, lancia un'eccezione altrimenti.

# Informazioni

Tramite il metodo **numViaggiPerPersona()** resituire una lista ordinata degli utenti che hanno preso parte a più viaggi .

Il metodo **TopDrivers()** resituire una lista ordinata degli utenti driver per km percorsi (Num tappe), resiture nella lista stringhe “email Targa viaggi:tappe”

**travelByEmail()** restituisce tutti i viaggi associati a una mail sia come driver sia come user.