**ЛАБОРАТОРНАЯ РАБОТА №7**

**«ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ НАД СТРУКТУРАМИ И НАД МАССИВАМИ СТРУКТУР»**

**Цель работы**: Изучение способов описания структур данных на языке С. **1. Вариант задания**

Описать структуру с именем clerk, содержащую следующие поля: порядковый

номер, фамилия, имя, должность, стаж, подразделение, оклад. Данные сортировать

по полю «оклад». Выполнить поиск индекса служащего с наибольшим окладом из

подразделения, введенного пользователем.

**2. Алгоритм программы**

Программа состоит из нескольких функций которые вызываются в основной функции main. Для реализации данной программы были сделаны несколько отдельных функций. Схемы всех функций будут приведены ниже.



Рисунок 1 – Основное тело программы



Рисунок 2 – Ввод записей с клавиатуры(слева) и вывод записей на экран (справа)



Рисунок 3 – Алгоритм сортировки по полю «оклад»



Рисунок 4 – Алгоритм поиска работника с максимальным окладом в определенной отделе.

**3. Код программы на языке C**

#pragma warning(disable : 4996) ;

#include <stdio.h>

#include <string.h>

#include <conio.h>

#include <windows.h>

#include <stdlib.h>

#define DESCENDING\_SORT (0)

#define ASCENDING\_SORT (1)

typedef

struct clerk

{

int

num, exp;

char

name[32],

surname[32],

position[32],

division[32];

double

salary;

}

clerk\_t;

/\* Печать информации \*/

void print\_clerkOne(const clerk\_t\* e);

/\* Печать информации о всех\*/

void print\_clerk(const clerk\_t\* e, int arr\_len);

/\* Ввод информации \*/

clerk\_t\* scan\_clerk(int arr\_len);

/\* Сортировка по полю \*/

void sort\_array(clerk\_t\* arr, int arr\_len, int sort\_order\_flag);

/\*

Поиск Клерка с максимальным окладом.

Вернуть индекс этого Клерка.

\*/

static int search\_element(clerk\_t arr[], int arr\_size, const char\* division);

main()

{

clerk\_t\* mas;

SetConsoleCP(1251); // Задаем таблицу символов для консоли.

SetConsoleOutputCP(1251);

int selector = 1; int n = 0; mas = 0; // селектор - выбор пункта, мас -сам массив, его над задавать как 0

while (selector != 0) //цикл меню , выходит из него как ток будет 0

{

system("cls");

printf("\n\nВыберите необходимый пункт \n");

printf("1 - Ввод клерков с клавиатуры \n");

if (n > 0) {

printf("2 - Вывод клерков на экран \n");

printf("3 - Сортировка по полю \n");

printf("4 - Поиск клерка с максимальным окладом \n");

}

printf("\n0 - Выход\n\n");

scanf("%d", &selector); //считали селектор

switch (selector) //вызов функции в зависимости от селектора

{

case 1: {printf("\n Введите размер массива --> "); scanf("%d", &n); mas = scan\_clerk(n); break; }; //ввод

case 2: {print\_clerk(mas, n); break; }; //вывод

case 3: {int flag; printf("Введите флаг сортировки, 0 - по убыванию, 1 - по возрастанию -->"); scanf("%d", &flag); sort\_array(mas, n, flag); break; }; //положительные в квадрат

case 4: { char divis[32]; printf("Введите подразделение в котором будет выполнен поиск --> "); scanf("%s", divis);

int max\_ock = search\_element(mas, n, divis);

if (max\_ock != -1) { printf("\nКлерк с максимальным окладом(номер) --> %d \n", max\_ock);

print\_clerkOne(&mas[max\_ock]);

}

else printf("\nНе правильно введена должность/нет подходящего клерка ");

printf("\a");

getch();

break; }; // произведение на интервале

default:

break;

}

}

free(mas);

return 0;

}

/\* Печать информации \*/

void print\_clerkOne(const clerk\_t\* e)

{

printf("Клерк:\r\n");

printf(" \* Порядковый номер: %02d.\r\n", e->num);

printf(" \* Фамилия: <%s>.\r\n", e->surname);

printf(" \* Имя: <%s>.\r\n", e->name);

printf(" \* Должность: <%s>.\r\n", e->position);

printf(" \* Стаж: %d.\r\n", e->exp);

printf(" \* Подразделение: <%s>.\r\n", e->division);

printf(" \* Оклад: %.3f руб.\r\n", e->salary);

}

/\* Печать информации о всех \*/

void print\_clerk(const clerk\_t\* e, int arr\_len)

{

for (int i = 0; i < arr\_len; i++) { //цикл вывода

print\_clerkOne(&e[i]);

}

printf("\a"); // звук окончания вывода

getch(); //ждем ввод чего либо

}

/\* Ввод информации \*/

clerk\_t\* scan\_clerk(int arr\_len)

{

clerk\_t\* e = (clerk\_t\*)malloc(sizeof(clerk\_t) \* arr\_len); //выделение памяти

for (int i = 0; i < arr\_len; i++) { //цикл ввода

printf("Добавление информации о Клерк:\r\n");

printf(" \* Введите Фамилию: "); scanf("%s", &e[i].surname);

printf(" \* Введите Имя: "); scanf("%s", e[i].name);

printf(" \* Введите Должность: "); scanf("%s", e[i].position);

printf(" \* Введите Стаж(лет): "); scanf("%d",&e[i].exp);

printf(" \* Введите Подразделение: "); scanf("%s", e[i].division);

printf(" \* Введите Оклад: "); scanf("%lf", &e[i].salary);

e[i].num = i; //личный идинтификатор = номеру при создании

}

return e; //возвращаем указатель

}

/\* Сортировка по полю \*/

void sort\_array(clerk\_t\* arr, int arr\_len, int sort\_order\_flag)

{

clerk\_t tmp = {0};

for (int i = arr\_len-1; i > 0; i--) // сортировка методом пузырька

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

if (

(sort\_order\_flag) ?

(((arr[j]).salary) > ((arr[j+1]).salary)) : // Если сортировка по возрастанию

((arr[j]).salary) < ((arr[j+1]).salary)) // Если сортировка по убыванию

{

tmp = arr[j]; //временная переменная

arr[j] = arr[j+1];// замена местами

arr[j+1] = tmp;

}

// Расстановка правильных порядковых номеров

for (int i = 0; i < arr\_len; i++)

arr[i].num = i;

}

/\*

Поиск Клерка с максимальным окладом.

Вернуть индекс этого Клерка.

\*/

static int search\_element(clerk\_t arr[], int arr\_size, const char\* division)

{

int max\_salary\_num = -1; //задаем начальный флаг как -1;

for (int i = 0; i < arr\_size; i++) //в цикле просматриваем весь массив записей

if (!strcmp(arr[i].division, division) && //если совпадает должность

(

(max\_salary\_num != -1) ? //проверяем , если флаг не равен -1, то происходит сравнение максимального заработка с текущим заработком

(arr[i].salary > arr[max\_salary\_num].salary) :

(1) //иначе выдадим 1, да бы запомнить индекс элемента с максимальным окладом

)

)

max\_salary\_num = i; //запоминаем индекс с максимальным окладом

return max\_salary\_num; // возвращаем максимальный оклад

}

**4. Код программы на языке C++**

#pragma warning(disable : 4996) ;

#include <stdio.h>

#include <string.h>

#include <conio.h>

#include <windows.h>

#include <stdlib.h>

#define DESCENDING\_SORT (0)

#define ASCENDING\_SORT (1)

typedef

struct clerk

{

int

num, exp;

char

name[32],

surname[32],

position[32],

division[32];

double

salary;

}

clerk\_t;

/\* Печать информации \*/

void print\_clerkOne(const clerk\_t\* e);

/\* Печать информации о всех\*/

void print\_clerk(const clerk\_t\* e, int arr\_len);

/\* Ввод информации \*/

clerk\_t\* scan\_clerk(int arr\_len);

/\* Сортировка по полю \*/

void sort\_array(clerk\_t\* arr, int arr\_len, int sort\_order\_flag);

/\*

Поиск Клерка с максимальным окладом.

Вернуть индекс этого Клерка.

\*/

static int search\_element(clerk\_t arr[], int arr\_size, const char\* division);

main()

{

clerk\_t\* mas;

SetConsoleCP(1251); // Задаем таблицу символов для консоли.

SetConsoleOutputCP(1251);

int selector = 1; int n = 0; mas = 0; // селектор - выбор пункта, мас -сам массив, его над задавать как 0

while (selector != 0) //цикл меню , выходит из него как ток будет 0

{

system("cls");

printf("\n\nВыберите необходимый пункт \n");

printf("1 - Ввод клерков с клавиатуры \n");

if (n > 0) {

printf("2 - Вывод клерков на экран \n");

printf("3 - Сортировка по полю \n");

printf("4 - Поиск клерка с максимальным окладом \n");

}

printf("\n0 - Выход\n\n");

scanf("%d", &selector); //считали селектор

switch (selector) //вызов функции в зависимости от селектора

{

case 1: {printf("\n Введите размер массива --> "); scanf("%d", &n); mas = scan\_clerk(n); break; }; //ввод

case 2: {print\_clerk(mas, n); break; }; //вывод

case 3: {int flag; printf("Введите флаг сортировки, 0 - по убыванию, 1 - по возрастанию -->"); scanf("%d", &flag); sort\_array(mas, n, flag); break; }; //положительные в квадрат

case 4: { char divis[32]; printf("Введите подразделение в котором будет выполнен поиск --> "); scanf("%s", divis);

int max\_ock = search\_element(mas, n, divis);

if (max\_ock != -1) { printf("\nКлерк с максимальным окладом(номер) --> %d \n", max\_ock);

print\_clerkOne(&mas[max\_ock]);

}

else printf("\nНе правильно введена должность/нет подходящего клерка ");

printf("\a");

getch();

break; }; // произведение на интервале

default:

break;

}

}

free(mas);

return 0;

}

/\* Печать информации \*/

void print\_clerkOne(const clerk\_t\* e)

{

printf("Клерк:\r\n");

printf(" \* Порядковый номер: %02d.\r\n", e->num);

printf(" \* Фамилия: <%s>.\r\n", e->surname);

printf(" \* Имя: <%s>.\r\n", e->name);

printf(" \* Должность: <%s>.\r\n", e->position);

printf(" \* Стаж: %d.\r\n", e->exp);

printf(" \* Подразделение: <%s>.\r\n", e->division);

printf(" \* Оклад: %.3f руб.\r\n", e->salary);

}

/\* Печать информации о всех \*/

void print\_clerk(const clerk\_t\* e, int arr\_len)

{

for (int i = 0; i < arr\_len; i++) { //цикл вывода

print\_clerkOne(&e[i]);

}

printf("\a"); // звук окончания вывода

getch(); //ждем ввод чего либо

}

/\* Ввод информации \*/

clerk\_t\* scan\_clerk(int arr\_len)

{

clerk\_t\* e = (clerk\_t\*)malloc(sizeof(clerk\_t) \* arr\_len); //выделение памяти

for (int i = 0; i < arr\_len; i++) { //цикл ввода

printf("Добавление информации о Клерк:\r\n");

printf(" \* Введите Фамилию: "); scanf("%s", &e[i].surname);

printf(" \* Введите Имя: "); scanf("%s", e[i].name);

printf(" \* Введите Должность: "); scanf("%s", e[i].position);

printf(" \* Введите Стаж(лет): "); scanf("%d",&e[i].exp);

printf(" \* Введите Подразделение: "); scanf("%s", e[i].division);

printf(" \* Введите Оклад: "); scanf("%lf", &e[i].salary);

e[i].num = i; //личный идинтификатор = номеру при создании

}

return e; //возвращаем указатель

}

/\* Сортировка по полю \*/

void sort\_array(clerk\_t\* arr, int arr\_len, int sort\_order\_flag)

{

clerk\_t tmp = {0};

for (int i = arr\_len-1; i > 0; i--) // сортировка методом пузырька

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

if (

(sort\_order\_flag) ?

(((arr[j]).salary) > ((arr[j+1]).salary)) : // Если сортировка по возрастанию

((arr[j]).salary) < ((arr[j+1]).salary)) // Если сортировка по убыванию

{

tmp = arr[j]; //временная переменная

arr[j] = arr[j+1];// замена местами

arr[j+1] = tmp;

}

// Расстановка правильных порядковых номеров

for (int i = 0; i < arr\_len; i++)

arr[i].num = i;

}

/\*

Поиск Клерка с максимальным окладом.

Вернуть индекс этого Клерка.

\*/

static int search\_element(clerk\_t arr[], int arr\_size, const char\* division)

{

int max\_salary\_num = -1; //задаем начальный флаг как -1;

for (int i = 0; i < arr\_size; i++) //в цикле просматриваем весь массив записей

if (!strcmp(arr[i].division, division) && //если совпадает должность

(

(max\_salary\_num != -1) ? //проверяем , если флаг не равен -1, то происходит сравнение максимального заработка с текущим заработком

(arr[i].salary > arr[max\_salary\_num].salary) :

(1) //иначе выдадим 1, да бы запомнить индекс элемента с максимальным окладом

)

)

max\_salary\_num = i; //запоминаем индекс с максимальным окладом

return max\_salary\_num; // возвращаем максимальный оклад

}

**5. Результаты тестирования и отладки программы**

Для тестирования программы был создан файлик в котором хранятся 7 тестовых записей для проверки работоспособности программы.

При первом запуске программы, в меню у нас будет только один пункт. Это связанно с тем что записей пока что ещё нет.

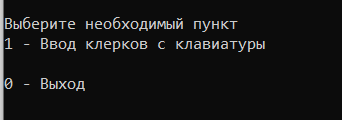


Рисунок 5 – Меню при первом запуске программы

При выборе первого пункта , программа спросит количество записей и предложит ввести каждую запись по полям.

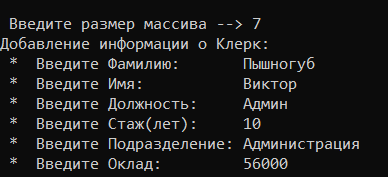


Рисунок 6 – Пример ввода записей

После того как мы введем все записи, откроется меню, но вид у него будет уже другой.

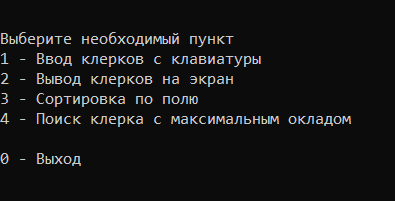


Рисунок 7 – Меню после ввода всех данных

Второй пункт выведет записи на экран.

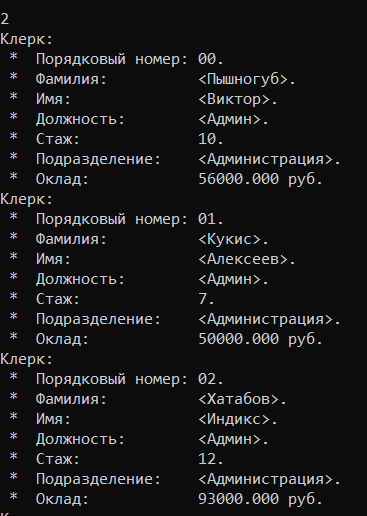


Рисунок 8 – Пример вывода записей на экран

Третий пункт позволяет отсортировать записи в массиве по окладу, причем выбрав метод сортировки (по возрастанию или по убыванию).

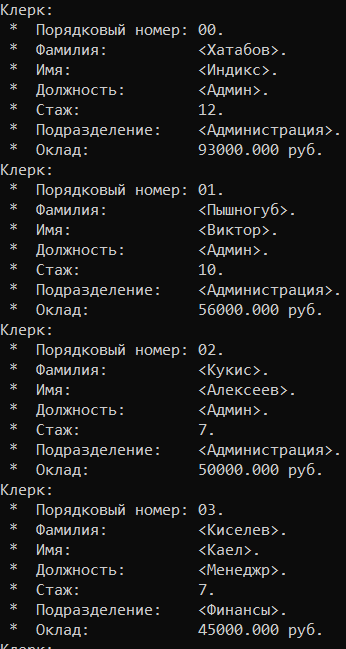
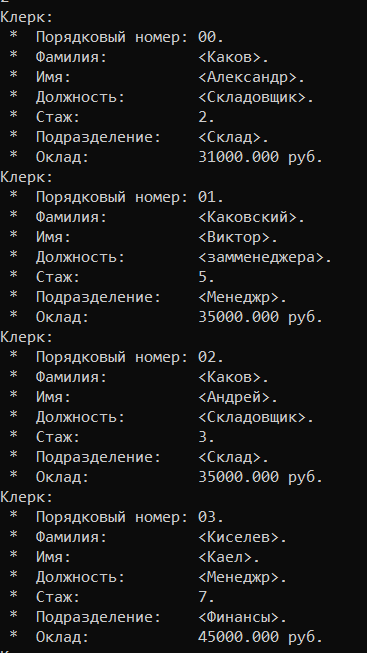


Рисунок 9 – пример сортировки по возрастанию(слева) и по убыванию ( справа)

Можно выполнить поиск по определенному полю, при этом необходимо указать подразделение, в котором будет выполнен поиск клерка с максимальным окладом.

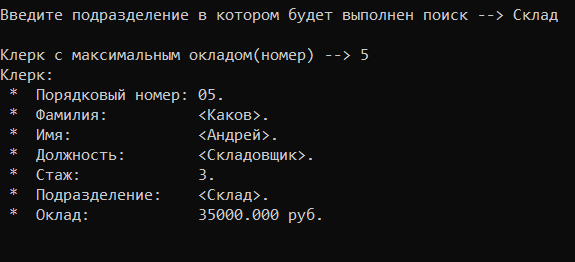
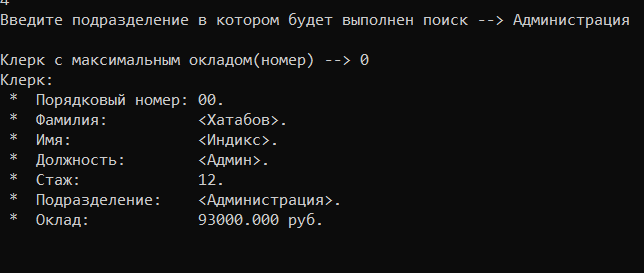


Рисунок 10 – Примеры поиска клерка с максимальным окладом в разных подразделениях

**Вывод**

В ходе выполнения данной лабораторной работы были изучены особенности представления и обработки структур и массивов структур в языках С/С++ с учетом связи указателей. Получены практические навыки реализации алгоритмов обработки динамических массивов структур средствами языков С/С++.