

## **Struktura klas**

### **Klasy bazowe i dziedziczące**

**Klasa Vehicle** (bazowa klasa dla pojazdów):

Atrybuty:

std::string make: marka pojazdu.

std::string model: model pojazdu.

int year: rok produkcji.

Metody:

Konstruktor: inicjalizuje atrybuty pojazdu.

std::string getMake() const: zwraca markę pojazdu.

std::string getModel() const: zwraca model pojazdu.

int getYear() const: zwraca rok produkcji.

Przeciążenie operatora <<: umożliwia strumieniowe wypisywanie informacji o pojeździe.

Klasa Car (dziedziczy po Vehicle)

Klasa Truck (dziedziczy po Vehicle)

### **Klasa klienta**

Klasa Customer (klient):

Atrybuty:

std::string name: imię i nazwisko klienta.

std::string contact: dane kontaktowe klienta.

std::string license: numer prawa jazdy klienta.

Metody:

Konstruktor: inicjalizuje atrybuty klienta.

std::string getName() const: zwraca imię i nazwisko klienta.

std::string getContact() const: zwraca dane kontaktowe klienta.

std::string getLicense() const: zwraca numer prawa jazdy klienta.

Przeciążenie operatora <<: umożliwia strumieniowe wypisywanie informacji o kliencie.

Klasa zarządzająca systemem

Klasa RentalSystem (system wypożyczalni):

Atrybuty:

std::vector<Customer\*> customers: wektor wskaźników do obiektów Customer.

std::vector<Vehicle\*> vehicles: wektor wskaźników do obiektów Vehicle.

std::vector<std::tuple<std::string, std::string, std::string>> reservations: wektor rezerwacji (imię klienta, marka pojazdu, model pojazdu).

Metody:

-addVehicle: dodaje pojazd do systemu.

-registerCustomer: rejestruje klienta w systemie.

-makeReservation: dokonuje rezerwacji pojazdu.

-rentVehicle: wypożycza pojazd.

-returnVehicle: przyjmuje zwrot pojazdu.

-printReservations: wyświetla wszystkie rezerwacje.

-printVehicles: wyświetla wszystkie pojazdy.

-printCustomers: wyświetla wszystkich klientów.

-saveVehiclesToFile: zapisuje pojazdy do pliku.

-loadVehiclesFromFile: wczytuje pojazdy z pliku.

-saveCustomersToFile: zapisuje klientów do pliku.

-loadCustomersFromFile: wczytuje klientów z pliku.

## **Powiązania klas**

### **RentalSystem:**

Zarządza instancjami klas Vehicle (i jej podklas Car i Truck) oraz Customer.

Przechowuje listy pojazdów i klientów, co umożliwia rezerwacje, wypożyczenia i zwroty pojazdów.

Umożliwia zapis i odczyt danych z plików, co zapewnia trwałość danych między uruchomieniami programu.

### **Vehicle, Car, Truck:**

Klasy Car i Truck dziedziczą po klasie Vehicle, co pozwala na wykorzystanie polimorfizmu.

Instancje tych klas są zarządzane przez RentalSystem.

**Customer:**

Przechowuje informacje o klientach.

Instancje tej klasy są zarządzane przez RentalSystem.

**Struktury danych**

Wektory i Listy.np:

-std::vector<Vehicle\*> vehicles;

-std::list<Customer\*> customers;