 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №15**

з дисципліни «Алгоритмізація та програмування»

на тему: «Структури»

Варіант №2

**Виконав:**

студент гр. БС-71

Батуркіна А.М.

**Перевірив:**

ас. каф. БМК

Рисін С. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2018

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

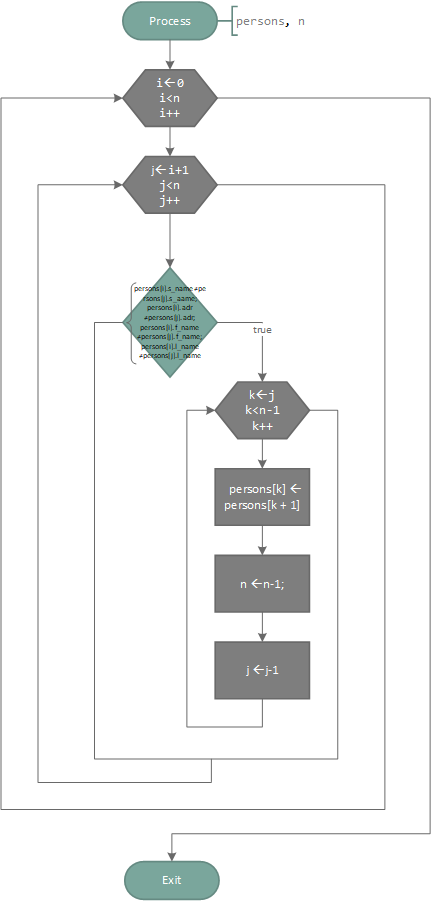
**Завдання:**

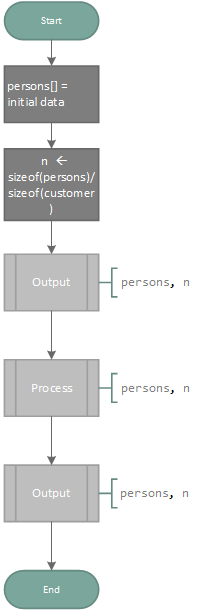
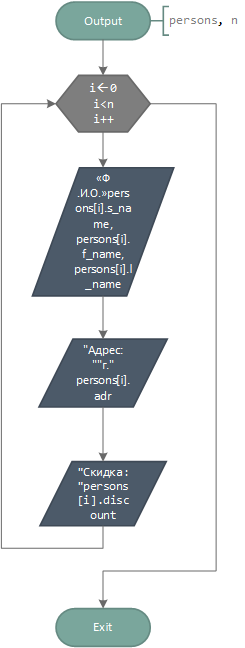
1. Ознайомитися з теоретичними відомостями щодо використання структур в мові С++ та способів доступу до їх елементів.
2. Побудувати блок-схему алгоритму для вирішення задачі відповідно до свого варіанту.
3. Створити статичний масив структур, що містить інформацію відповідного до свого варіанту завдання та розробити програмний застосунок, що включає його ініціалізацію, обробку та виведення на екран результатів (поля структури, що є рядками, задавати статичними масивами символів, ініціалізацію масиву структур провести в коді програми, глобальні змінні не використовувати, функція main має бути призначена тільки для виклику функцій користувача):

*У магазині сформовано список постійних клієнтів, який включає прізвище, ім’я, по батькові, домашню адресу покупця та розмір наданої знижки. Видалити з цього списку всі повторювані записи, перевіривши прізвище, ім’я, по батькові та домашню адресу покупця.*

1. Скласти та захистити звіт по роботі.

**Блок-схеми програми:**

****

****

**Лістинг:**

#include <iostream>

#include <stdio.h>

#include <conio.h>

#include <string.h>

using namespace std;

struct customer

{

const char \*s\_name;

const char \*f\_name;

const char \*l\_name;

const char \*adr;

int discount;

};

void Output(customer \*persons, int n)

{

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

{

cout << "Ф.И.О.: " << persons[i].s\_name << " " << persons[i].f\_name << " " << persons[i].l\_name << endl;

cout << "Адрес: " << "г." << persons[i].adr << endl;

cout << "Скидка: " << persons[i].discount << "%" << endl;

}

}

void Process(customer \*persons, int &n)

{

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

{

for (int j = i + 1; j < n; j++)

{

if (i != j && (strcmp(persons[i].s\_name, persons[j].s\_name) == 0) && (strcmp(persons[i].adr, persons[j].adr) == 0) &&

(strcmp(persons[i].f\_name, persons[j].f\_name) == 0) && (strcmp(persons[i].l\_name, persons[j].l\_name) == 0))

{

for (int k = j; k < n - 1; k++)

persons[k] = persons[k + 1];

n--; j--;

}

}

}

}

int main()

{

setlocale(LC\_ALL, "rus");

customer persons[] = { { "Рязанов", "Владимир", "Анатольевич", "Донецк", 15 },

{ "Ульянов", "Виталий", "Сергеевич", "Донецк", 8 },

{ "Белоус", "Максим", "Леонидович", "Киев", 3 },

{ "Сомко", "Людмила", "Николаевна", "Житомир", 16 },

{ "Сомко", "Людмила", "Николаевна", "Житомир", 17 } };

int n = sizeof(persons) / sizeof(customer);

Output(persons, n);

Process(persons, n);

cout << endl << "After : " << endl << endl;

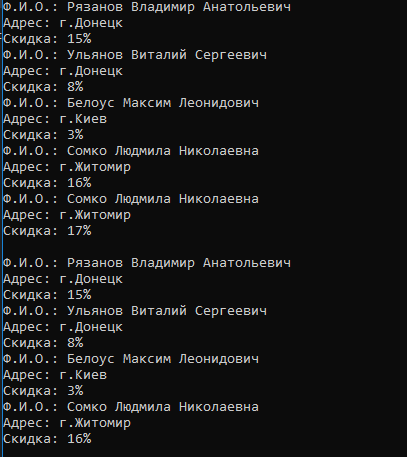
Output(persons, n);

\_getch();

return 0;

}

**Результати:**

****

**Контрольні запитання:**

1. Що таке структура, для чого вона призначена?

* Структура – це тип даних користувача, в якому під одним ім’ям об’єднанні дані різних типів. Їх використання: полегшує використання і розуміння програм; допомагає згрупувати дані, що об’єднують якесь загальне поняття; дозволяє групу пов’язаних між собою змінних використовувати як безліч окремих елементів і водночас як єдине ціле.

1. Що таке вкладена структура?

* Це структура в структурі.

1. Способи оголошення нових структурних змінних.

* Структурні змінні можуть оголошуватися
* Одразу після оголошення самої структури після фігурних дужок.
* Можуть визначатися окремо в коді програми, після оголошенняя назви структури у дужках“{}”
* Можуть бути введенні з клавіатури

1. Як ініціалізувати поле структури, яке є масивом?
2. Як оголосити покажчик на структуру?

- з використанням операції розіменування «\*», тобто

gets ((\*p).name); (\*p).data = 5;

* з використанням покажчика \*->\*,

наприклад, gets (p -> name); p-> data = 5; тощо.

Крім того, до полів змінної sti можна звертатися, вказуючи поля через операцію «.», як це робилося раніше.

1. Як ініціалізувати покажчик на структуру?

* Покажчики на структури нічим не відрізняються від покажчиків на звичайні змінні. Об’явлення struct student \*ptr;

повідомляє, що ptr - це покажчик на структуру типу student. Якщо ptr вказує на структуру student, то \* ptr - це сама структура, а, наприклад, (\*ptr).a і (\*ptr).b - її елементи (поля). Використовуючи покажчик ptr, ми могли б написати

struct student data, \*ptr;

ptr = &data; - ініціалізація покажчика на структуру

1. Які операції икористовуються для доступу до елементів структури, в чому їх різниця?

* Для доступу до елементів структури використовуються операції доступу до елементів - операція крапка (.) І операція стрілка (->).

Операція крапка звертається до елементу структури по імені змінної об'єкта. Операція стрілка, що складається зі знака мінус (-) і знака більше (>), записаних без пробілу, забезпечує доступ до елементу структури через покажчик на об'єкт.

1. Яким чином передаються структури до функцій?

- Структури до функцій можна передати використовуючи покажчик, посилання, або за значенням.

Структури можуть передаватися функціям шляхом передачі окремих елементів структури, передачі всієї структури або передачі вказівника на структуру. Коли структури або окремі їх елементи передаються функції, проходить передача по значенню. А тому функція, що викликає не може змінювати елементи у структурі що викликається. Щоб здійснити передачу структури по посиланню, необхідно передати їй адресу.

1. Чи можна передавати в функцію масив структур?

* Так, у функцію можна передавати масив структур, оскільки він не відрізняється від звичайних масивів. Зокрема, масиви структур можна передавати через покажчик або по посиланню.

1. Які проблеми можуть виникнути під час копіювання змінних структурного типу одна в одну?

* Дані типу простих структур можна присвоювати один одному тільки в тому випадку, якщо всі члени обох структур мають числові типи. При цьому в операції присвоювання обидва операнда, і зліва і справа, повинні бути структурного типу. Тобто проводиться просте копіювання. Якщо розміри структур відрізняються, то копіюється кількість байт, що відповідає меншому розміру.