МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпропетровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №5**

**з дисципліни «Основи програмування»**

**на тему: «Розробка програм з циклічною обробкою даних»**

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2019

**Лабораторна робота №5**

**Тема.** Структурний тип даних в мові С++.

**Мета.** Засвоїти поняття складеного типу даних. Отримати практичні навички роботи з інтегрованими типами даних – структурами і масивами структур – мови С++.

**Постановка задачи**

1.Загальна постановка: розробити програму відповідно до обраного варіанту завдання.

2.Індивідуальна постановка:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Предметна область | Дані | Рівень складності | | | | |
| E  (достатньо) | D  (задовільно) | C  (добре) | B  (дуже добре) | A  (відміно) |
| 9 | Квартира  (2) | * прізвище та ініціали власника * адреса (структура з трьох полів – вулиця, будинок, номер квартири) * ціна | * Додавання квартири у список * Виведення списку квартир | Виведення списку квартир вказаної ціни | Пошук квартир з мінімальною ціною | Пошук вказаної квартири в списку | Сортування списку квартир за ціною |

3.Вимоги до програми:

- всі вхідні дані вводяться з клавіатури;

- передбачити перевірку вхідних даних на відповідність діапазону значень і некоректні символи;

- введені дані зберігаються в масиві структур (масив складається з 10 елементів);

- символьні поля структури оголошуються як масив символів;

- інформація, яка зберігається у символьному полі структури, може складатися з декількох слів;

- три перши елементи масиву необхідно ініціалізувати при оголошенні масиву;

- управління виконанням програми здійснюється на основі текстового меню користувача;

- результати роботи програми виводяться на екран.

**4.Вимоги до тексту програми:**

- коментарі щодо призначення програми, її вхідних і вихідних даних;

- коментарі щодо призначення кожного блоку програми, дій окремих операторів для пояснення алгоритму;

- самодокументованість коду: всі ідентифікатори повинні мати назви, що відповідають суті змінних.

**Зовнішні специфікації**

1. Вхідні дані

Вхідні дані:

**Формат вхідних даних**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | Номер меню | menu\_paragraf | Ціле число | 1, 2, 3, 4, 5, 6 |
| 2 | ФІО и вулиця | house[quantity].name  house[quantity].built.street |  | Фомин Ждан Станиславович,  Старокозацька |
| 3 | Номер будинку, квартири та ціни | house[quantity].built.building  house[quantity].built.number  house[quantity].price | Ціле число | 49, 21, 12000 |
| 4 | Пошук за ціною | search\_price | Ціле число | 12000 |

2. Вихідні дані:

**Формат вихідних даних**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | Виведення квартир у список |  |  | №1  ФІО: Сідоров Заур Данилович  Вулиця: Карла Маркса, 555, 55  Ціна: 55000$  №2  ФІО: Семёнов Родион Петрович  Вулиця: Кірова, 485, 6  Ціна: 45000$  №3  ФІО: Фомин Ждан Станиславович  Вулиця: Старокозацька, 49, 21  Ціна: 12000$ |
| 2 | Виведення списку квартир вказаної ціни |  |  | №1  ФІО: Сідоров Заур Данилович  Вулиця: Карла Маркса, 555, 55  Ціна: 55000$ |
|  |  |  |  |  |
|  |  |  |  |  |

3. Функціональні вимоги до програми

Програма повинна реалізувати такі дії:

-введення даних у програму

-перевірку заданих даних

-вивід результату перевірки

**Вибір методу рішення задачі**

-

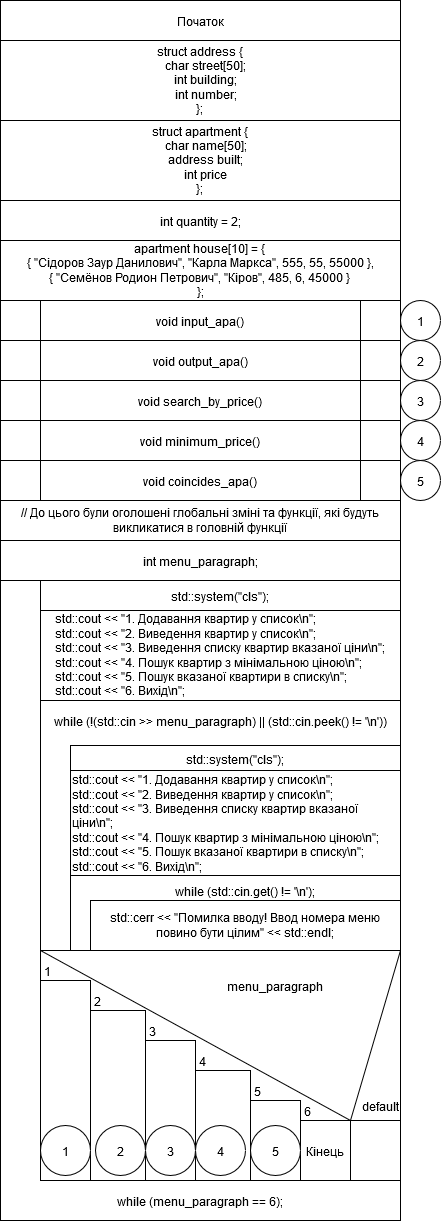
**Розробка тестів**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва | Вхідні дані | Очікувані результати |
| 1 | Номер меню | menu\_paragraf = 1(2, 3, 4, 5, 6) | Буде викликан той пунк меню номер якого він відповідяє:  1.Додавання квартир у список  (2.Виведення квартир у список  3.Виведення списку квартир вказаної ціни 4.Пошук квартир з мінімальною ціною  5.Пошук вказаної квартири в списку  6.Вихід) |
| 2 | Некоректний номер меню | menu\_paragraf = і | Помилка вводу! Ввод номера меню повино бути цілим. |
| 3 | Додавання квартири в список | menu\_paragraf = 1  ФІО: Фомин Ждан Станиславович  Вулиця: Старокозацька  Номер будинка: 49  Номер квартири: 21  Ціна:12000 | Відбудеться додавання квартири в список та поверненя на екран меню. |
| 4 | Виведення квартир у список | menu\_paragraf = 2 | №1  ФІО: Сідоров Заур Данилович  Вулиця: Карла Маркса, 555, 55  Ціна: 55000$  №2  ФІО: Семёнов Родион Петрович  Вулиця: Кірова, 485, 6  Ціна: 45000$  №3  ФІО: Фомин Ждан Станиславович  Вулиця: Старокозацька, 49, 21  Ціна: 12000$ |
| 5 | Виведення списку квартир вказаної ціни | menu\_paragraf = 3  search\_price = 12000 | Вивід даних про квартиру с такою ціною:  ФІО: Фомин Ждан Станиславович  Вулиця: Старокозацька, 49, 21  Ціна:12000$ |
| 6 | Виведення списку квартир вказаної ціни якої не має у списку | menu\_paragraf = 3  search\_price = 34000 | Не знайдено в списку кватири з такою ціною! |
| 7 | Пошук квартир з мінімальною ціною | menu\_paragraf = 4 | ФІО: Фомин Ждан Станиславович  Вулиця: Старокозацька, 49, 21  Ціна:12000$ |
| 8 | Пошук вказаної квартири в списку | menu\_paragraf = 5  ФІО: Сідоров Заур Данилович  Вулиця: Карла Маркса  Номер будинка: 555  Номер квартири: 55  Ціна:55000 | Квартира с такими реквізитами є у списку. |
| 9 | Пошук вказаної квартири в списку якої не має | menu\_paragraf = 5  ФІО: Сідоров Заур Данилович  Вулиця: Карла Маркса  Номер будинка: 562  Номер квартири: 55  Ціна:55000 | Не має квартири в списку. Можливо ви помилилися з номером будинка!  (Якщо буде помилка в іншому пункті програма вам порекомендує пункт в якому можливо є помилка ) |

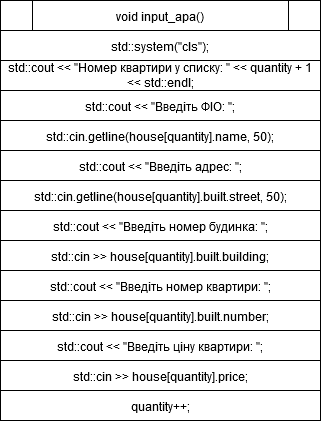
**Алгоритм програм**

1.Діаграма Н-Ш

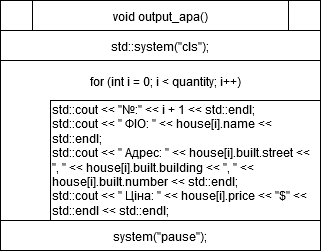
а) Головна функція ( з глобальними змінами):



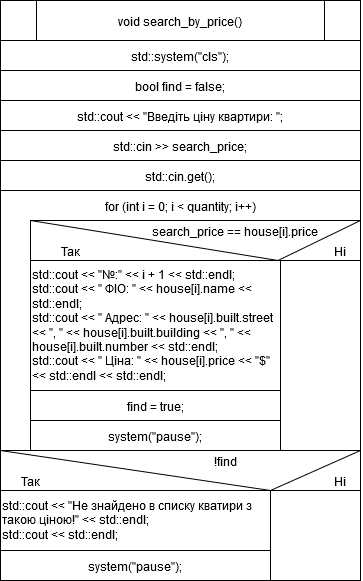
б) Функція додавання квартир в список:



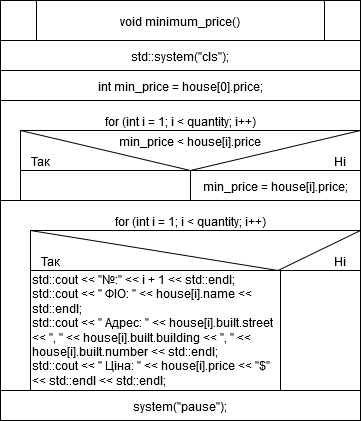
в) Функція вивід списку всіх квартир:



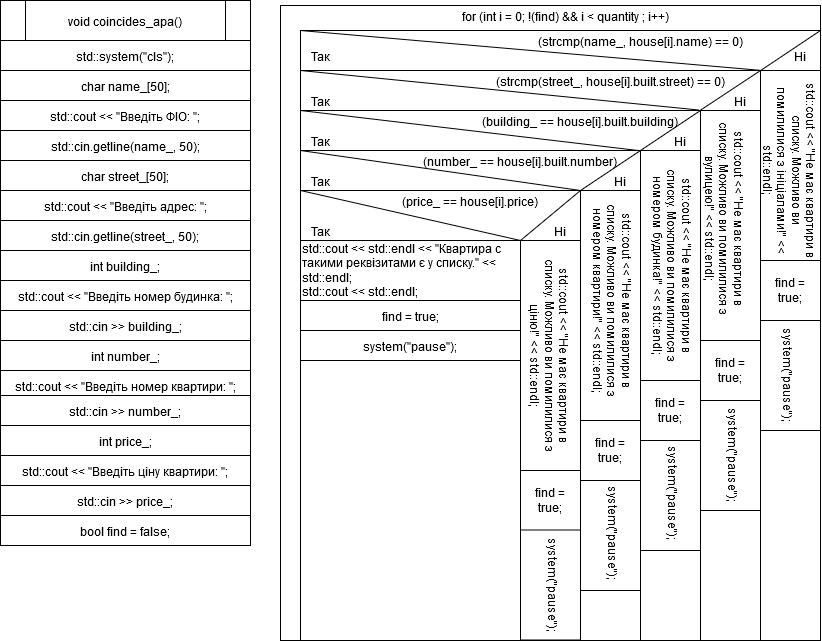
г) Функція виведення списку квартир вказаної ціни:



г) Функція пошук квартир з мінімальною ціною:



г) Функція пошук вказаної квартири в списку:



2.Текст програми

//Програма з власним меню для роботи зі структурними типами даних

/\*Програма повинна додавати реквізити квартир в список структур, виводити список руквізитів квартири на екран,

виведення списку квартир вказаної ціни, пошук квартир з мінімальною ціною, пошук вказаної квартири в списку\*/

#include<iostream>

#include<Windows.h>

#include <string>

#include <cstdlib>

struct address {

char street[50];

int building;

int number;

};

struct apartment {

char name[50];

address built;

int price

};

int quantity = 2; // кількість введених квартир в структуру

apartment house[10] = { { "Сідоров Заур Данилович", "Карла Маркса", 555, 55, 55000 },

{ "Семёнов Родион Петрович", "Кіров", 485, 6, 45000 }

};// количество уже введеных номеров

void input\_apa() // додавання квартир в список

{

std::system("cls");

std::cout << "Номер квартири у списку: " << quantity + 1 << std::endl;

std::cout << "Введіть ФІО: ";

std::cin.getline(house[quantity].name, 50);

std::cout << "Введіть адрес: ";

std::cin.getline(house[quantity].built.street, 50);

std::cout << "Введіть номер будинка: ";

std::cin >> house[quantity].built.building;

std::cout << "Введіть номер квартири: ";

std::cin >> house[quantity].built.number;

std::cout << "Введіть ціну квартири: ";

std::cin >> house[quantity].price;

quantity++;

}

void output\_apa() // вивід списку всіх квартир

{

std::system("cls");

for (int i = 0; i < quantity; i++)

{

std::cout << "№:" << i + 1 << std::endl;

std::cout << " ФІО: " << house[i].name << std::endl;

std::cout << " Адрес: " << house[i].built.street << ", " << house[i].built.building << ", " << house[i].built.number << std::endl;

std::cout << " Ціна: " << house[i].price << "$" << std::endl << std::endl;

}

system("pause");

}

void search\_by\_price() // Виведення списку квартир вказаної ціни

{

std::system("cls");

int search\_price;

bool find = false;

std::cout << "Введіть ціну квартири: ";

std::cin >> search\_price;

std::cin.get(); // очистка буфера

for (int i = 0; i < quantity; i++)

{

if (search\_price == house[i].price)

{

std::cout << "№:" << i + 1 << std::endl;

std::cout << " ФІО: " << house[i].name << std::endl;

std::cout << " Адрес: " << house[i].built.street << ", " << house[i].built.building << ", " << house[i].built.number << std::endl;

std::cout << " Ціна: " << house[i].price << "$" << std::endl << std::endl;

find = true;

system("pause");

}

}

if (!find)

{

std::cout << "Не знайдено в списку кватири з такою ціною!" << std::endl;

std::cout << std::endl;

system("pause");

}

}

void minimum\_price() // Пошук квартир з мінімальною ціною

{

std::system("cls");

int min\_price = house[0].price;

for (int i = 1; i < quantity; i++)

{

if (min\_price < house[i].price)

{

//std::cout << "Найменьша ціна " << min\_price << std::endl;

}

else

{

min\_price = house[i].price;

}

}

for (int i = 0; i < quantity; i++)

{

if (min\_price == house[i].price)

{

std::cout << "№:" << i + 1 << std::endl;

std::cout << " ФІО: " << house[i].name << std::endl;

std::cout << " Адрес: " << house[i].built.street << ", " << house[i].built.building << ", " << house[i].built.number << std::endl;

std::cout << " Ціна: " << house[i].price << "$" << std::endl << std::endl;

}

}

system("pause");

}

void coincides\_apa() // Пошук вказаної квартири в списку

{

std::system("cls");

char name\_[50];

std::cout << "Введіть ФІО: ";

std::cin.getline(name\_, 50);

char street\_[50];

std::cout << "Введіть адрес: ";

std::cin.getline(street\_, 50);

int building\_;

std::cout << "Введіть номер будинка: ";

std::cin >> building\_;

int number\_;

std::cout << "Введіть номер квартири: ";

std::cin >> number\_;

int price\_;

std::cout << "Введіть ціну квартири: ";

std::cin >> price\_;

bool find = false;

for (int i = 0; !(find) && i < quantity ; i++)

{

if (strcmp(name\_, house[i].name) == 0)

{

if (strcmp(street\_, house[i].built.street) == 0)

{

if (building\_ == house[i].built.building)

{

if (number\_ == house[i].built.number)

{

if (price\_ == house[i].price)

{

std::cout << std::endl << "Квартира с такими реквізитами є у списку." << std::endl;

std::cout << std::endl;

find = true;

system("pause");

}

else

{

std::cout << "Не має квартири в списку. Можливо ви помилилися з ціню!" << std::endl;

find = true;

system("pause");

}

}

else

{

std::cout << "Не має квартири в списку. Можливо ви помилилися з номером квартири!" << std::endl;

find = true;

system("pause");

}

}

else

{

std::cout << "Не має квартири в списку. Можливо ви помилилися з номером будинка!" << std::endl;

find = true;

system("pause");

}

}

else

{

std::cout << "Не має квартири в списку. Можливо ви помилилися з вулицею!" << std::endl;

find = true;

system("pause");

}

}

else

{

std::cout << "Не має квартири в списку. Можливо ви помилилися з ініціалами!" << std::endl;

find = true;

system("pause");

}

}

}

void main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int menu\_paragraph;

do

{

std::system("cls");

std::cout << "1. Додавання квартир у список\n";//E ч.1 +

std::cout << "2. Виведення квартир у список\n";//E ч.2 +

std::cout << "3. Виведення списку квартир вказаної ціни\n";//D +

std::cout << "4. Пошук квартир з мінімальною ціною\n";//C +

std::cout << "5. Пошук вказаної квартири в списку\n";//B +

std::cout << "6. Вихід\n";

while (!(std::cin >> menu\_paragraph) || (std::cin.peek() != '\n')) // цикл для перевірки введеної зміної на символи

{

std::system("cls");

std::cout << "1. Додавання квартир у список\n";

std::cout << "2. Виведення квартир у список\n";

std::cout << "3. Виведення списку квартир вказаної ціни\n";

std::cout << "4. Пошук квартир з мінімальною ціною\n";

std::cout << "5. Пошук вказаної квартири в списку\n";

std::cout << "6. Вихід\n";

std::cin.clear();

while (std::cin.get() != '\n');

std::cerr << "Помилка вводу! Ввод номера меню повино бути цілим" << std::endl;

std::cout << std::endl;

}

std::cin.get();//очистка буфера

switch (menu\_paragraph)

{

case 1:

input\_apa();

break;

case 2:

output\_apa();

break;

case 3:

search\_by\_price();

break;

case 4:

minimum\_price();

break;

case 5:

coincides\_apa();

break;

case 6:

return;

break;

default:

break;

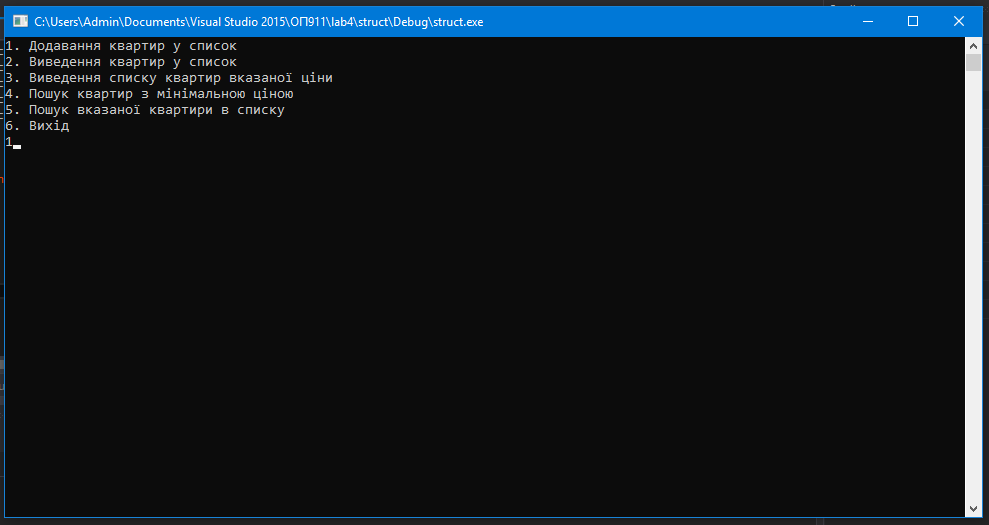
}

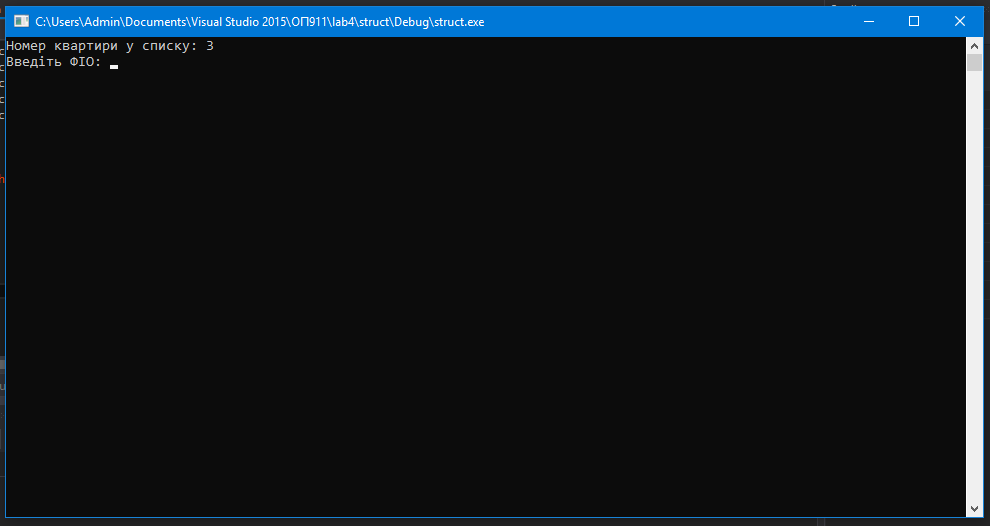
} while (menu\_paragraph == 6);

}

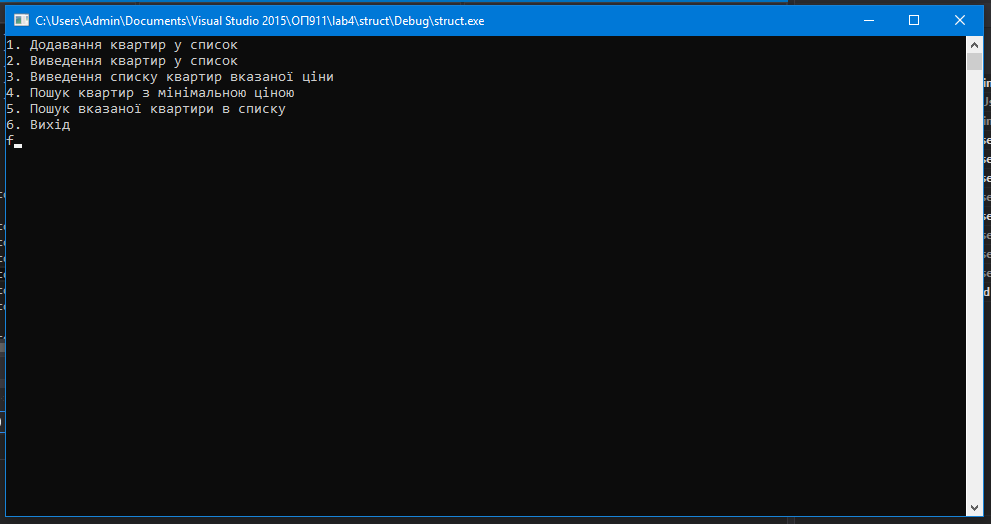
4.Результати виповнення програми

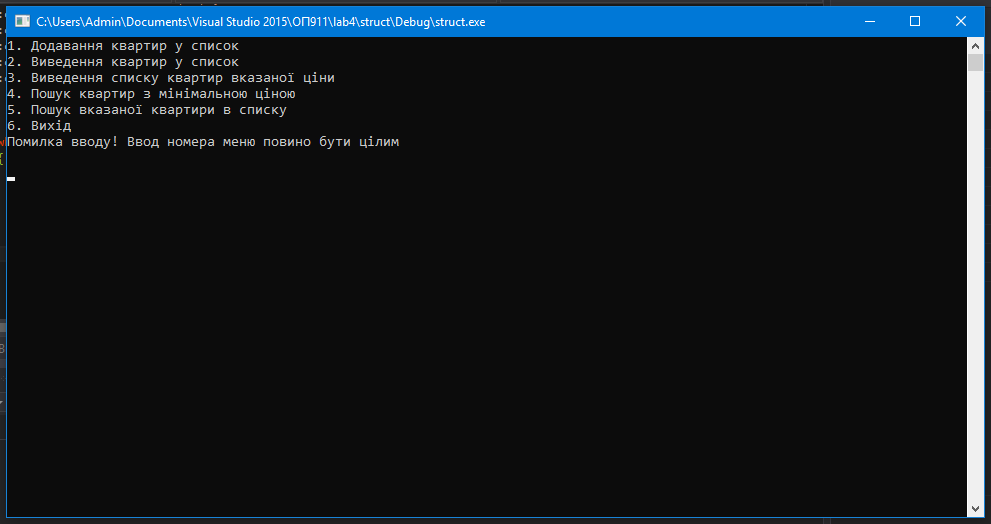
Перший тест:



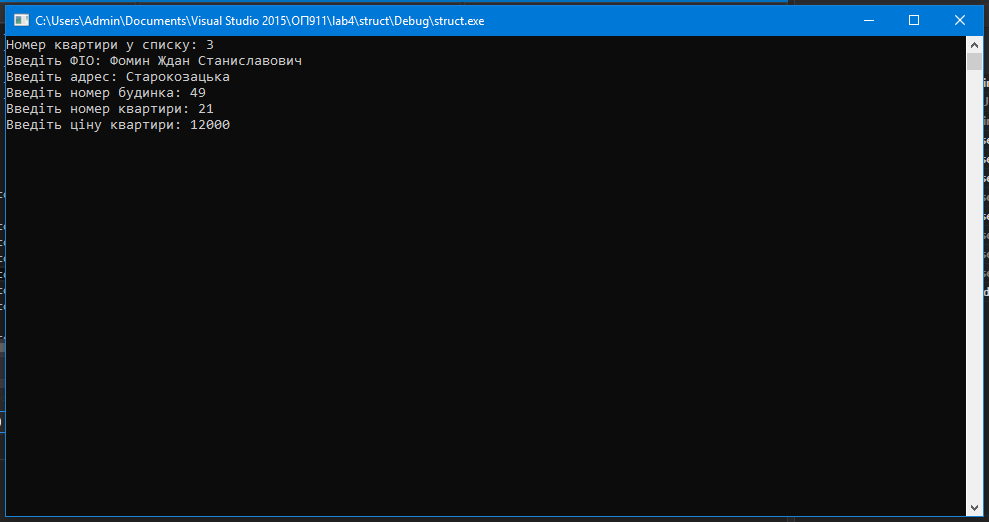


Другий тест:

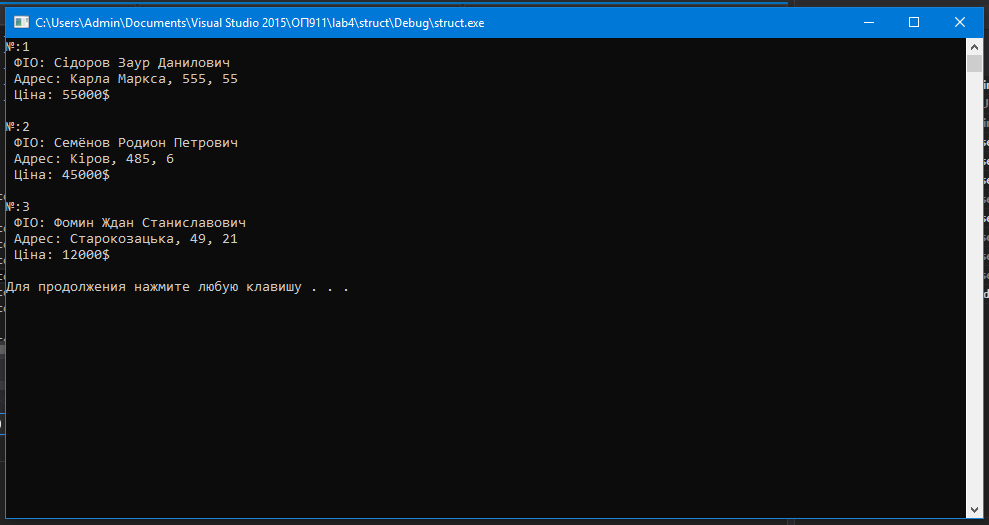




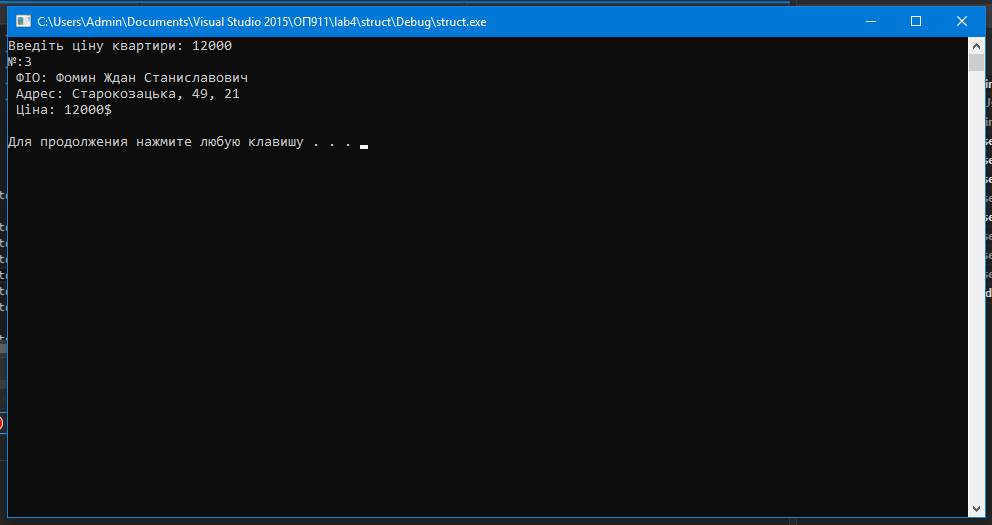
Третій тест:



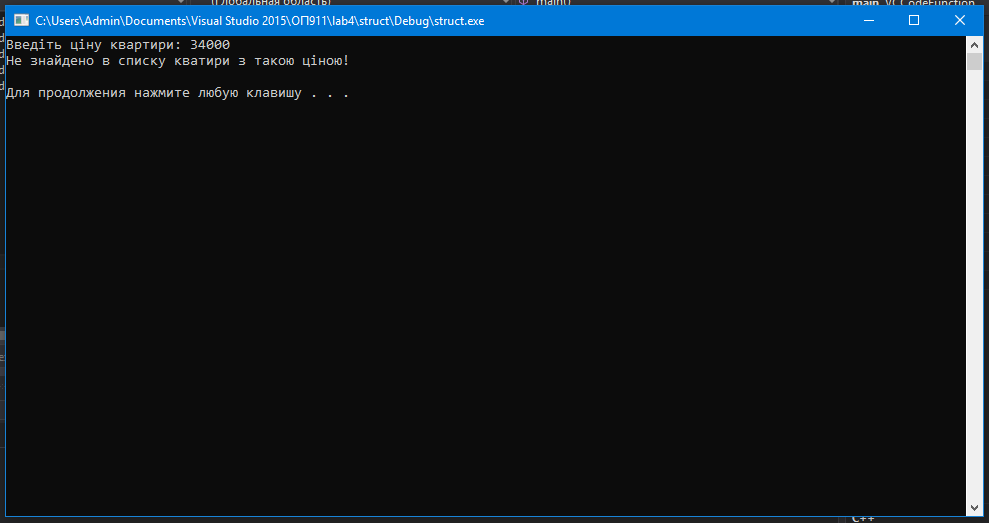
Четвертий тест



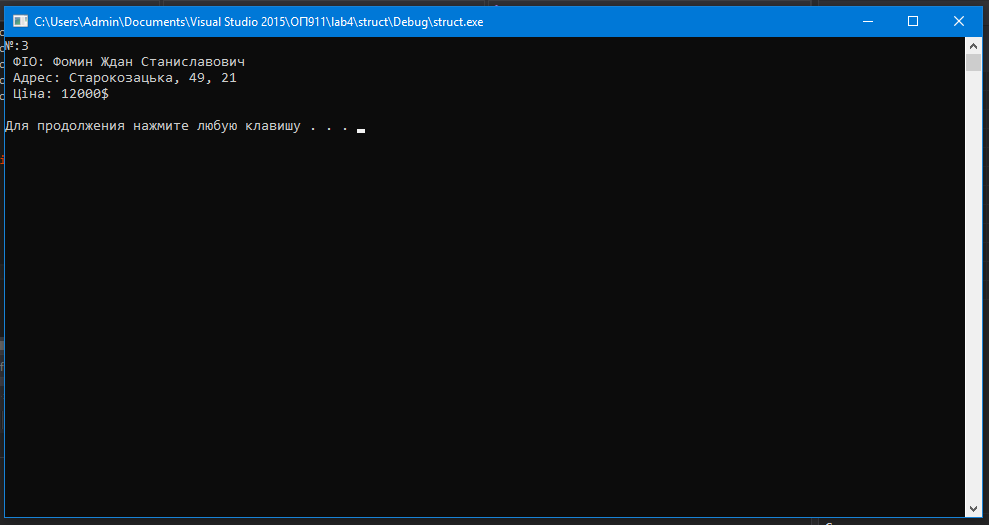
П’ятий тест



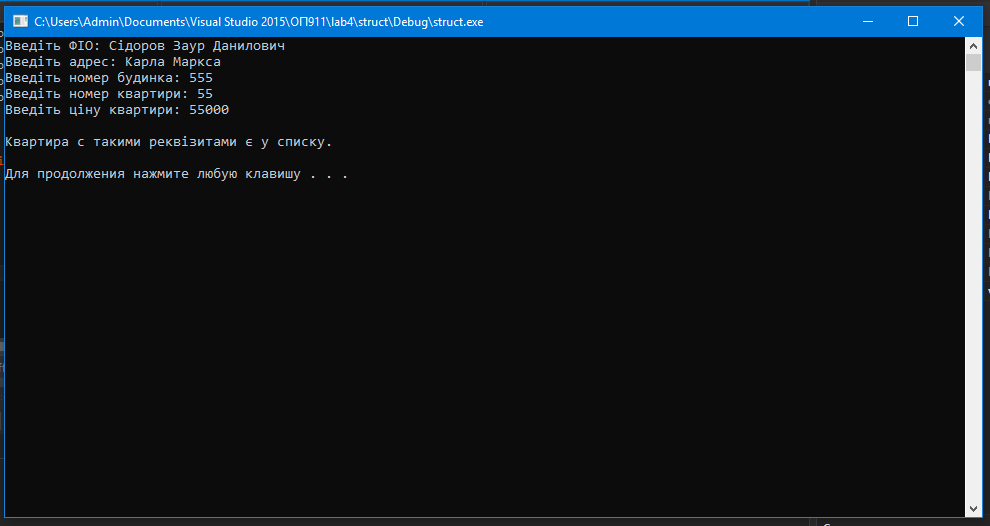
Шостий тест



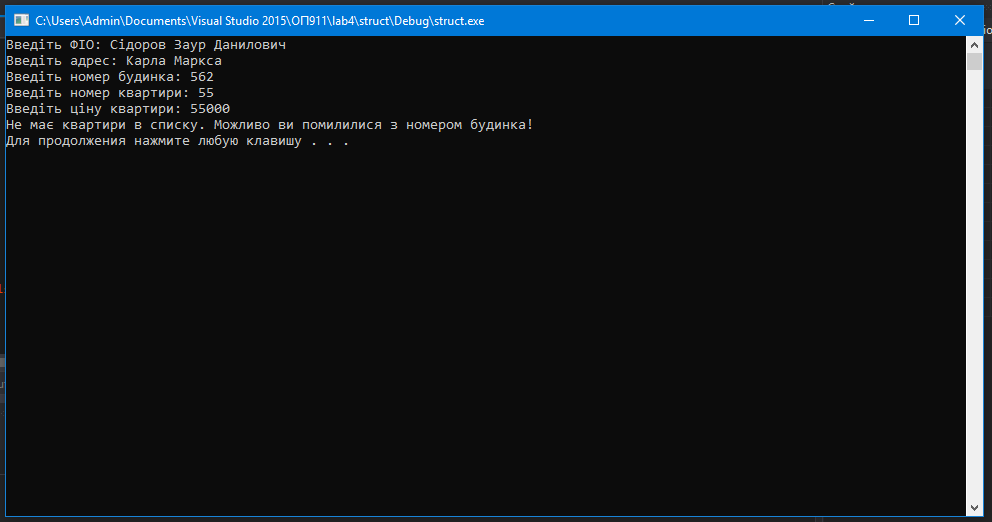
Сьомий тест



Восьмій тест



Дев’ятий тест



Висновок: C ++ дозволяє програмістам створювати свої власні користувацькі типи даних - типи, які групують кілька окремих змінних разом. Одним з найпростіших користувацьких типів даних є структура. Структура дозволяє згрупувати змінні різних типів в єдине ціле. В даній лабораторній роботі ми створювали структурний масив, за допомогою якого ми могли додавати нові квартири ( або якусь іншу інформацію ). Я також в цій лабораторній роботі робив кожний пункт case в іншій функції для більш зручної роботи з кодом.