**Dokumentacja do programu**

1. **Eduard Ulasenka**

*Projekt 3* Napisać spis pojazdów (lub odpowiednio rozbudować poprzedni program (zad. 2)) zorganizowanych w katalogach rodzajów pojazdów (dowolna nazwa, np. ciężarowe, diesel, do 20 ton itp.). Zarówno liczba katalogów jak i samochodów nie jest ograniczona. W ramach każdego rodzaju pojazdu pozycje opisywane są jak w zad 2, tzn.: imię i nazwisko użytkownika, unikatowy nr rejestracyjny i rok produkcji. Dany pojazd może przynależeć do kilku katalogów jednocześnie lub do żadnego. Program powinien umożliwiać:  
  
-    dodawanie nowych i modyfikację już istniejących pozycji,  
-       utworzenie nowego lub usunięcie istniejącego katalogu,  
-       zmianę nazwy dowolnego katalogu,  
-       przenoszenie określonego pojazdu z jednego katalogu do drugiego,  
-       dodawanie i usuwanie określonego pojazdu z danego katalogu,  
-       usuwanie określonego pojazdu z całego spisu,  
-       wyszukiwanie i wyświetlanie pojazdów wg zadanych  parametrów – imię i nazwisko użytkownika, nr rejestracyjny i rok produkcji (np.: wyświetl wszystkie pojazdy z roku 1991), wyświetlone pozycje powinny zawierać pełną informację, tzn. imię i nazwisko użytkownika, nr rejestracyjny, rok produkcji i rodzaje pojazdów, do których należą  
-       wyświetlanie zawartości dowolnie wybranego katalogu  
-       wyświetlanie istniejących katalogów rodzajów pojazdów  
-       wyświetlanie istniejących pojazdów  
  
Program powinien umożliwiać zapis danych w trzech plikach tekstowych (rozszerzenie \*.txt) umożliwiających odczytywanie i zapisywanie oraz modyfikację danych przy użyciu dowolnego edytora tekstowego:  
1. plik zawierający same katalogi – katalog.txt  
2. plik zawierający rekordy – pozycje.txt  
3. plik zawierający kompletne dane (rekordy+katalogi+łącza) - baza.txt  
  
Każdy plik może być zapisywany i odczytywany niezależnie poprzez wydawanie odpowiednich komend np.: save katalog.txt lub load baza.txt. Wczytanie pliku pozycje.txt skutkuje dopisaniem nowych rekordów jako nienależących do żadnego katalogu (identyfikacja po numerze rej., pojazdy których nr istnieje w bazie są ignorowane). Wczytanie pliku katalog.txt skutkuje dopisaniem nowych rodzajów pojazdów (identyfikacja po nazwie, rodzaje pojazdów których nazwa istnieje w bazie są ignorowane). Po wczytaniu pliku baza.txt dotychczas istniejąca struktura zostaje usunięta.  
  
Program powinien reagować i być odporny na nielogiczne wprowadzanie danych (np. rok produkcji: ABCD). Niedopuszczalne jest utworzenie dwóch katalogów o tej samej nazwie.

1. Program posiada menu użytkownika.

*Menu zawiera następujące opcje:*

1. Dodać pozycję
2. Zmienić pozycję
3. Dodać katalog
4. Usunąć katalog
5. Zmienić nazwę kataloga
6. Przenieść określony pojazd z jednego katalogu do innego
7. Dodać pozycję do katalogu
8. Usunąć pozycję z wybranego katalogu
9. Usunąć pozycję z całego spisu
10. Wyszukać pojazd wg zadanych parametrów
11. Wyświetlić zawartość waybranego katalogu
12. Wyświetlić istniejące katalogi
13. Wyświetlić wszystkie pozycje
14. Manipulacje nad plikami
15. Wyjście z programu
16. Program został napisany w oparciu na użycie dynamicznych struktur danych. W szczegolności przy użyciu list jednokierunkowych, list zagnieżdżonych.

Pierwsza lista zawiera same pozycje.

Najtrudniejszym elementem programu jest realizacja drugiej lisy, która jest naszym spisem. Jej elementy są katalogami. Każdy katalog ma swoją unikatową nazwę, oraz drugą listę,zagnieżdżoną. Czyli mamy do czynienia z listami w liście.

Schemat ten jest przedstawiony na rysunku 1.

1. Program działa w sposób następujący:  
    Przy dodawaniu lub usunięciu pozycji zostaje przydzielana pamięć wartość której odpowiada rozmiarowi tworzonego elementu. W przypadku usunięcia pamięć zostanie zwolniona. Wszystkie pozycje są zapisywane w pliku pozycje.txt.  
   Podobnie wygląda dodawanie katalogu. Nazwe wszystkich katalogów są zapisywane w pliku katalog.txt, kompletne dane – w pliku baza.txt
2. Przy włączeniu programu będą odczytane te pliki. Na podstawie ich zawartości będą utworzone listy. Z którymi będziemy mogli pracować. Podczas działania programu te pliki są otwierane i zapisywane ponownie, np. jeżeli użytkownik wprowadza jakieś zmiany. Pliki mogą być odczytywane i zapisywane niezależnie, poprzez wydawanie odpowiednich komend (jak w poleceniu).
3. Program jest odporny na nielogiczne wprowadzenie danych. Tak np przy przyswojeniu zmiennej typu int wartości typu char program wyświetla komunikat błędu i prosi użytkownika o ponowne podanie danych.
4. Niemożliwe jest tworzenie pozycji zawierającej pojazd o takim samym numerze rejestracyjnym. Zostanie w tym przypadku wyświetlony odpowiedni komunikat.  
   Podobnie jest dla katalogu. Nazwa katalogu jest unikatowa.
5. To samo dotyczy podczas zmiany nazwy katalogu lub numeru rejestracyjnego już istniejących pozycji.  
   Użytkownik proszony o ponowne wpisanie danych.
6. Program prawdopodobnie nie działa przy użyciu wartości większych od maksimów i mniejszych od minimów zakresów typów danych.

*Rysunek 1*

