

Samochody

Projekt zawiera:

1. Program.cs z funkcją main
2. Interfejsy części samochodowe
3. Car.cs zawierający klasę Car.

Założenia:

1. Dodanie nowego modelu samochodu, lub części nie powinno spowodować zmian w kodzie fabryki samochodów.
2. Tylko jeden model będzie produkowany w danym czasie.
3. Fabryka samochodów nie jest świadoma tego jaki model produkuje.

Twoje zadanie:

1. Implementacja interfejsów maszyn. Opis poniżej.
2. Implementacja funkcji która stworzy samochody i uruchomi je - opis poniżej.
3. Implementacja funkcji main, która uruchomi fabrykę samochodów dla trzech różnych zestawów modeli - opis poniżej.

Interfejsy części:

Gotowy samochód składa się z trzech kategorii części: silnika, wnętrza kabiny, oraz karoserii. Możemy wyróżnić trzy kategorie części: IEngine, Interior, oraz IBody. Produkt końcowy (samochód) powstaje poprzez ich połączenie.

IEngine: Capacity oraz typ (ustawiane poprzez konstruktor).

- SportEngine - pojemność wynosi minimum 2 - jeśli podana jest mniejsza to jest ignorowana. Typ Lpg również jest ignorowany i zamiast niego ustawiany jest typ Petrol
- EfficientEngine - dowolny typ silnika, ale pojemność wynosi co najwyżej 3.
- FamilyEngine - Typ Diesel jest ignorowany i zamiast niego ustawiany jest typ Petrol. Pojemność może wynosić co najwyżej 2.5.

Interior: Lista wybranych opcji przekazywana jest poprzez konstruktor. Możliwe opcje to: "alufelgi", "skórzane fotele", "ABS", "ASR", "DVD", "Imobilizer".

- FamilyCarInterior - nie może mieć na wyposażeniu alufelg, oraz DVD (zostaje odfiltrowane z przekazanej listy)
- StandardInterior - nie może mieć zamontowanego imobilizera.

- PremiumInterior - DVD jest zbyt niskiego standardu, więc nie może zostać wybrane

IBody: Typ, oraz kolor przekazywane przez konstruktor. Możliwe kolory to:

- FancyBody - nie może być samochodem Combi - zamiast tego Coupe. Nie może mieć nudnego koloru jak White i Black; są one zamieniane na Blue.
- FamilyBody - kolory tylko White lub Black. Zamiast typu Cabriolet: Sedan.
- SportBody - niemożliwy typ Combi, zamiast tego: Cabriolet. Nie może mieć koloru NeonPink, zamiast tego: Rouge.

Każdy z interfejsów zawiera metodę getDescription. Powinna ona zwrócić opis nadej części samochodu. Np.: "Fancy NeonPink Coupe", albo "Family interior with ABS and Immobilizer". Te opisy powinny łączyć się w spójny opis samochodu: "Fancy NeonPink Coupe powered by Efficient 1.9 LPG engine with family interior with ABS, Immobilizer, alufelgi".

Funkcja budująca samochody:

Funkcja ta będzie tworzyła samochody. Powinna ona otrzymać sposób na stworzenie 2 modeli samochodów (modelem nazywamy zestaw rodzajów części, np.: SportEngine + FamilyCarInterior + FancyBody).

- Stwórz po dwa samochody każdego rodzaju. Dobierz rodzaje silnika, opcje wnętrza tak, aby któreś z nich mogły zostać przefiltrowane przez podane powyżej zasady.

Funkcja main:

Powinna ona wywołać funkcję budującą samochody dla 3 różnych zestawów modeli.