Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**Алгоритмізація та програмування 2: [Процедурне програмування](https://classroom.google.com/u/1/c/NDQ5MDIwMzkxMzg1" \t "https://classroom.google.com/u/1/c/NDQ5MDIwMzkxMzg1/a/NDYwMzEwMzE1MTE4/_self)**

ЗВІТ

до лабораторної роботи №3

«**Структури**»

*(ТЕМА)*

Варіант № 1

Дата «2» травня 2022 Виконав: студент 1 курсу

гр. ТР-15

Руденко Владислав Ігорович

*(П.І.Б.)*

Оцінка «\_\_\_\_\_\_\_\_\_» Перевірив:

Крячок Олександр Степанович

Дата «\_\_\_» \_\_\_\_\_\_\_ 20\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(П.І.Б., підпис)*

Київ – 2022

***Завдання до роботи:***

1. Ознайомитись зі структурами та масивами структур. Дослідити особливості доступу до змінних структурного типу.  
   2. Розробити Блок-схему програмного алгоритму.  
   3. У якості індивідуального завдання необхідно написати код програми, що створює масив структур згідно з індивідуальним завданням та виконує запис даних в файл у вигляді структурних блоків (елементів масиву структур). Ім'я файлу – «імя\_структури».txt.  
   4. Реалізувати читання із файлу «імя\_структури».txt окремих елементів масиву структур за заданим індексом масиву.  
   5. Оформити ЗВІТ до лабораторної роботи згідно вимог та методичних рекомендацій.



Додаток В- 3

***Теоритичні відомості:***  
 **Структура** — це сукупність різнотипних елементів, яким присвоюється одне ім’я (воно може бути відсутнім), що займає одну ділянку пам’яті. Елементи, що складають структуру, на­зиваються полями.

Змінна типу структура, як і будь-яка змінна, повинна бути описана. Цей опис складається з двох кроків: опису шаблону (тобто складу) або типу структури та опису змінних структур­ного типу. Якщо ініціювання виконується в тілі програми, то для звер­нення до імені поля треба спочатку записати ім’я структурної змінної, а потім ім’я поля. Ці обидва записи відокремлюються крапкою і являють собою складене ім’я. Коли при описі структур у деякій функції або в межах ви­димості змінних у різних функціях є багато (але не всі) одна­кових полів, то їх слід об’єднати в окрему структуру. Її можна застосовувати при описі інших структур, тобто поля структури можуть самі бути типу **struct**. Це називається **вкладеністю структур** — її можна використати, наприклад, якщо треба обробляти списки студентів та викладачів університету. Студентські списки містять дані: прізвище та ініціали, дата (день, місяць, рік) народження, група та середній бал успішності, а в списках викладачів присутні такі дані: прізвище, ініціали, дата народження, кафедра, посада, тощо.  
  
 **Масив структур** - Структури часто утворюють масиви. Щоб об’єднати масивну структуру, спочатку необхідно визначити структуру (щоб було визначити агрегатний тип даних), а потім об’єднати змінну масу цього же типу. Наприклад, щоб об’єднати 100-елементний масив структури типу addr, який був визначений раніше, напишіть наступне:

struct addr addr\_list[100];

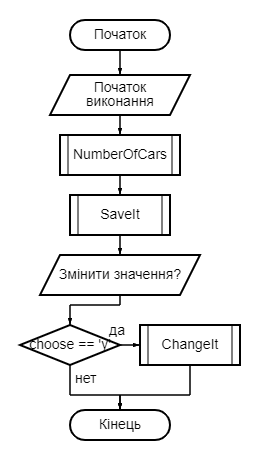
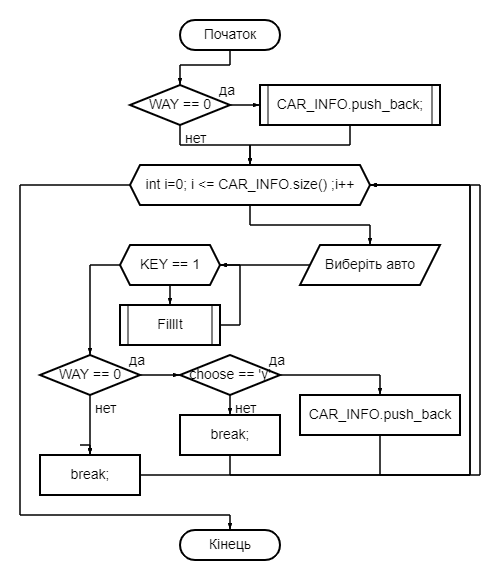
Це створює 100 наборів змінних, кожен із яких організовано так, як визначено в структурі addr.

Щоб отримати доступ до певної структури, вкажіть ім’я маси з індексом. Наприклад, щоб вивести ZIP-код із третьої структури, напишіть наступне:

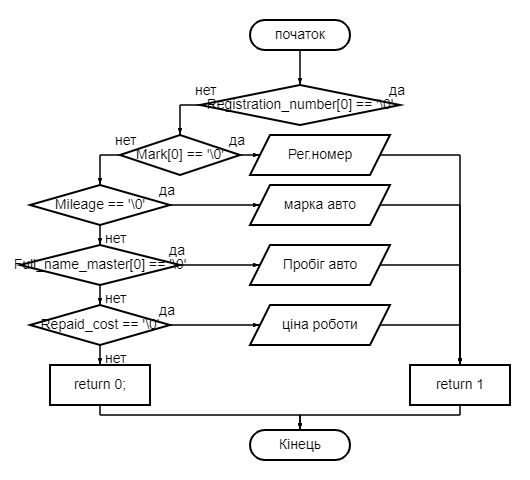
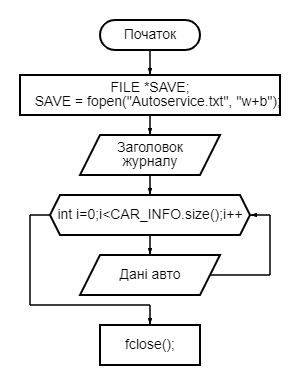
printf("%d", addr\_list[2].zip);

Як і в інших масах змінних, в масивах структурування індексування починається з 0.

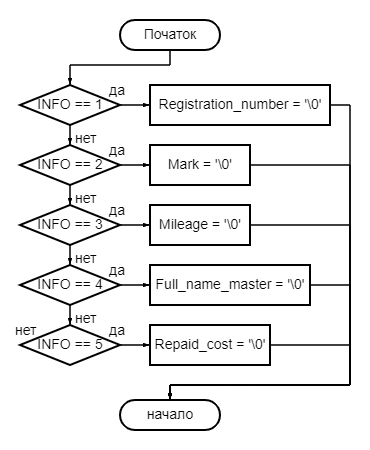
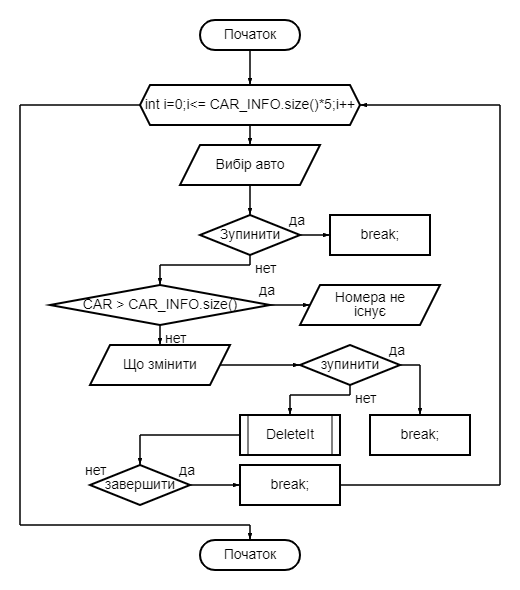
***Блок схема алгоритму:***

*** ***

Блок-Схема 1 (Main) Блок-Схема 2 (NumberOfCars)

Блок-Схема 3 (FillIt) Блок-Схема 4 (SaveIt)

Блок-Схема 5 (DeleteIt) Блок-Схема 6 (ChangeIT)

***Результати роботи:***



***Посилання на програму в repl.it:  
<https://replit.com/join/nbqzljgkjw-hetik>***

***Висновки:***

В ході виконання Лабораторної роботи №3 було ознайомленр зі структурами та масивами структур. Досліджено доступ до змінних структурного типу. Задачею лабораторної роботи було створення програми для запису вхідних даних до файлу з назвою Autoservice.txt , з чим програма впоралась на відмінно, реалізовано перезапис деяких змінних. Також було знайдено та виправлено деяку кількість помилок.

**Програмний код (у текстовому варіанті):**#include <iostream>

#include <chrono>

#include <vector>

#include <string>

using namespace std;

using milliseconds = chrono::duration<long long, micro>;

#define RED "\x1b[31m"

#define BLUE "\x1b[34m"

#define PINK "\033[1;35m"

#define RESET "\x1b[0m"

#define Date\_SIZE 64

struct Autoservice

{

string Registration\_number;

string Mark;

int Mileage;

string Full\_name\_master;

float Repaid\_cost;

}

service;

int FillIt(vector<Autoservice>& CAR\_INFO, int

counter);

void KZ15();

void NumberOfCars(vector<Autoservice>& CAR\_INFO, int WAY);

void SaveIt(vector<Autoservice>& CAR\_INFO);

void DeleteIt(vector<Autoservice>& CAR\_INFO, int CAR, int INFO);

void ChangeIt(vector<Autoservice>& CAR\_INFO);

int main()

{

char choose;

cout << "Лабораторна робота №2. <<Структури>>\n by Руденко Владислав." << "\n\nПрограма передбачає запис даних до \"Журналу Автосервіса\" у вигляді окремого файлу та їх редагування. \n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n" << "Початок виконання...\n" << endl;

vector <Autoservice> CAR\_INFO;

NumberOfCars(CAR\_INFO, 0);

SaveIt(CAR\_INFO);

cout << " Бажаєте ввести зміни?" RED " (Y/N) " RESET ": ";

cin >> choose;

if (choose == 'Y' || choose == 'y')

ChangeIt(CAR\_INFO);

else if (choose != 'N' || choose != 'n')

cout << "The value is perceived as"<<RED<<" N"<< RESET << endl;

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \nВиконання завершено.";

}

int FillIt(vector<Autoservice>& CAR\_INFO, int

counter)

{

int KEY;

if(CAR\_INFO[counter].Registration\_number[0] == '\0')

{

cout << "Необхідно ввести регістраціонний номер авто: ";

cin >> CAR\_INFO[counter].Registration\_number;

return 1;

}

else if(CAR\_INFO[counter].Mark[0] == '\0')

{

cout << "Необхідно ввести марку авто: ";

cin >> CAR\_INFO[counter].Mark;

return 1;

}

else if(CAR\_INFO[counter].Mileage == '\0')

{

cout << "Необхідно ввести пробіг авто: ";

cin >> CAR\_INFO[counter].Mileage;

return 1;

}

else if(CAR\_INFO[counter].Full\_name\_master[0] == '\0')

{

cout << "Необхідно ввести П.І.П майстра, що виконував роботи (через \_): ";

cin >> CAR\_INFO[counter].Full\_name\_master;

return 1;

}

else if(CAR\_INFO[counter].Repaid\_cost == '\0')

{

cout << "Необхідно ввести ціну роботи: ";

cin >> CAR\_INFO[counter].Repaid\_cost;

return 1;

}

else

return 0;

}

void NumberOfCars(vector<Autoservice>& CAR\_INFO, int WAY)

{

if(WAY == 0)

CAR\_INFO.push\_back(service);

for(int i=0; i <= CAR\_INFO.size() ;i++)

{

int lock = 0, KEY = 1;

char choose;

cout << "Запис даних про авто "<< RED <<" №" << i+1 << RESET << " станом на "<< RESET <<" \*\*/05/2022\n";

while(KEY == 1)

KEY = FillIt(CAR\_INFO, i);

KEY = 1;

if (WAY == 0)

{

cout << "\nЗапис даних завершено...\n Бажаєте додати ще запис до журналу? "<< RED <<"(Y/N)" << RESET <<": ";

cin >> choose;

if (choose == 'Y' || choose == 'y')

{

CAR\_INFO.push\_back(service);

cout << endl;

}

else if (choose != 'N' || choose != 'n')

{

cout << "The value is perceived as"<<RED<<" N"<< RESET << endl;

break;

}

}

else if(WAY == 1 && CAR\_INFO.size()-1 == i)

break;

}

}

void SaveIt(vector<Autoservice>& CAR\_INFO)

{

FILE \*SAVE;

SAVE = fopen("Autoservice.txt", "w+b");

fprintf(SAVE, "Журнал виконаної роботи за \*\*/05/2022\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nКількісит записів за день: %i\n", CAR\_INFO.size());

for(int i=0;i<CAR\_INFO.size();i++)

{

fprintf(SAVE, "\nДані автомобіля №%i \n",i+1);

fprintf(SAVE, " Регістраціоний номер авто: %s;\n Марка авто: %s;\n Пробіг авто: %i;\n П.І.П майстра: %s;\n Ціна робіт: %.2f;\n\n", CAR\_INFO[i].Registration\_number.c\_str(), CAR\_INFO[i].Mark.c\_str(), CAR\_INFO[i].Mileage, CAR\_INFO[i].Full\_name\_master.c\_str(), CAR\_INFO[i].Repaid\_cost);

}

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n";

fclose(SAVE);

}

void DeleteIt(vector<Autoservice>& CAR\_INFO, int CAR, int INFO)

{

if(INFO==1)

CAR\_INFO[CAR].Registration\_number = '\0';

else if(INFO == 2)

CAR\_INFO[CAR].Mark = '\0';

else if(INFO == 3)

CAR\_INFO[CAR].Mileage = '\0';

else if(INFO == 4)

CAR\_INFO[CAR].Full\_name\_master = '\0';

else if(INFO == 5)

CAR\_INFO[CAR].Repaid\_cost = '\0';

cout << "\n Успішно видалено!\n";

}

void ChangeIt(vector<Autoservice>& CAR\_INFO)

{

char choose;

int CAR, INFO;

vector <int> NeedChange;

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n";

for(int i=0;i<= CAR\_INFO.size()\*5;i++)

{

cout << "Виберіть номер авто для редагування " RED " (0 для відміни): " RESET << endl;

cin >> CAR;

if(CAR == 0)

break;

else

if(CAR > CAR\_INFO.size())

cout << RED "Заданого авто не інсує" RESET;

else

{

cout << "Що бажаєте змінити?\n Регістраціонний номер - " BLUE " <1> " RESET "\n Марка авто -" BLUE " <2>" RESET "\n Пробіг авто -" BLUE " <3>" RESET "\n П.І.П майстра -" BLUE " <4>" RESET "\n Ціна роботи -" BLUE " <5>" RESET "\n Відміна -" RED " <0>" RESET "\n::";

cin >> INFO;

if(INFO == 0)

break;

else

DeleteIt(CAR\_INFO, CAR-1, INFO);

}

cout << "Зміни завершені?" RED " (Y/N)" RESET ": ";

cin >> choose;

if (choose == 'Y' || choose == 'y')

break;

else if (choose != 'N' || choose != 'n')

cout << "The value is perceived as"<<RED<<" N"<< RESET << endl;

}

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n";

NumberOfCars(CAR\_INFO, 1);

SaveIt(CAR\_INFO);

cout << "Запис даних завершено...";

}