# **Московский государственный технический университет имени Н. Э. Баумана.**

Факультет “Радиотехнический”

Кафедра ИУ5 ”Системы обработки информации и управления”

Курс “Парадигмы и конструкции языков программирования”

Отчет по домашнему заданию

Выполнила:

Студент группы РТ5-31Б

Губанова В.Е

Подпись и дата:

Проверил:

Преподаватель кафедры ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва,2024 г.

### Текст программы:

### Main.py

import abc

class User:

def \_\_init\_\_(self, user\_id, name, balance):

self.user\_id = user\_id

self.name = name

self.balance = balance

self.rental\_history = []

self.rental = False

self.now\_p = None

def top\_up\_balance(self, amount):

self.balance += amount

def get\_rental\_history(self):

print(self.rental\_history)

return self.rental\_history

def rent\_car(self, car, duration\_minutes, distance\_km):

if self.balance >= 0 and self.rental == False and car.status == 'available':

if isinstance(car, StandardCar):

if (car.fuel\_consumption \* distance\_km) / 100 <= car.current\_fuel:

self.rental = True

car.status = 'rented'

self.balance -= duration\_minutes \* car.price\_per\_minute

dict = {}

dict['car'] = car

dict['duration'] = duration\_minutes

dict['distance'] = distance\_km

self.now\_p = dict

return True

elif isinstance(car, ElectricCar):

if (car.battery\_capacity \* distance\_km) / 100 <= car.current\_charge:

self.rental = True

car.status = 'rented'

self.balance -= duration\_minutes \* car.price\_per\_minute

dict = {}

dict['car'] = car

dict['duration'] = duration\_minutes

dict['distance'] = distance\_km

self.now\_p = dict

return True

return False

def end\_rental(self):

if self.now\_p:

self.now\_p['car'].status = 'available'

if isinstance(self.now\_p['car'], StandardCar):

self.now\_p['car'].current\_fuel -= (self.now\_p['car'].fuel\_consumption / 100) \* self.now\_p['distance']

elif isinstance(self.now\_p['car'], ElectricCar):

self.now\_p['car'].current\_charge -= (self.now\_p['car'].energy\_consumption / 100) \* self.now\_p['distance']

self.rental\_history.append(self.now\_p)

self.rental = False

self.now\_p = None

class Car(metaclass=abc.ABCMeta):

def \_\_init\_\_(self, model, registration\_number, price\_per\_minute):

self.model = model

self.registration\_number = registration\_number

self.price\_per\_minute = price\_per\_minute

self.status = 'available'

@abc.abstractmethod

def refill(self):

pass

class StandardCar(Car):

def \_\_init\_\_(self, model, registration\_number, price\_per\_minute, fuel\_capacity, fuel\_consumption, current\_fuel):

super().\_\_init\_\_(model, registration\_number, price\_per\_minute)

self.fuel\_capacity = fuel\_capacity

self.fuel\_consumption = fuel\_consumption

self.current\_fuel = current\_fuel

def refill(self):

self.current\_fuel = self.fuel\_capacity

class ElectricCar(Car):

def \_\_init\_\_(self, model, registration\_number, price\_per\_minute, battery\_capacity, energy\_consumption, current\_charge):

super().\_\_init\_\_(model, registration\_number, price\_per\_minute)

self.battery\_capacity = battery\_capacity

self.energy\_consumption = energy\_consumption

self.current\_charge = current\_charge

def refill(self):

self.current\_charge = self.battery\_capacity

# Пример проверки работы каршеринга

def main():

user1 = User(user\_id=1, name="Максим", balance=100)

user2 = User(user\_id=2, name="Настя", balance=50)

car1 = StandardCar(model="Toyota Camry", registration\_number="ABC123", price\_per\_minute=1.0,

fuel\_capacity=50, fuel\_consumption=8, current\_fuel=40)

car2 = ElectricCar(model="Tesla Model 3", registration\_number="XYZ789", price\_per\_minute=1.5,

battery\_capacity=75, energy\_consumption=15, current\_charge=60)

# Пользователь 1 арендует стандартный автомобиль на 30 минут и проезжает 100 км

print("Пользователь 1 арендует машину:")

if user1.rent\_car(car1, duration\_minutes=30, distance\_km=100):

print(f"Пользователь {user1.name} успешно арендовал {car1.model}. Оставшийся баланс: {user1.balance}")

else:

print(f"Пользователь {user1.name} не может арендовать {car1.model}.")

# Пользователь 2 пытается арендовать электрический автомобиль на 20 минут и проехать 80 км

print("\nПользователь 2 арендует электрический автомобиль:")

if user2.rent\_car(car2, duration\_minutes=20, distance\_km=80):

print(f"Пользователь {user2.name} успешно арендовал {car2.model}. Оставшийся баланс: {user2.balance}")

else:

print(f"Пользователь {user2.name} не может арендовать {car2.model}.")

# Пользователь 1 завершает аренду

print("\nПользователь 1 завершил аренду:")

user1.end\_rental()

print(f"Пользователь {user1.name} история аренды: {user1.get\_rental\_history()}")

# Проверяем состояние автомобиля после аренды

print(f"Машина {car1.model} статус: {car1.status}, оставшееся топливо: {car1.current\_fuel}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

Результат

