

ZADANIE INDYWIDUALNE – OOP w Groovy

Abstrakcja i klasa abstrakcyjna (Samochod)

Cel zadania

Celem zadania jest:

- zrozumienie **abstrakcji** w programowaniu obiektowym,
 - umiejętność użycia **klasy abstrakcyjnej**,
 - poprawne zastosowanie **dziedziczenia i polimorfizmu** w Groovy.
-

Część 1 – Klasa abstrakcyjna

1. Utwórz **abstrakcyjną klasę** Samochod.
2. Klasa musi zawierać:
 - pola:
 - marka (String)
 - rokProdukcji (int)
 - konstruktor inicjalizujący oba pola
 - **dwie metody abstrakcyjne**:
 - rodzajSilnika() – zwraca String
 - maksymalnaPredkosc() – zwraca int
 - **jedną metodę konkretną**:
 - info() – zwraca opis samochodu jako String

Część 2 – Klasy konkretne

3. Utwórz co najmniej **dwie klasy dziedziczące po Samochod**:
 - SamochodSpalinowy
 - SamochodElektryczny
4. Każda z klas:
 - musi posiadać **własne pole charakterystyczne**:
 - spalinowy: pojemnoscSilnika
 - elektryczny: zasieg
 - **musi zaimplementować wszystkie metody abstrakcyjne** z klasy Samochod

Część 3 – Program główny

5. Utwórz klasę Main.
 6. W metodzie main:
 - utwórz co najmniej **dwa obiekty różnych typów samochodów**
 - przypisz je do zmiennych typu Samochod
 - dodaj je do listy List<Samochod>
 - wyświetl informacje o każdym samochodzie, wywołując metodę info()
-

Ograniczenia (bardzo ważne)

W rozwiązaniu:

- **nie wolno używać** instanceof
 - **nie wolno sprawdzać typu** obiektu przez if/else
 - **nie wolno tworzyć obiektu klasy Samochod**
-

!!!! Dla chętnych (nieobowiązkowe)

- Dodaj klasę SamochodHybrydowy
- Dodaj metodę kosztPrzejazduNa100km()
- Posortuj listę samochodów według maksymalnej prędkości