

Il traffico

Lambda expression e interfacce funzionali



Descrizione 1 - versione base

- ▶ Si vuole realizzare un metodo che applica dei filtri su una lista di automobili per stabilire quali siano abilitate a circolare.
- ▶ Implementare la classe **Automobile** con marca, modello e targa.
- ▶ Implementare il filtro come metodo statico che
 - ▶ riceverà in input la lista di automobili e un oggetto di tipo **Predicate<Automobile>**
 - ▶ ritornerà la nuova lista di automobili (che saranno quelle abilitate a circolare)
- ▶ Scrivere la classe col main per impostare i seguenti criteri, usando la sintassi lambda
 - ▶ circolano solo le auto di marca Fiat
 - ▶ circolano solo le auto con targhe pari
 - ▶ circolano solo le auto con targhe dispari

Descrizione 2 - versione avanzata

- ▶ Scrivere una classe **Traffico** per la gestione di **Automobili**.
- ▶ Le automobili sono definite da marca, modello, targa, euro5 (**boolean**)
- ▶ La lista di automobili viene generata dinamicamente e avrà una dimensione e una composizione casuale
- ▶ Viene poi generato randomicamente un valore per le condizioni dell'aria (valore di CO2, compreso tra 360 e 1000 ppm)
- ▶ Nella classe sarà memorizzata l'ultima ordinanza di circolazione legata alle targhe (pari o dispari), la prima ordinanza sarà decisa casualmente.
- ▶ Scrivere quindi una classe col main che, dopo avere generato un oggetto **Traffico**, stabilisca quali auto circolano in base a questi criteri:
 - ▶ Quando il livello di CO2 nell'aria supera il limite di 700, solo le auto euro5 potranno circolare
 - ▶ Quando il numero di auto supera 10 mila unità dovranno scattare le misure che utilizzano le targhe alterne: pari oppure dispari, l'opposto dell'ordinanza precedente (valore memorizzato nella classe **Traffico**)