

Dokumentacja – projekt 3

„Centralna Ewidencja Pojazdów i Kierowców”

Radosław Tuzimek

Założenia:

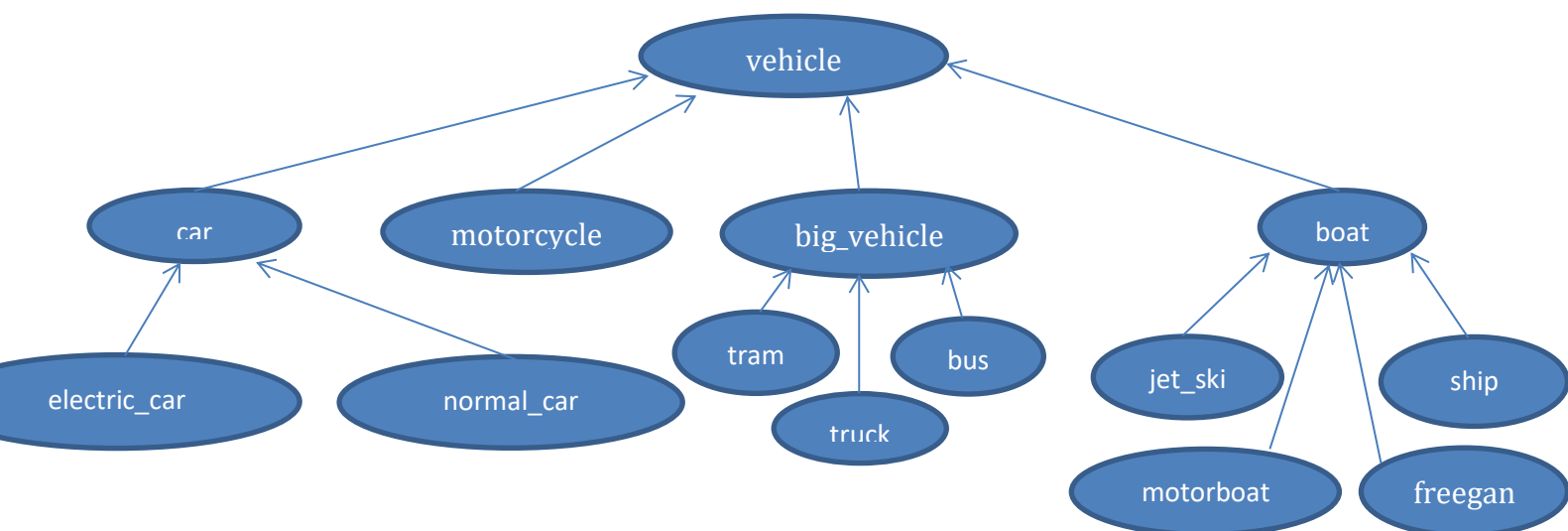
1. Każdy pojazd może mieć wielu właścicieli (współwłaścicieli), jeśli właścicielem jest firma to pojazd nie może mieć innych współwłaścicieli.
2. Każdy pojazd posiada własny typ i nie możliwe jest ich łączenie lub przypisanie pojazdu do więcej niż jednego.
3. Program umożliwia: rejestrację nowych pojazdów, przerejestrowanie pojazdów, zmianę właścicieli oraz przypisanie pojazdom polisy OC.

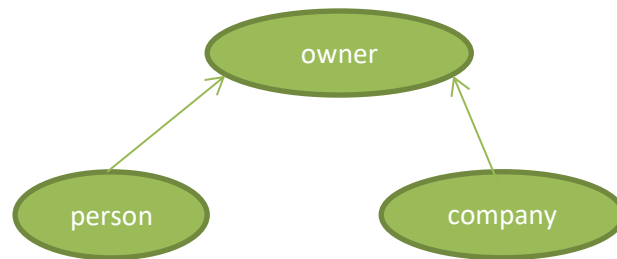
Opis klas:

1. **vehicle** – klasa bazowa dla wszystkich klas pojazdów, oprócz danych (numer rejestracyjny, numer OC, numer VIN, typ pojazdu, marka, rodzaj paliwa) zawiera wskaźnik na właściciela pojazdu. Klasa zawiera również metody umożliwiające zmianę numeru rejestracyjnego, dodanie OC, zmianę właściciela, wyświetlenie informacji o pojeździe (w dwóch wersjach, wybranych danych w formacie umożliwiającym wyświetlenie ich w tabeli oraz pełnym który wyświetla użytkownikowi szczegółowe dane na temat pojazdu), dodając typ pojazdu oraz metody pozwalające na podgląd aktualnego numeru rejestracyjnego, numeru OC czy właściciela.
2. **car** – klasa pochodna klasy vehicle, zawiera dane (kolor, moc silnika, ilość miejsc), oraz rozszerzenie metody wypisującej szczegółowe informacje.
3. **electric_car** - klasa pochodna klasy car, zawiera dodatkowe dane (pojemność baterii, zasięg (maksymalna ilość kilometrów przejechana na jednym ładowaniu)), oraz rozszerzenie metody wypisującej szczegółowe informacje.
4. **normal_car** - klasa pochodna klasy car, zawiera dodatkowe dane (pojemność baku, maksymalna prędkość) oraz rozszerzenie metody wypisującej szczegółowe informacje.
5. **motorcycle** - klasa pochodna klasy vehicle, zawiera dodatkowe dane (model, maksymalna prędkość, przyśpieszenie) oraz rozszerzenie metody wypisującej szczegółowe informacje.
6. **big_vehicle** - klasa pochodna klasy vehicle, zawiera dodatkowe dane (kolor, waga) oraz rozszerzenie metody wypisującej szczegółowe informacje.
7. **tram** - klasa pochodna klasy big_vehicle, zawiera dodatkowe dane (model, wiek) oraz rozszerzenie metody wypisującej szczegółowe informacje.

8. **truck** - klasa pochodna klasy `big_vehicle`, zawiera dodatkowe dane (długość, rodzaj przewożonych produktów, ładowność) oraz rozszerzenie metody wypisującej szczegółowe informacje.
9. **bus** - klasa pochodna klasy `big_vehicle`, zawiera dodatkowe dane (długość, maksymalną ilość pasażerów) oraz rozszerzenie metody wypisującej szczegółowe informacje.
10. **boat** - klasa pochodna klasy `vehicle`, zawiera dodatkowe dane (kolor, wyporność, maksymalną ilość osób na pokładzie) oraz rozszerzenie metody wypisującej szczegółowe informacje.
11. **motorboat** - klasa pochodna klasy `boat` zawiera dodatkowe dane (pojemność baku, maksymalną prędkość) oraz rozszerzenie metody wypisującej szczegółowe informacje.
12. **jet_ski** - klasa pochodna klasy `boat` zawiera dodatkowe dane (pojemność baku, maksymalną prędkość, wagę) oraz rozszerzenie metody wypisującej szczegółowe informacje.
13. **ship** - klasa pochodna klasy `boat` zawiera dodatkowe dane (przeznaczenie (do jakich celów wykorzystywany jest okręt)) oraz rozszerzenie metody wypisującej szczegółowe informacje.
14. **freegan** - klasa pochodna klasy `boat` zawiera dodatkowe dane (ładowność, spalanie) oraz rozszerzenie metody wypisującej szczegółowe informacje.
15. **owner** – klasa bazowa dla właścicieli pojazdów. Zawiera informację o numerze właściciela, oraz metody wyświetlające dane właściciela.
16. **person** – klasa pochodna klasy `owner`, zawiera informację o imieniu i nazwisku właściciela oraz wskazanie na współwłaściciela. Posiada metody umożliwiające podgląd imienia i nazwiska właściciela.
17. **company** - klasa pochodna klasy `owner`, zawiera informację o nazwie firmy oraz posiada metodę umożliwiającą jej podgląd.

Hierarchia klas:





Interfejs:

- Dane do programu wprowadzają odpowiednie funkcje w zależności od podjętego przez użytkownika działania. Funkcja `LOAD()` odpowiada za wczytanie informacji o pojeździe podczas jego tworzenia, funkcje `RE_REGISTER`, `OC_ADD()` i `CHANGE_OWNER()` wczytują dane przy zmianach poszczególnych atrybutów (numeru rejestracyjnego, numeru polisy OC, właściciela).
- Użytkownik wybiera poszczególne działania z menu (dodaj nowy pojazd, wyświetl listę pojazdów, wyświetl szczegóły wybranego pojazdu, zmiana numer rejestracyjny wybranego pojazdu, dodaj OC, zmień właściciela wybranego pojazdu, wykonaj testy, zakończ program) wpisując odpowiednią cyferkę i zatwierdzając enterem.
- W przypadku kiedy użytkownik popełni błąd przy podejmowaniu decyzji o działaniu (np. wpisze literkę zamiast cyferki) program zwróci komunikat o błędzie „Błąd! To nie jest liczba, podaj jeszcze raz:” i poprosi o ponowne wpisanie numeru działania.
- Po wykonaniu odpowiedniego działania program wróci do menu głównego.
- Program umożliwia podgląd informacji z systemu w dwóch wersjach: wypisanie tabeli (listy) zawierającej wszystkie pojazdy znajdujące się w systemie wraz z ich podstawowymi danymi oraz wersję umożliwiającą podgląd szczegółowych informacji o danym pojeździe.
- Dane nie są zapisywane po zamknięciu programu.

Testowanie:

Funkcja test przeprowadza następujące testy:

- `test_1` – test sprawdzający poprawność działania metody zmieniającej numer rejestracyjny, test dwukrotnie zmienia numer i sprawdza czy na każdym etapie został on zmieniony poprawnie.
- `test_2` – test sprawdzający poprawność działania metody zmieniającej numer polisy OC, test dwukrotnie zmienia numer i sprawdza czy na każdym etapie został on zmieniony poprawnie.

- test_3 – test sprawdzający poprawność działania metody zmieniającej właściciela, test dwukrotnie zmienia go i sprawdza czy na każdym etapie został on zmieniony poprawnie.

Po wybraniu opcji testowania na ekranie wyświetla się komunikat o rezultacie każdego z testów (PASS - test zaliczony, FAIL – test nie zaliczony).