

ZARZĄDZANIE KOLEKCJĄ SAMOCHODÓW

Zaimplementuj klasę **Car**. Klasa posiada pola składowe **model**, **price**, **color**, **mileage** oraz kolekcję napisów **components** reprezentująca wyposażenie samochodu. Dla klasy przygotuj podstawowe metody ułatwiające korzystanie z klasy. Przygotuj również logikę, która pozwoli walidować pola składowe klasy. Model musi składać się tylko i wyłącznie z dużych liter oraz białych znaków. Kolor przyjmuje wartości typu wyliczeniowego **Color** (przygotuj przykładowe wartości dla typu wyliczeniowego). Pole **mileage** oraz **price** mogą przyjmować wartości tylko nieujemne. Kolekcja **components** może składać się z napisów, które zawierają tylko i wyłącznie duże litery i białe znaki. Możesz zastosować wzorzec projektowy builder.

Następnie zaimplementuj klasę **Cars**, której polem składowym jest kolekcja obiektów klasy **Car** o nazwie **cars**. Dla klasy przygotuj konstruktor, który jako argument przyjmuje nazwę pliku w formacie JSON przechowującego dane o przykładowych samochodach. Przykładowa postać pliku została przedstawiona poniżej. Dane z pliku należy pobrać do kolekcji znajdującej się w klasie **Cars**.

W klasie **Cars** przygotuj metody, które pozwolą uzyskać następujące informacje:

- Przesłonięta metoda **toString**, która zwróci napis pokazujący dane wszystkich samochodów z kolekcji w przejrzystym formacie.
- Metoda, która zwraca nową kolekcję elementów **Car** posortowaną według podanego jako argument metody kryterium. Metoda powinna mieć możliwość sortowania po nazwie modelu, kolorze, cenie oraz przebiegu. Dodatkowo należy określić czy sortowanie ma odbywać się malejąco czy rosnąco.

- Metoda zwraca kolekcję elementów typu Car, które posiadają przebieg o wartości większej niż wartość podana jako argument metody.
- Metoda zwraca mapę, której kluczem jest kolor, natomiast wartością ilość samochodów, które posiadają taki kolor. Mapa powinna być posortowana malejąco po wartościach.
- Metoda zwraca mapę, której kluczem jest nazwa modelu samochodu, natomiast wartością obiekt klasy Car, który reprezentuje najdroższy samochód o tej nazwie modelu. Mapa powinna być posortowana kluczami malejąco.
- Metoda wypisuje statystykę samochodów w zestawieniu. W statystyce powinny znajdować się wartość średnia, wartość najmniejsza, wartość największa dla pól opisujących cenę oraz przebieg samochodów.
- Metoda zwraca samochód, którego cena jest największa. W przypadku kiedy więcej niż jeden samochód posiada największą cenę należy zwrócić kolekcję tych samochodów.
- Metoda zwraca kolekcję samochodów, w której każdy samochód posiada posortowaną alfabetycznie kolekcję komponentów.
- Metoda zwraca mapę, której kluczem jest nazwa komponentu, natomiast wartością jest kolekcja samochodów, które posiadają ten komponent. Pary w mapie powinny być posortowane malejąco po ilości elementów w kolekcji reprezentującej wartość pary.
- Metoda zwraca kolekcję samochodów, których cena znajduje się w przedziale cenowym $\langle a, b \rangle$. Wartości a oraz b przekazywane są jako argument metody. Kolekcja powinna być posortowana alfabetycznie według nazw samochodów.

Przykładowa postać pliku JSON przechowująca informacje o dwóch samochodach:

```
{
  "cars": [
    {
      "model": "BMW",
      "price": 120,
      "color": "BLACK",
      "mileage": 1500,
      "components": [
        "ABS",
        "ALLOY WHEELS"
      ]
    },
    {
      "model": "MAZDA",
      "price": 160,
      "color": "WHITE",
      "mileage": 2500,
      "components": [
        "AIR CONDITIONING",
        "BLUETOOTH"
      ]
    }
  ]
}
```