

Wariant 701

- Stwórz nowy projekt a w nim pakiet `pl.edu.uwm.vehicles` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `Car` z prywatnymi polami: `model` (typu `String`, pol. model), `yearOfManufacture` (typu `int`, pol. rok produkcji) oraz `mileage` (typu `double`, pol. przebieg).
- Zaimplementuj konstruktor parametryczny ustawiający pola klasy z argumentów.
- Nadpisz metodę `toString()`, tak aby zwracała napis postaci:

Samochód. Model: `[model]`, rok produkcji: `[yearOfManufacture]`, przebieg: `[mileage]` km.

lub gdy model jest pusty lub nullem to pomiń nazwę:

Samochód. Rok produkcji: `[yearOfManufacture]`, przebieg: `[mileage]` km.

Zwróć uwagę na wielkość znaków i znaki interpunkcyjne.

- Stwórz klasę `TestCar` z metodą `main`. W metodzie `main` stwórz obiekt klasy `Car` i przetestuj wszystkie metody.

Zadanie należy umieścić na pendrive. Cały projekt ma być spakowany jako archiwum zip. Nazwa według schematu NUMERWARIANTU_NUMERALBUMU.zip np. 602_123456.zip.

Wariant 702

- Zainicjuj nowy projekt i w nim utwórz pakiet `pl.edu.uwm.motorcycles`.
- Stwórz oddzielne pliki dla klas
- W klasie `Motorcycle` utwórz prywatne pola: `brand` (typu `String`, pol. marka), `yearOfProduction` (typu `int`, pol. rok produkcji) i `fuelConsumption` (typu `double`, pol. zużycie paliwa).
- Implementuj konstruktor z parametrami inicjującymi te pola.
- Nadpisz metodę `toString()` tak, aby zwracała informacje o motocyklu w formacie:

Motocykl. Marka: `[brand]`, rok prod.: `[yearOfProduction]`, zużycie: `[fuelConsumption]` l/100km.

lub gdy brand jest pusty lub nullem:

Motocykl. Rok prod.: `[yearOfProduction]`, zużycie: `[fuelConsumption]` l/100km.

- Stwórz klasę `TestMotorcycle` z metodą `main`, gdzie będziesz testować wszystkie metody klasy `Motorcycle`.

Zadanie należy umieścić na pendrive. Cały projekt ma być spakowany jako archiwum zip. Nazwa według schematu NUMERWARIANTU_NUMERALBUMU.zip np. 602_123456.zip.