

Interfejsy

Wypożyczalnia pojazdów udostępnia swoim klientom:

- **samochody,**
- **motorówki,**
- **rowery,**
- **hulajnogi.**

Wszystkie samochody i motorówki są pojazdami spalinowymi, zasilanymi różnymi rodzajami paliwa : olejem napędowym, benzyną, a niektóre z samochodów benzynowych również gazem LPG lub CNG.

Samochody oraz rowery są pojazdami parkowalnymi i są parkowane w **garażu**. Wypożyczalnia ma kilka **garaży**, w których mogą być parkowane pojazdy (w garażu może parkować jednocześnie tylko jeden parkowalny pojazd).

1. Początkową bazę danych wczytać podczas uruchomienia programu z pliku XML (format poniżej). Przed zakończeniem programu zapisać do pliku aktualny stan pojazdów w wypożyczalni.

```
<pojazdy>
  <samochod>
    <nazwa>Honda Civic</nazwa>
    <rodzajPaliwa>3</rodzajPaliwa>
  </samochod>
  <motorowka>
    <nazwa>Super motorowka</nazwa>
    <rodzajPaliwa>0</rodzajPaliwa>
  </motorowka>
  <samochod>
    <nazwa>Mercedes CLK</nazwa>
    <rodzajPaliwa>5</rodzajPaliwa>
  </samochod>
  <rower>
    <nazwa>Giant</nazwa>
  </rower>
  <hulajnoga>
    <nazwa>Najfajniejsza hulajnoga</nazwa>
  </hulajnoga>
</pojazdy>
```

2. Podczas wczytywania pojazdów, każdemu nadać unikalne ID (wykorzystać pole statyczne w klasie Pojazd).
3. Umożliwić parkowanie pojazdu w jednym z pięciu garaży (liczbę garaży przyjąć w konstruktorze wypożyczalni).
4. Umożliwić dodawanie nowych pojazdów (wczytać dane z klawiatury) oraz usuwanie pojazdów o określonym ID.
5. Umożliwić wypisanie wszystkich pojazdów znajdujących się w wypożyczalni.

6. Umożliwić sortowanie zbioru pojazdów według kolejności: parkowane, typ pojazdu, nazwa pojazdu, rodzaj paliwa, ilość paliwa.

Wskazówki do rozwiązania zadania:

- wszystkie pogrubione słowa to klasy (pochylone – klasy abstrakcyjne),
- wszystkie podkreślone słowa to interfejsy,
- rodzaj paliwa należy przechować wykorzystując arytmetykę bitową,
- garaż powinien posiadać informacje o tym czy jest pusty, a pojazd parkowalny powinien posiadać referencję do garażu, w którym jest zaparkowany,
- interfejs Spalinowy powinien wymuszać implementację metody umożliwiającej tankowanie (określoną liczbę litrów danego typu paliwa; wartość zwracana określająca czy wybrano odpowiedni rodzaj paliwa) oraz odczyt rodzaju paliwa,
- interfejs Parkowalny powinien implementację metody umożliwiającej parkowanie pojazdu w podanym garażu – wartość zwracana określająca czy udało się zaparkować auto (czy garaż był pusty) oraz wyparkowanie pojazdu,
- każda klasa powinna implementować własną metodę toString „przedstawiającą” dany pojazd,
- każda klasa i interfejs powinna być zaimplementowana w oddzielnym pliku kodu,
- używając klasy ArrayList w celu przechowywania listy pojazdów do sortowania można wykorzystać metodę Collections.sort(List<T> list) zamiast Array.sort().