НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

Звіт

про виконання лабораторної роботи № 14 з дисципліни «Алгоритмізація та програмування»

Виконав:

студент I курсу, групи КА-07

Лещинський Богдан Денисович

Прийняв:

Київ — 2020

Завдання.

Створити масив структур. Кожна структура складається з таких елементів: марка автомобіля, тип (вантажний або легковий), номерний знак, строк служби. Для легкових вказується колір, для вантажних - вантажність. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити номерні знаки автомобілів, вантажність котрих не менше за задану; 2) вивести марки автомобілів заданого типу і вказаного кольору; 3) вилучити з файлу відомості про автомобілі, строк служби яких перевищує п'ять років.

Аналіз умови задачі.

Виходячи з умови завдання потрібно розбити алгоритм на наступні базові підзадачі :

- створення функції додавання елементів у масив структур та запис збережених у масиві даних у файл

- створення функції змінення параметрів окремих елементів масива або видалення окремих елементів

- створення функції видалення усіх даних з файлу

- створення функції знаходження номерних знаків машин з заданою мінімальною вантажністю

- створення функції отримання марок машин за заданим типом та кольором

- створення функції видалення даних за файлу про машини зі строком використання більше п’яти років

- створення інтерфейсу меню для створення та використання файлу користувачем за допомогою вище згаданих функцій

Код програми.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <cstdlib>

#include <string>

using namespace std;

struct LightCar {

char\* color = new char[30];

};

struct Truck {

int loadCapacityKg;

};

union CarType {

LightCar lightcar;

Truck truck;

CarType() {}

~CarType() {}

};

enum Type{

LCAR, TRUCK

};

struct Car {

char\* mark = new char[40];

CarType type;

int licencePlate;

int serviceLifeYears;

Type typ;

};

FILE\* file;

int entitiesNum = 0;

void addData() {

system("cls");

file = fopen("cars.dat", "ab");

int n;

cout << "How many cars do you want to add to the file: ";

cin >> n;

cout << endl;

getchar();

Car\* carArr = new Car[n];

entitiesNum += n;

int i = 0;

int carsCounter = 0;

do {

system("cls");

carsCounter++;

cout << "Car : " << carsCounter<< "\n\n";

cout << "Enter car`s mark: \n";

fgets(carArr[i].mark, 40, stdin);

int carsType;

cout << "Choose car`s type: 1. Light car, 2. Truck; 1/2?\n";

cin >> carsType;

getchar();

if (carsType == 1) {

carArr[i].typ = LCAR;

cout << "Enter car`s color: \n";

carArr[i].type.lightcar = LightCar();

fgets(carArr[i].type.lightcar.color, 30, stdin);

}

else if (carsType == 2) {

carArr[i].typ = TRUCK;

cout << "Enter car`s load capacity: \n";

carArr[i].type.truck = Truck();

cin >> carArr[i].type.truck.loadCapacityKg;

getchar();

}

else {

cerr << "Wrong option entered!\n";

}

cout << "Enter car`s licence plate: \n";

cin >> carArr[i].licencePlate;

getchar();

cout << "Enter car`s service life in years: \n";

cin >> carArr[i].serviceLifeYears;

getchar();

fwrite(&carArr[i], sizeof(Car), 1, file);

i++;

} while (i != n);

fclose(file);

delete[] carArr;

getchar();

}

void editData() {

system("cls");

Car\* carArr = new Car[entitiesNum];

int counter = 0;

if (!(file = fopen("cars.dat", "rb"))) {

cerr << "File does not exist!\n";

remove("cars.dat");

exit(1);

}

else {

while (!feof(file)) {

fread(&carArr[counter], sizeof(carArr[counter]), 1, file);

counter++;

}

}

fclose(file);

char cont;

do {

system("cls");

cout << "Current data in the file: \n";

for (int i = 0; i < entitiesNum; i++) {

if (carArr[i].typ == LCAR) cout << "---Car№" << i + 1 << " " << "Mark: " << carArr[i].mark << " " << "Type: light car " << "Licence plate: " << carArr[i].licencePlate << " " << "Service life in years: " << carArr[i].serviceLifeYears << " " << "Color: " << carArr[i].type.lightcar.color << endl;

else cout << "---Car№" << i + 1 << " " << "Mark: " << carArr[i].mark << " " << "Type: truck " << "Licence plate: " << carArr[i].licencePlate << " " << "Service life in years: " << carArr[i].serviceLifeYears << " " << "Load capacity in Kg: " << carArr[i].type.truck.loadCapacityKg << endl;

}

cout << endl;

int car;

int edit;

cout << "Which car`s data do want to edit: ";

cin >> car;

cout << "\nHow do you want to edit the data : 1.delete 2.change\n";

cin >> edit;

if (edit == 1) {

for (int i = car-1; i < entitiesNum-1; i++) {

carArr[i] = carArr[i + 1];

}

entitiesNum--;

cout << "Selected car deleted\n";

cout << "Continue editing? y/n\n";

getchar();

cont = getchar();

if (cont == 'y') {

continue;

}

else cerr << "Wrong option entered!\n";

}

else if (edit == 2) {

do {

cout << "What car`s parameters are to be changed?\n";

if (carArr[car - 1].typ == LCAR) {

cout << "1.Mark; 2.Licence Plate; 3.Service life in years; 4.Color\n";

cin >> edit;

switch (edit) {

case 1:

getchar();

cout << "Enter chages to the selected car`s mark: ";

fgets(carArr[car - 1].mark, 40, stdin);

cout << endl;

break;

case 2:

getchar();

cout << "Enter chages to the selected car`s licence plate: ";

cin >> carArr[car - 1].licencePlate;

cout << endl;

break;

case 3:

getchar();

cout << "Enter chages to the selected car`s life service in years: ";

cin >> carArr[car - 1].serviceLifeYears;

cout << endl;

break;

case 4:

getchar();

cout << "Enter chages to the selected car`s color: ";

fgets(carArr[car - 1].type.lightcar.color, 30, stdin);

cout << endl;

break;

default:

cerr<<"Wrong optionentered\n";

break;

}

cout << "Continue changes? y/n\n";

cont = getchar();

}

else {

cout << "1.Mark; 2.Licence Plate; 3.Service life in years; 4.Load capasity\n";

cin >> edit;

switch (edit) {

case 1:

getchar();

cout << "Enter chages to the selected car`s mark: ";

fgets(carArr[car - 1].mark, 40, stdin);

cout << endl;

break;

case 2:

getchar();

cout << "Enter chages to the selected car`s licence plate: ";

cin >> carArr[car - 1].licencePlate;

cout << endl;

break;

case 3:

getchar();

cout << "Enter chages to the selected car`s life service in years: ";

cin >> carArr[car - 1].serviceLifeYears;

cout << endl;

break;

case 4:

getchar();

cout << "Enter chages to the selected car`s load capacity: ";

cin >> carArr[car - 1].type.truck.loadCapacityKg;

cout << endl;

break;

default:

cerr << "Wrong optionentered\n";

break;

}

cout << "Continue changes? y/n\n";

cont = getchar();

}

} while (cont != 'n');

}

} while (cont != 'n');

file = fopen("cars.dat", "wb");

for (int i = 0; i < entitiesNum; i++) {

fwrite(&carArr[i], sizeof(carArr[i]), 1, file);

}

fclose(file);

delete[] carArr;

}

void deleteData() {

remove("cars.dat");

entitiesNum = 0;

cout << "All data das been deleted from the file\n";

system("pause");

getchar();

}

void getLicencePlates() {

system("cls");

Car\* carArr = new Car[entitiesNum];

int counter = 0;

if (!(file = fopen("cars.dat", "rb"))) {

cerr << "File does not exist!\n";

remove("cars.dat");

exit(1);

}

else {

while (!feof(file)) {

fread(&carArr[counter], sizeof(carArr[counter]), 1, file);

counter++;

}

}

fclose(file);

int minLoadCapacity;

cout << "Enter manimal load capacity: ";

cin >> minLoadCapacity;

cout << endl;

counter = 0;

cout << "Licence plates of the cars with entered minimal load capacity: \n";

for (int i = 0; i < entitiesNum; i++) {

if (carArr[i].typ == TRUCK && carArr[i].type.truck.loadCapacityKg >= minLoadCapacity) {

cout << carArr[i].licencePlate << endl;

counter++;

}

}

if (counter == 0) cout << "Ups..No cars have matched the entered minimal load capacity(\n";

system("pause");

delete[] carArr;

getchar();

}

void getMarks() {

system("cls");

Car\* carArr = new Car[entitiesNum];

int counter = 0;

if (!(file = fopen("cars.dat", "rb"))) {

cerr << "File does not exist!\n";

remove("cars.dat");

exit(1);

}

else {

while (!feof(file)) {

fread(&carArr[counter], sizeof(carArr[counter]), 1, file);

counter++;

}

}

fclose(file);

int carT;

Type chosenType;

char colorf[30];

cout << "Enter car`s type: 1.Light car 2.Truck \n";

cin >> carT;

if (carT == 1) {

chosenType = LCAR;

}

else if (carT == 2) {

chosenType = TRUCK;

}

else cerr << "Wrong option entered!\n";

cout << endl;

getchar();

cout << "Enter car`s color: ";

fgets(colorf, 30, stdin);

cout << endl;

counter = 0;

cout << "Matched cars` marks: \n";

for (int i = 0; i < entitiesNum; i++) {

if (carArr[i].typ == chosenType || carArr[i].type.lightcar.color == colorf) {

cout << carArr[i].mark;

counter++;

}

}

if (counter == 0) cout << "Ups..No cars have matched the entered parameters(\n";

system("pause");

delete[] carArr;

}

void deleteOld() {

system("cls");

Car\* carArr = new Car[entitiesNum];

int counter = 0;

if (!(file = fopen("cars.dat", "rb"))) {

cerr << "File does not exist!\n";

remove("cars.dat");

exit(1);

}

else {

while (!feof(file)) {

fread(&carArr[counter], sizeof(carArr[counter]), 1, file);

counter++;

}

}

fclose(file);

cout << "Deleteing cars with more than 5 years of cervice life...-\n";

counter = 0;

for (int i = 0; i < entitiesNum; i++){

if (carArr[i].serviceLifeYears > 5) {

for (int j = i; j < entitiesNum - 1; j++) {

carArr[j] = carArr[j + 1];

}

entitiesNum--;

counter++;

}

}

cout << counter << " cars have benn deleted\n";

file = fopen("cars.dat", "wb");

for (int i = 0; i < entitiesNum; i++) {

fwrite(&carArr[i], sizeof(carArr[i]), 1, file);

}

fclose(file);

system("pause");

delete[] carArr;

getchar();

}

int main(){

while (1) {

system("cls");

cout << "-----Options-----\n";

cout << "1.Add data to the file\n";

cout << "2.Edit data in the file\n";

cout << "3.Delete data from the file\n";

cout << "4.Get licence plates of the cars with entered minimal load capaxity\n";

cout << "5.Get marks of the car with entered type and color\n";

cout << "6.Delete data about the cars with more than 5 years of life service\n";

cout << "7.Exit\n";

int key;

cout << "\n\n" << "Enter an option: ";

cin >> key;

cout << endl;

switch (key) {

case 1:

addData();

break;

case 2:

editData();

break;

case 3:

deleteData();

break;

case 4:

getLicencePlates();

break;

case 5:

getMarks();

break;

case 6:

deleteOld();

break;

case 7:

remove("cars.dat");

exit(0);

break;

default:

remove("cars.dat");

cerr << "Wrong option entered!\n";

exit(1);

break;

}

}

return 0;

}

Результат виконання програми(інтерфейс меню з опціями).

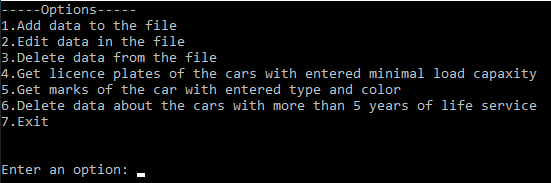


Рис.1 - результат виконання програми

Висновок.

У результаті виконання даної лабораторної роботи було створено програму, що, дозволяє за допомогою спеціально створених функцій створювати файл та зберігати у ньому інформацію по заданій структурі машини. Також були реалізовані можливості виконувати запити доповнення файлу, видалання всіх даних з файлу, зміни параметрів вже існуючих даних, або їх часткове видалення, запити знаходження номерних знаків машин з заданою мінімальною вантажністю, отримання марок машин за заданим типом та кольором. Був створений інтерфейс меню для користувача.