ОДЕССКИЙ НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИНСТИТУТ КОМПЬЮТЕРНЫХ СИСТЕМ

Кафедра информационных систем и технологий

**Лабораторная работа №9**

по дисциплине: «Алгоритмизация и программирование»

Вариант № 1

**Выполнил:**

АД-192

Березовский В. А.

**Проверили:**

Одесса 2020

**Тема:** работа с бинарными файлами.

**Цель:** научиться правильно, работать с бинарными файлами.

**Ход работы**

**Задание:**

Создать структуру, согласно варианту, создали пять объектов соответствующие нашей структуре, заполнить их данными и записать в бинарный файл. С помощью прямого доступа, найти в этом файле 2-ую и 5-ую запись, вывести их на экран. В 3-ей записи изменить одно из полей и записать в файл.

«Владелец автомобиля»: фамилия имя отчество; номер телефона; марка автомобиля; номер автомобиля; номер техпаспорта.

**Листинг:**

#include <iostream>

#include <fstream> // подключаем библиотеку для работы с потоками

#include <string> // подключаем библиотеку для работы со строками

using namespace std;

struct Car\_owner { // создаем структуру с массивами для заполнения данных

char FullName[50]; // Ф.И.О.

char Phone[15]; // номер телефона

char Car\_model[10]; // марка машины

char Car\_number[15]; // номер машины

char Certificate\_number[40]; // номер техпаспорта

};

int main() {

setlocale(LC\_ALL, "en");

Car\_owner Person[6]; // создаем массив который будет содержать все массивы в структуре

Person[0] = { "Berezovskiy Vladislav Alexsandrovich", "0966192686", "BMW", "6936", "37472341" }; // данные 1 человека

Person[1] = { "Petrov Vladimir Victorovich", "0973430832", "Hunday", "7362", "47730971" }; // данные 2 человека

Person[2] = { "Panchenko Ivan Igorovich", "06749356743", "Honda", "8463", "7248290001" }; // данные 3 человека

Person[3] = { "Chechelnitckiy Maksim Nikolaevich", "0966236422", "Bugati", "7489", "374839991" }; // данные 4 человека

Person[4] = { "Arman Maksim Leonardovich", "0862836744", "Lada", "6731", "43440077" }; // данные 5 человека

ofstream things("File.txt", ios::binary); // открываем поток для записи в бинарном файле

if (!things.is\_open()) { // проверка на открытие файла

cout << "Error!" << endl << endl; // если файл не откроеться то выведит в консоль это сообщение

}

else { // если откроеться то будем работь с ним

things.write((char\*)&Person[0], sizeof(Person[0])); // записываем в файл данные 1 человека

things.write((char\*)&Person[1], sizeof(Person[1])); // записываем в файл данные 2 человека

things.write((char\*)&Person[2], sizeof(Person[2])); // записываем в файл данные 3 человека

things.write((char\*)&Person[3], sizeof(Person[3])); // записываем в файл данные 4 человека

things.write((char\*)&Person[4], sizeof(Person[4])); // записываем в файл данные 5 человека

}

things.close(); // закрываем файл после работы с записью

fstream file("File.txt", ios::binary | ios::in | ios::out); // открываем поток для работы с бинарным файлом

if(!file.is\_open()) { // открываем поток для записи в бинарном файле

cout << "Error!" << endl << endl; // если файл не откроеться то выведит в консоль это сообщение

}

else { // если откроеться то будем работь с ним

Car\_owner fs; // контейнер

file.seekg(1 \* sizeof(Car\_owner)); // находим вторую запись в файле

file.read((char\*)&fs, sizeof(fs)); // считываем данные и запоминаем их в файл

cout << "Peson[2]: " << fs.FullName << "\t" << fs.Phone << "\t" << fs.Car\_model << "\t" << fs.Car\_number << "\t" << fs.Certificate\_number << endl; // выводим данные на консоль

file.seekg(4 \* sizeof(Car\_owner)); // находим пятую запись в файле

file.read((char\*)&fs, sizeof(fs)); // считываем данные и запоминаем их в файл

cout << "Peson[5]: " << fs.FullName << "\t" << fs.Phone << "\t" << fs.Car\_model << "\t" << fs.Car\_number << "\t" << fs.Certificate\_number << endl; // выводим данные на консоль

int a, fullname, phone, car\_model, car\_number, size; // буфера для дальнейшего использования в формуле

cout << "\nWhat do you want to change?" << endl << endl; // для меню

cout << "1) FullName \n2) Phone \n3) Car model \n4) Car number \n5) Certificate number \n\nEnter the number: "; // меню для выбора

TryAgain: // возврат при неправильном выборе из пункта меню

cin >> a;

switch (a) {

case 1: { // если выберем пункт 1

fullname = 0; // изменения данных

phone = 0; // данные остаються

car\_model = 0; // данные остаються

car\_number = 0; // данные остаються

size = sizeof(Person[0].FullName); // присваем размер

break; // остановка чтоб не пробегал дальше

}

case 2: { // если выберем пункт 2

fullname = 1; // данные остаються

phone = 0; // изменения данных

car\_model = 0; // данные остаються

car\_number = 0; // данные остаються

size = sizeof(Person[0].Phone); // присваем размер

break; // остановка чтоб не пробегал дальше

}

case 3: { // если выберем пункт 3

fullname = 1; // данные остаються

phone = 1; // данные остаються

car\_model = 0; // изменения данных

car\_number = 0; // данные остаються

size = sizeof(Person[0].Car\_model); // присваем размер

break; // остановка чтоб не пробегал дальше

}

case 4: { // если выберем пункт 4

fullname = 1; // данные остаються

phone = 1; // данные остаються

car\_model = 1; // изменения данных

car\_number = 0; // данные остаються

size = sizeof(Person[0].Car\_number); // присваем размер

break; // остановка чтоб не пробегал дальше

}

case 5: { // если выберем пункт 5

fullname = 1; // данные остаються

phone = 1; // данные остаються

car\_model = 1; // данные остаються

car\_number = 1; // данные остаються

size = sizeof(Person[0].Certificate\_number); // присваем размер

break; // остановка чтоб не пробегал дальше

}

default: { // если вышли за предел меню

cout << "This number does not exist!" << endl; // сообщение если мы не правильно выбрали несуществующего пункта из меню

goto TryAgain; // чтоб вернуться назад к меню для правильного выбора

}

}

char\* cont = new char[size]; // создаем динамический массив чтоб засунуть в него все данные

cout << "Enter the value you want to change: "; // записываем новые данные которые мы хотим изменить

cin >> cont;

file.seekp(2 \* sizeof(Car\_owner) + fullname \* sizeof(Person[2].FullName) + phone \* sizeof(Person[2].Phone) + car\_model \* sizeof(Person[2].Car\_model) + car\_number \* sizeof(Person[2].Car\_number)); // помещаем данные третьего человека по данной формуле

file.write(cont, sizeof(Person[2].FullName)); // записываем данные в файл отталкиваясь от первых данных

file.seekg(2 \* sizeof(Car\_owner)); // получаем изменные данные третьего человека

file.read((char\*)&fs, sizeof(Car\_owner)); // читаем данные их файла и запоминаем их в контейнер

cout << "Person[3]: " << fs.FullName << "\t" << fs.Phone << "\t" << fs.Car\_model << "\t" << fs.Car\_number << "\t" << fs.Certificate\_number << endl; // выводим измененые данные на консоль

delete[] cont; // вытесняем данные с динамической памяти

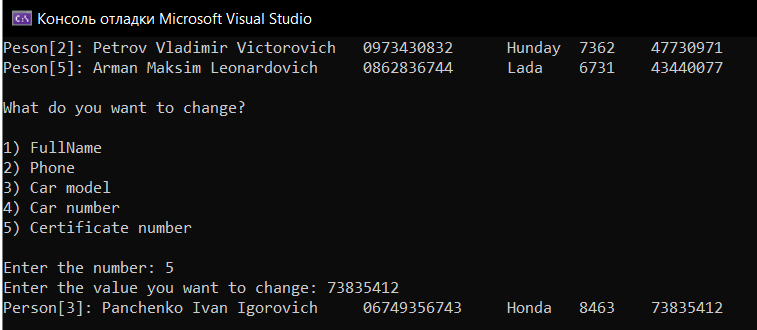
cont = 0; // удаляем висячий указатель

file.close(); // закрываем файл после работы с ним

}

}

**Тестирование программы:**

****

**Вывод:** я научился работать с бинарными файлами. Все задания были успешно закреплены и выполнены и готовы к проверке.