

Do wykonania załóż repozytorium. Jeśli będziesz potrzebować pomocy, prześlij informację z czym masz problem oraz prześlij adres repozytorium. Commity powinny być regularne tj. po każdej napisanej funkcjonalności/funkcji/metodzie powinien być zrobiony commit.

Tematyka: enum, scanner, LocalDate, DateTimeFormatter, stream, filter, map, comparator, metody, obiekty, kompozycja, varargs, sort

- **Stwórz enum o nazwie "CarType" który posiada wartości:**
SEDAN, CABRIO, SUV, HATCHBACK, KOMBI
- **Stwórz klasę CarBrand o wartościach:**
AUDI, BMW, MERCEDES, HONDA, MAZDA, HYUNDAI, FORD, FIAT... (możesz kilka dodać)
- **Stwórz klasę Car o polach:**
 - marka(CarBrand),
 - nazwa,
 - typ(CarType),
 - spalanie(double),
 - wielkość silnika(double),
 - ilość miejsc,
 - moc silnika,
 - data wyprodukowania(LocalDate),
 - automat(boolean),
 - (*)napęd na przód/tył (wymyśl jak to zrobić)

DOPISZ TEJ KLASIE GETTERY, SETTERY ORAZ KONSTRUKTOR

-Commit1-

Stwórz zestaw danych testowych(zbiór samochodów) na których będziesz pracować.

- **Stwórz klasę oferta samochodowa, która posiada pola:**
 - Car
 - cena zakupu
 - cena najmu
- **Stwórz klasę CarShop (sklep/magazyn/warsztat samochodowy) który powinien posiadać pola:**
 - nazwę
 - listę ofert samochodowych

-Commit2-

Stwórz zestaw danych testowych(zbiór ofert) na których będziesz pracować.

Stwórz metody:

- metodę z varargs, która jako parametr przyjmuje ofertę samochodową (różną ilość ofert samochodowych) i dodaje wszystkie obiekty przekazane w parametrze do listy ofert podanego sklepu.

-Commit3-

metody (samochód):

- pobierz wszystkie samochody (getter dla listy)
- pobierz listę wszystkich marek które posiada warsztat (filtrowanie streamem)
- *pobierz listę wszystkich samochodów z napędem na 4 koła

-Commit4-

- pobierz wszystkie samochody nie starsze niż podana przez użytkownika data (parsowanie daty od użytkownika, porównywanie dat wyprodukowania, filtrowanie streamu)
- pobierz wszystkie samochody podanej marki (parsowanie stringa na enuma/markę)
- pobierz samochody i posortuj je po najniższym spalaniu (Comparator)

-Commit5-

- pobierz samochody i posortuj je po najwyższej mocy, a następnie zwróć (Sort, comparator, stream)

metody (oferty)

- pobierz listę samochodów posortowanych po cenie
- pobierz listę ofert najtańszych najmów
- pobierz listę wszystkich samochodów
- pobierz listę samochodów które mają najwyższą moc przy najniższej cenie

-Commit6-

Tematyka2: Interfejsy, klasa abstrakcyjna, (dziedziczenie), listy, sety, mapy, wyjątki, wątek, tablica, pliki (odczyt i zapis), klasa/metoda statyczna