**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

Факультет комп’ютерних наук

Кафедра «Інтелектуальних інформаційних систем»

**ЗВІТ**

*з лабораторної роботи № 5*

**Варіант № 8**

Дисципліна "Структури та організація даних"

Напрям підготовки: 122 Комп’ютерні науки

ЛР.ПЗ.01-201.1910109

***Cтудент*** *\_\_\_\_\_\_\_\_\_\_\_\_\_Грабовськиий Є.О.*

*(підпис)*

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

*(дата)*

***Викладач***

*\_\_\_\_\_\_\_\_\_\_\_\_Кірей К.О.\_\_\_ (підпис)*

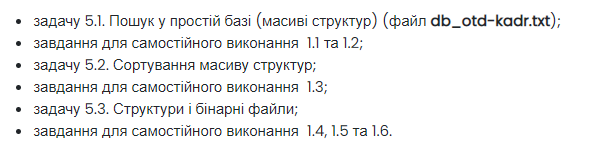
*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

Миколаїв – 2020

**ЗВІТ**

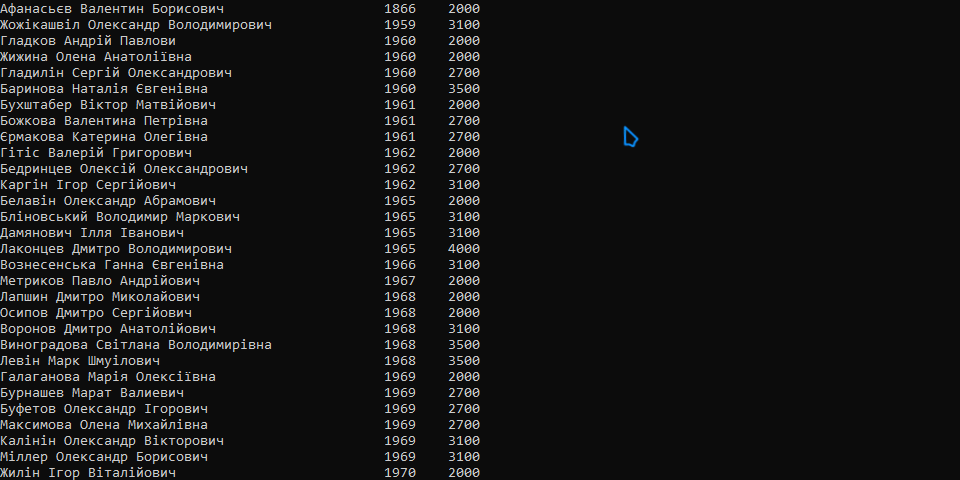
**Завдання**



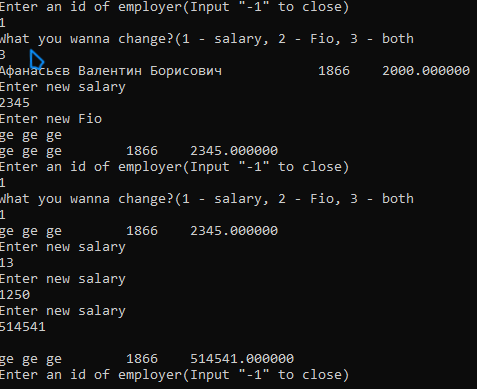
1 завдання



2 завдання



4 завдання



Опис роботи

1. Чтение с помощью функций strncpy && atoi, заполнение массива типа Employer, ввод символов, при нажатии клавиши «ESC» производится выход из программы путём считывания кода клавиши. Поиск нужного работника по имени.
2. То же самое, только вместо поиска – сортировка по двум ключам – зарплата и год рождения
3. Открытые двух потоков – чтение и запись, запись во второй файл данных первого файла с помощью функций fgets(запись в структуру) и fwrite(запись в файл).
4. Открытые потока на редактирование(чтение-запись), подсчёт записей, производится с помощью перемещения флага в конец и получение размера файла, деление этого размера на размер структуры(char[35] + int + float). Ввод номера работника и действия(какую часть данных изменить), введение новых данных, их замена совершается перемещением флага на «стартовую» позицию работника и полная перезапись.

**Лістинг коду**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <fstream>

#include <algorithm>

#include <cstdio>

#include <conio.h>

#include <windows.h>

void clear();

void firstEx();

void secondEx();

void thirdEx();

void fourthEx();

const int nameLength = 35;

const int yearLength = 5;

const int paymentLength = 10;

const int dbSize = 100;

const int lineLenght = nameLength + yearLength + paymentLength;

struct Employer

{

char name[nameLength + 1]{};

int year;

float salary;

Employer()

{

this->year = 0;

this->salary = 0;

}

void show() const

{

std::cout << name << '\t' << year << '\t' << salary << '\n';

}

};

int main()

{

firstEx();

}

void firstEx()

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

std::ifstream fin("db\_otd-kadr.txt");

if (!fin.is\_open())

{

std::cout << "The file don`t exist\n";

return;

}

Employer employee[dbSize];

char\* buffer = new char[lineLenght + 1];

int index = 0;

while (!fin.eof())

{

fin.getline(buffer, lineLenght);

//запись имени в структуру

strncpy(employee[index].name, buffer, nameLength);

employee[index].name[nameLength] = '\0';

//запись зп и года рождения в структуру, atoi переводит из char\* в int

employee[index].year = std::atoi(&buffer[nameLength]);

employee[index].salary = std::atoi(&buffer[nameLength + yearLength]);

++index;

}

std::string name;

char ch;

bool isQuit = false;

while (true)

{

std::cout << "Write the name of employer\n";

do {

ch = \_getch();

//код клавиши ESC

if ((int)ch == 27)

{

isQuit = true;

break;

}

putchar(ch);

//код упр. cимвола \n

if ((int)ch == 13) {

break;

}

name += ch;

} while (1);

std::cout << '\n';

if (isQuit)

{

std::cout << "Thank you for using that program\n";

break;

}

ch = 0;

bool isContain = true;

for (int index = 0; index < dbSize; ++index)

{

for (int index2 = 0; employee[index].name[index2] != 0; ++index2) {

if(employee[index].name[index2] == ' ')

{

break;

}

if (index2 == name.length()) {

while (employee[index].name[index2] != 0) {

if (employee[index].name[index2] != ' ' && employee[index].name[index2] != '\t') {

isContain = false;

break;

}

++index2;

}

break;

}

if (name[index2] != employee[index].name[index2])

{

isContain = false;

break;

}

}

if (isContain)

{

std::cout << "Finded\n";

employee[index].show();

break;

}

isContain = true;

}

\_getch();

name.clear();

clear();

}

fin.close();

}

void secondEx()

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

std::ifstream fin("db\_otd-kadr.txt");

if (!fin.is\_open())

{

std::cout << "The file don`t exist\n";

return;

}

Employer employee[dbSize];

char\* buffer = new char[lineLenght + 1];

int index = 0;

while (!fin.eof())

{

fin.getline(buffer, lineLenght);

//запись имени в структуру

strncpy(employee[index].name, buffer, nameLength);

employee[index].name[nameLength] = '\0';

//запись зп и года рождения в структуру, atoi переводит из char\* в int

employee[index].year = std::atoi(&buffer[nameLength]);

employee[index].salary = std::atoi(&buffer[nameLength + yearLength]);

++index;

}

Employer key;

int j;

for (int i = 1; i < dbSize; i++) {

key = employee[i];

j = i;

while (j > 0 && employee[j - 1].year >= key.year && employee[j - 1].salary > key.salary) {

employee[j] = employee[j - 1];

j--;

}

employee[j] = key;

}

for (int index = 0; index < dbSize; ++index)

{

employee[index].show();

}

fin.close();

}

void thirdEx()

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

std::FILE\* fin, \* fout;

fin = fopen("db\_otd-kadr.txt", "r");

fout = fopen("db\_otd-kadr.bin", "wb");

if (fin == NULL || fout == NULL)

{

puts("The file don`t exist\n");

return;

}

Employer employer = Employer();

while (!feof(fin))

{

//запись имени работника в структуру

fgets(employer.name, nameLength, fin);

//запись зп и года рождения в структуру

fscanf(fin, "%i%f\n", &employer.year, &employer.salary);

printf("%s\t%i\t%f\n", employer.name, employer.year, employer.salary);

//запись в файл данных о работнике

fwrite(&employer, sizeof(employer), 1, fout);

}

fclose(fin);

fclose(fout);

}

void fourthEx()

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

std::FILE\* fout;

fout = fopen("db\_otd-kadr.bin", "r + b");

if (fout == NULL)

{

std::cout << "The file don`t exist\n";

return;

}

Employer employer;

//переход в конец файла

fseek(fout, 0, SEEK\_END);

//подсчёт количества полей по известным данным

int count = ftell(fout) / sizeof(employer);

std::cout << "The count of employer is " << count << '\n';

int num = 0;

int helper = 0, choice = 0;

do

{

puts("Enter an id of employer(Input \"-1\" to close)");

std::cin >> helper;

puts("What you wanna change?(1 - salary, 2 - Fio, 3 - both");

std::cin >> choice;

//очистка буфера

fflush(stdin);

if (num == -1)break;

//переход к строчке с работником

fseek(fout, num \* sizeof(employer), SEEK\_SET);

//считывание и запись в переменную сотрудника

fread(&employer, sizeof(employer), 1, fout);

printf("%s\t%i\t%f\n", employer.name, employer.year, employer.salary);

if (choice == 1) {

int tmp = employer.salary;

do

{

puts("Enter new salary");

scanf("%f", &employer.salary);

} while (employer.salary < tmp / 2 || employer.salary <= tmp);

}

else if (choice == 2) {

puts("Enter new Fio");

std::cin >> employer.name;

}

else if (choice == 3)

{

int tmp = employer.salary;

do

{

puts("Enter new salary");

scanf("%f", &employer.salary);

} while (employer.salary < tmp / 2 || employer.salary <= tmp);

puts("Enter new Fio");

std::cin >> employer.name;

}

else

{

puts("Try again");

continue;

}

//повторный переход и перезапись

fseek(fout, num \* sizeof(employer), SEEK\_SET);

fwrite(&employer, sizeof(employer), 1, fout);

printf("%s\t%i\t%f\n", employer.name, employer.year, employer.salary);

helper = 0;

} while (helper != 1);

fclose(fout);

}

void clear() {

std::cout << "\x1B[2J\x1B[H";

}