

# 1. Klasa Vehicle (abstrakcyjna baza dla pojazdów)

## **Pola:**

licensePlate – numer rejestracyjny pojazdu (np. KR12345)

entryTime – moment, w którym pojazd wjechał na parking

## **Metody:**

getLicensePlate() – zwraca numer rejestracyjny pojazdu

getEntryTime() – zwraca godzinę i datę wjazdu na parking

setEntryTime(LocalDateTime entryTime) – ustawia nowy czas wjazdu

getHourlyRate() – zwraca ile kosztuje jedna godzina parkowania (każdy typ pojazdu ustala swoją stawkę)

getType() – zwraca nazwę typu pojazdu, np. „Samochód” lub „Van dostawczy”

toString() – zwraca krótki tekst opisujący pojazd i jego numer rejestracyjny

## 2. Klasa Car (samochód osobowy)

### **Stałe:**

HOURLY\_RATE = 5.0 – stała cena za godzinę parkowania (5 jednostek)

### **Metody:**

getHourlyRate() – zwraca wartość 5.0 jako opłatę za godzinę

getType() – zwraca tekst „Samochód”

### 3. Klasa DeliveryVan (van dostawczy)

#### **Stałe:**

HOURLY\_RATE = 8.0 – cena za godzinę dla vana (8 jednostek)

#### **Metody:**

getHourlyRate() – zwraca wartość 8.0 jako opłatę za godzinę

getType() – zwraca tekst „Van dostawczy”

## 4. Klasa ParkingSpot (pojedyncze miejsce parkingowe)

### **Pola:**

spotNumber – numer miejsca parkingowego (np. 1, 2, 3...)

parkedVehicle – pojazd, który jest zaparkowany w tym miejscu (może być pusty)

### **Metody:**

isOccupied() – sprawdza, czy ktoś stoi na tym miejscu

parkVehicle(Vehicle vehicle) – przypisuje pojazd do miejsca (czyli go zaparkowuje)

removeVehicle() – usuwa pojazd z miejsca (czyli robi je wolne)

getParkedVehicle() – zwraca pojazd, który stoi w tym miejscu

getSpotNumber() – zwraca numer tego miejsca

toString() – zwraca tekst z informacją czy miejsce jest zajęte i przez kogo

## 5. Klasa ParkingLot (główny parking – wszystkie miejsca)

### **Pola:**

capacity – ile miejsc ma cały parking

spots – lista wszystkich miejsc parkingowych (ParkingSpot)

parkingHistory – zapamiętuje, jakie pojazdy były na parkingu i kiedy wyjechały

dailyRecords – lista pojazdów, które opuściły parking danego dnia

dtf – format daty i godziny, np. „2025-06-04 15:00:00”

## **Metody publiczne:**

`getAvailableSpots()` – mówi ile miejsc jest teraz wolnych

`enterVehicle(Vehicle vehicle)` – próbuje zaparkować pojazd; zwraca `true`, jeśli się udało

`exitVehicle(String licensePlate, LocalDateTime exitTime)` – wypuszcza pojazd, oblicza ile zapłaci, i zapisuje dane

`listParkedVehicles()` – zwraca listę wszystkich pojazdów, które są teraz zaparkowane

`generateDailyReport()` – tworzy raport z dzisiejszych wyjazdów i pokazuje ile zarobiono

`clearDailyReport()` – usuwa dzisiejszy raport (tak jakby zaczął się nowy dzień)

## **Metody prywatne:**

`findFirstAvailableSpotIndex()` – szuka pierwszego wolnego miejsca i zwraca jego numer

`findVehicleSpot(String licensePlate)` – szuka, gdzie stoi pojazd o podanym numerze rejestracyjnym

`cloneVehicle(Vehicle vehicle)` – tworzy kopię pojazdu (potrzebne do zapisu historii)

## 6. Klasa ParkingPayment (informacje o opłacie po wyjeździe)

### **Pola:**

record – zapis z informacją kiedy pojazd wjechał, wyjechał i ile zapłacił

hoursParked – ile godzin pojazd stał na parkingu

fee – ile trzeba było zapłacić za parkowanie

### **Metody:**

getRecord() – zwraca cały zapis o parkowaniu tego pojazdu

getHoursParked() – ile godzin stał

getFee() – ile kosztowało

## 7. Klasa ParkingRecord (pojedynczy zapis o parkowaniu)

### **Pola:**

licensePlate – numer rejestracyjny pojazdu

vehicleType – typ pojazdu (np. "Samochód")

entryTime – czas wjazdu na parking

exitTime – czas wyjazdu z parkingu

fee – opłata, która została naliczona

### **Metody:**

getLicensePlate() – numer rejestracyjny

getVehicleType() – typ pojazdu

getEntryTime() – data i godzina wjazdu

getExitTime() – data i godzina wyjazdu

getFee() – ile zapłacono



## 8. Klasa Program (główna aplikacja z menu)

### **Pola:**

scanner – pozwala czytać dane od użytkownika (z konsoli)

parkingLot – obiekt reprezentujący cały parking

currentSimTime – aktualny czas w symulacji (można go przesuwąć do przodu)

## **Metody:**

`run()` – uruchamia menu i pozwala użytkownikowi wybierać opcje

`printWelcome()` – pokazuje powitanie i aktualny czas symulacji

`printMenu()` – wyświetla listę dostępnych opcji (np. zaparkuj, pokaż raport)

`readInt(String prompt)` – pobiera liczbę całkowitą od użytkownika

`readNonEmptyString(String prompt)` – pobiera tekst, który nie może być pusty

`handleVehicleEntry()` – obsługuje dodanie pojazdu na parking

`handleVehicleExit()` – obsługuje wyjazd pojazdu z parkingu

`isVehicleAlreadyParked(String licensePlate)` – sprawdza czy pojazd już stoi na parkingu

`showAvailability()` – pokazuje ile jest wolnych miejsc

`listParkedVehicles()` – wypisuje wszystkie zaparkowane pojazdy

`generateDailyReport()` – drukuje raport ile pojazdów wyjechało i ile zarobiono

`clearDailyReport()` – czyści dane z dzisiaj (nowy dzień)

`advanceTime()` – przesuwa czas symulacji o wybraną ilość godzin i minut

`main()` – uruchamia program z parkingiem o pojemności 20 miejsc