

## Projekt Car-Rent

Projekt ma na celu wspomóc pracę wypożyczalni samochodów. Bazuje on łącznie na sześciu plikach:

- ❖ CAR-RENT.py
- ❖ operations.py
- ❖ model.py
- ❖ cars\_data.json
- ❖ rentals\_data.json
- ❖ bookings\_data.json

Jak wskazuje na to rozszerzenie, pierwsze trzy pliki zawierają kod w języku python, dzięki któremu nasz program działa. Pozostałe trzy pliki służą nam jako bazę naszych danych. To z nich odczytujemy wcześniej zapisane dane. Nie są one jednak niezbędne - nawet bez nich zdołamy uruchomić nasz program, lecz nie będziemy mieli w nim wtedy żadnych danych. Jeśli by się tak stało, użytkownik zostanie powiadomiony o trudnościach z odczytem plików zawierających dane. Aczkolwiek, jeśli w trakcie działania programu dodamy nowe dane, po wyjściu zostaną one zapisane do plików o nazwach, których nam brakowało.

Program uruchamiamy poprzez tytułowy CAR-RENT.py. Naszym oczom pojawi się główne menu dające nam 6 opcji wyboru:

1. Pierwszą z nich jest wyszukiwanie samochodów w naszej bazie danych. Aby wyszukać samochód musimy wybrać parametr za pomocą którego będziemy go szukać. Jeśli podamy go poprawnie ujrzymy listę znalezionych samochodów. Jeżeli wyniki nie są zadowalające możemy zawęzić wyszukiwanie poprzez dodanie kolejnych parametrów. Musimy jednak pamiętać, że każdy parametr możemy podać tylko raz.
2. Kolejną opcją jest przejście do menu edycji samochodów. Wybierając ją, będziemy mieli możliwości dodania nowego samochodu do bazy danych, edytowania lub usunięcia już istniejącego samochodu.
3. Trzecią z kolei jest przejście do menu rezerwacji. Tam możemy zarezerwować samochód (o ile jest dostępny w wybranym przez nas czasie), edytować istniejącą rezerwację lub ją usunąć.
4. Następna opcja pozwala przejść do menu wypożyczeń. Możemy tam odebrać złożoną wcześniej rezerwację, jeśli zaczyna się tego samego dnia. Możemy również wypożyczyć „na miejscu” te samochody, które są aktualnie dostępne i niezarezerwowane. Ponadto, możemy zwrócić wypożyczony samochód, jeśli czas wypożyczenia się skończył lub uzyskać informacje o wypożyczeniach, które przekroczyły opłacony czas.
5. Piąta opcja prowadzi nas do menu bazy danych, a w niej, możemy wyświetlić na ekran listę: samochodów, rezerwacji lub wypożyczeń.
6. Ostatni wybór zamyka program i zapisuje dane do plików bazowych.

## Architektura programu

Jak już wcześniej wspomniano program opiera się na 3 plikach. W pliku model.py znajdziemy zaimplementowane 3 klasy:

- ❖ Car – tworząca obiekt samochód posiadający atrybuty takie jak marka, model samochodu, kolor, rok produkcji, typ samochodu, typ silnika, konsumpcja paliwa, liczba miejsc, unikalny identyfikator, oraz status dostępności.
- ❖ Booking - tworząca obiekt rezerwacji posiadający atrybuty takie jak identyfikator wypożyczonego samochodu, krótką informację o samochodzie, początek oraz koniec rezerwacji, unikalny identyfikator rezerwacji i całkowity koszt.
- ❖ Rental - tworząca obiekt wypożyczenia posiadający atrybuty takie jak identyfikator wypożyczonego samochodu, krótką informację o samochodzie, początek oraz koniec rezerwacji, całkowity koszt, unikalny identyfikator wypożyczenia oraz informację czy przekracza ono swój planowany czas.

Kolejnym plikiem jest operations.py. Jest on najobszerniejszym z wszystkich plików. Znajdziemy w nim klasę Database która pozwala nam stworzyć naszą bazę danych z listami samochodów, rezerwacji oraz wypożyczeń. Zawiera ona funkcje zapisujące oraz odczytujące dane z plików do list. W kolejnych wierszach znajdują się wszystkie funkcje, z których musimy korzystać, aby nasz program działał. Można je podzielić na trzy części. Pierwszą z nich są funkcje wspomagające zapis oraz odczyt danych, które używamy w funkcjach klasy Database. Kolejna część zawiera funkcje wypisujące nam na ekran poszczególne menu oraz wypisujące tabele zawierające nasze obiekty. Ostatnią część stanowią funkcje które wykonują poszczególne funkcjonalności programu oraz przejścia pomiędzy nimi.

Ostatnim i najmniejszym plikiem jest CAR-RENT.py. Zawiera on tylko jedną funkcję start, która uruchamia nasz program - tworzy instancje klasy Database, wczytuje dane z plików oraz przechodzi do funkcji main\_menu zawartej w pliku operations.py. Po wyjściu, zapisuje ona dane do plików i kończy swoje działanie.