Wypożyczalnia pojazdów
Projekt grupowy

Michał Brus, Łukasz Tumialis, Rafał Bednarz

8 czerwca 2020

1 Opis

Program symuluje wypożyczalnię samochodów.

2 Klasy i metody

Program składa się z następujących klas:

1. VehicleRental - główna klasa
2. Customer - klasa klienta
3. Vehicle - klasa pojazd:
    - Car - klasa pochodna samochód
    - Truck - klasa pochodna ciężarówka
4. MyStack - klasa stosu

Klasy Vehicle, Customer i VehicleRental są ze sobą zaprzyjaźnione.

2.1 VehicleRental

Klasa VehicleRental składa się z 3 wektorów:

- CustomerList
- CarList
- TruckList

i stosu, na którym zapisujemy historię operacji.

Metody klasy VehicleRental:

- Menu - metoda odpowiadająca za interaktywne menu
- ShowOperationsHistory - wyświetla historię operacji na obiektach
- Rent - odpowada za proces wypożyczenia pojazdu

- Payment - proces opłaty
- Return - zwrot pojazdu do wypożyczalni
- ShowVehicleList, ShowCustomerList - wyświetlanie listy pojazdów/klientów
- ShowInfo - funkcja zaprzyjaźniona, która wyświetla szczegółowe informacje o danym obiekcie

2.2 Customer

Klasa posiada typowe informacje jak:

- Imię i nazwisko
- PESEL
- Typ prawa jazdy
- Czy aktualnie wypożycza?
- Obciążenie konta

oraz odpowiedniki metod VehicleRental.

2.3 Vehicle

Klasa posiada informacje ogólne o pojeździe:

- Numer rejestracyjny
- Nazwa
- Rok produkcji
- Stan
- Czy można wypożyczyć?
- Czy pojazd jest sprawny?

Dodatkowo klasy Car i Truck posiadają dodatkowo odpwoiednio *Ilość siedzeń* i *Pojemność*.

2.4 MyStack

Klasa oparta na *template*, w naszym przypadku odpowiada za przechowywanie historii operacji na obiektach.

Metody klasy MyStack:

- Push - dodaje operację do historii
- Pop - usuwa ostanią operację
- Read - wypisuje historię i czyści stos

3 Obsługa wejścia i wyjścia (IO)

3.1 Pliki

Inormacje o klientach i pojazdach są oczytywane i zapisywane do plików txt. Poszczególne kolumny odpowiadają kolejnym zmiennym w klasach. Nazwy nie zawierają spacji, a liczby są int-ami.

```
10 Dariusz Szpakowski 1234567811 B BS00000 0
11 Grzegorz Brzeczyszczykiewicz 1234567812 B BS00000 0
12 Tomasz Hajto 1234567813 B BS00000 0
13 Zdzislaw Przelomiec 1234927807 B BS00000 5
```

Rysunek 1: Format danych w pliku Customers.txt:
IMIĘ...NAZWISKO...PESEL...KAT.PR.J....OBC.KONTA

```
2 Volvo 2018 BS23451 500 5 100 0 1
3 Fiat 2010 CT11111 5 5 100 1 1
4 Mercedes 2000 TK22222 2000 5 100 0 1
5 Maluch 1999 GD33333 10000 4 90 0 1
```

Rysunek 2: Format danych w pliku Cars.txt:
NAZWA...ROK...NR.REJ...KOSZT...SIEDZ..SPRAW...CZYWYPOZ...CZYSPRAW

```
5 Renault 1999 GD53334 10000 200 100 0 1
6 IVECO 1980 CT28886 300 200 100 0 1
7 DAF 2020 BS34967 350 200 100 0 1
8 Jelcz 1970 BS82478 250 200 90 0 1
```

Rysunek 3: Format danych w pliku Trucks.txt:
NAZWA...ROK...NR.REJ...KOSZT...POJBAG...SPRAW...CZYWYP...CZYSPRAW

3.2 Menu

Menu zawiera mechanizmy zabezpieczające przed nieprawidłowymi danymi wejściowymi.

3.3 Makefile

Program można uruchomić z katalogu w terminalu przy użyciu *make*:

- *make all* - tryb normalny (interaktywne menu)
- *make test* - tryb testowy
- *make clean* - usuwanie plików wykonywalnych

4 Podział pracy

Michał Brus:

- Wstępne przygotowanie deklaracji klas i metod.
- Metody VehicleRental: Add, Find, Delete, ShowOperationsHistory, Show...List, ShowInfo
- Klasa MyStack
- Main
- Makefile

Łukasz Tumialis:

- Klasa Vehicle
- Testy
- Metody Load i Export oraz pliki z danymi
- Metody VehicleRental: Rent, Payment, Return

Rafał Bednarz

- Klasa Customer
- Funkcje odpowiadające za IO
- Interaktywne menu
- Testy