Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Программная инженерия»

Кафедра инженерной психологии и эргономики

Учебная дисциплина «Основы Алгоритмизации и программирования»

Отчет

по лабораторной работе №7

«Структуры и файлы»

Вариант 18

Подготовил: Козинцев М.Л

Проверил: Усенко Ф.В.

Минск 2024

***Цель работы:*** сформировать навыки и умения обработки структурированных типов данных, организованных в виде структур и файлов.

***Задание:*** Создать структуру должностей Vacancies {Manager, Boss, Clerk, Salesman, etc.}. Создать структуру «Employee», состоящую из:

- поля name строкового типа;

- поля vacancy типа Vacancies;

- поля зарплата целого типа;

- поля дата приема на работу типа int[3].

Создать массив сотрудников. Длина массива задается пользователем, заполнение массива производится им же. Вывести полную информацию обо всех сотрудниках. Предусмотреть запись в файл

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

enum Vacancies { Manager, Boss, Clerk, Salesman, Other, Undefined };

struct Employee {

string name;

Vacancies vacancy;

int salary;

int hireDate[3];

};

void inputEmployee(Employee& employee);

void outputEmployee(Employee& employee);

void writeEmployeeToFile(Employee& employee, ofstream& file);

int main() {

int n, i;

cout << "Enter amount of employees: ";

cin >> n; //Ввод количества сотрудников

Employee\* arr = new Employee[n];

for (i = 0; i < n; i++) { //Заполнение массива структур

cout << "Enter " << i + 1 << " employee data:\n";

inputEmployee(arr[i]);

cout << '\n';

}

for (i = 0; i < n; i++) { //Вывод массива структур

cout << "\nEmployee " << i + 1 << " data:\n";

outputEmployee(arr[i]);

cout << '\n';

}

ofstream file("employees.txt");

//Запись структур в файл

if (file.is\_open()) {

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

file << "\nEmployee " << (i + 1) << ":\n";

writeEmployeeToFile(arr[i], file);

file << "\n";

}

file.close();

cout << "\nEmployee details have been written to employees.txt\n";

}

else {

cout << "\nUnable to open file";

}

delete[] arr;

return 0;

}

//Функция для заполнения структуры

void inputEmployee(Employee &employee) {

int vac;

cout << "Enter name: \n";

cin >> employee.name;

cout << "Enter vacancy (0 - Manager; 1 - Boss; 2 - Clerk; 3 - Salesman; 4 - Other): \n";

cin >> vac;

switch (vac) {

case 0:

employee.vacancy = Manager;

break;

case 1:

employee.vacancy = Boss;

break;

case 2:

employee.vacancy = Clerk;

break;

case 3:

employee.vacancy = Salesman;

break;

case 4:

employee.vacancy = Other;

break;

default:

cout << "Incorrect input";

employee.vacancy = Undefined;

}

cout << "Enter salary: \n";

cin >> employee.salary;

cout << "Enter hire date 'DD MM YYYY':\n";

cin >> employee.hireDate[0] >> employee.hireDate[1] >> employee.hireDate[2];

}

//Функция для вывода структуры

void outputEmployee(Employee &employee) {

cout << "Name: " << employee.name;

cout << "\nVacancy: ";

switch (employee.vacancy) {

case 0:

cout << "Manager"; break;

case 1:

cout << "Boss"; break;

case 2:

cout << "Clerk"; break;

case 3:

cout << "Salesman"; break;

case 4:

cout << "Other"; break;

case 5:

cout << "Undefined"; break;

}

cout << "\nSalary: " << employee.salary;

printf("\nHire date %d/%d/%d", employee.hireDate[0], employee.hireDate[1], employee.hireDate[2]);

}

//Функция для записи структуры в файл

void writeEmployeeToFile(Employee &employee, ofstream &file) {

file << "\nName: " << employee.name;

file << "\nVacancy: ";

switch (employee.vacancy) {

case 0:

file << "Manager"; break;

case 1:

file << "Boss"; break;

case 2:

file << "Clerk"; break;

case 3:

file << "Salesman"; break;

case 4:

file << "Other"; break;

case 5:

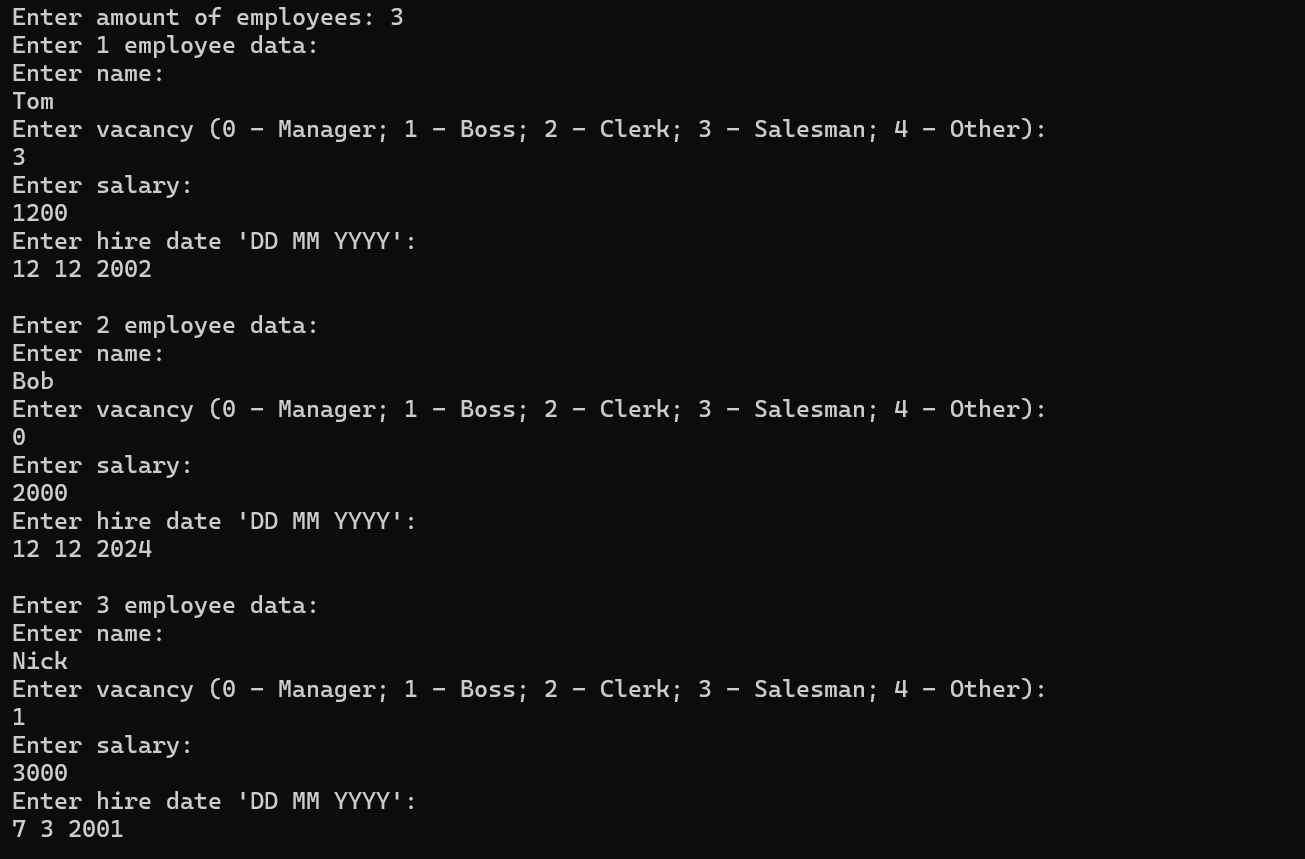
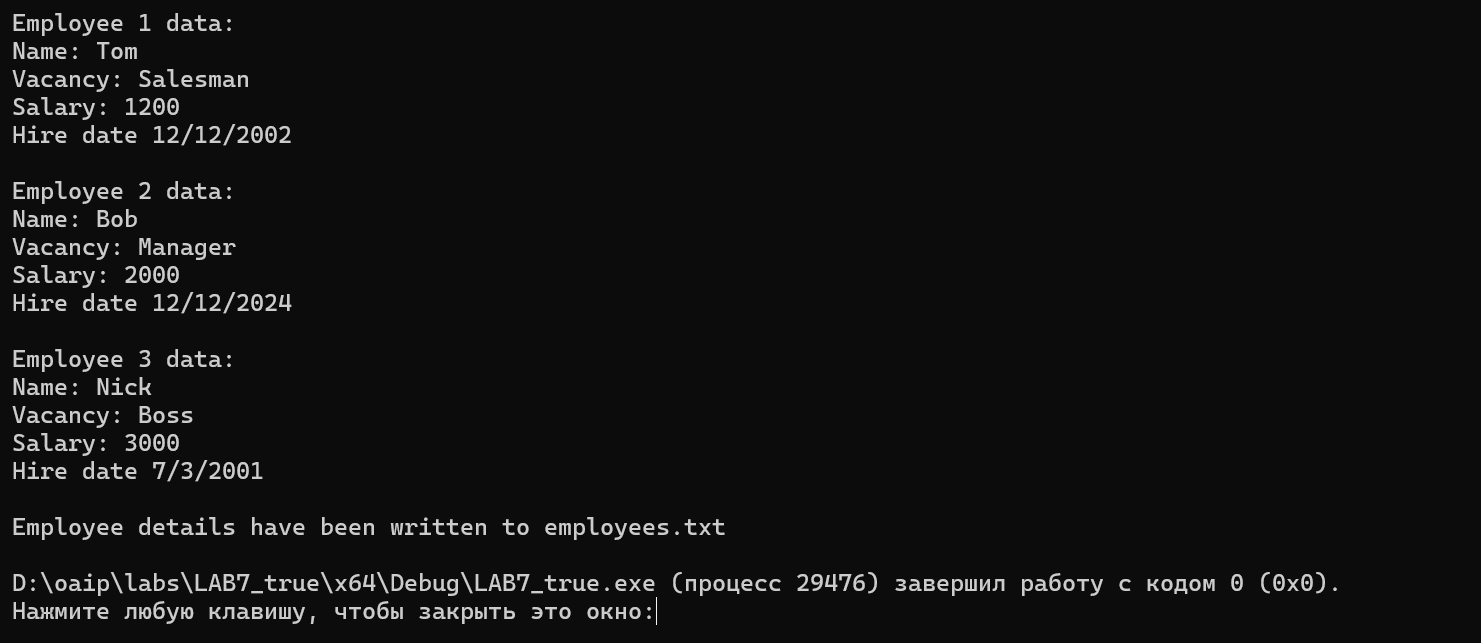
file << "Undefined"; break;

}

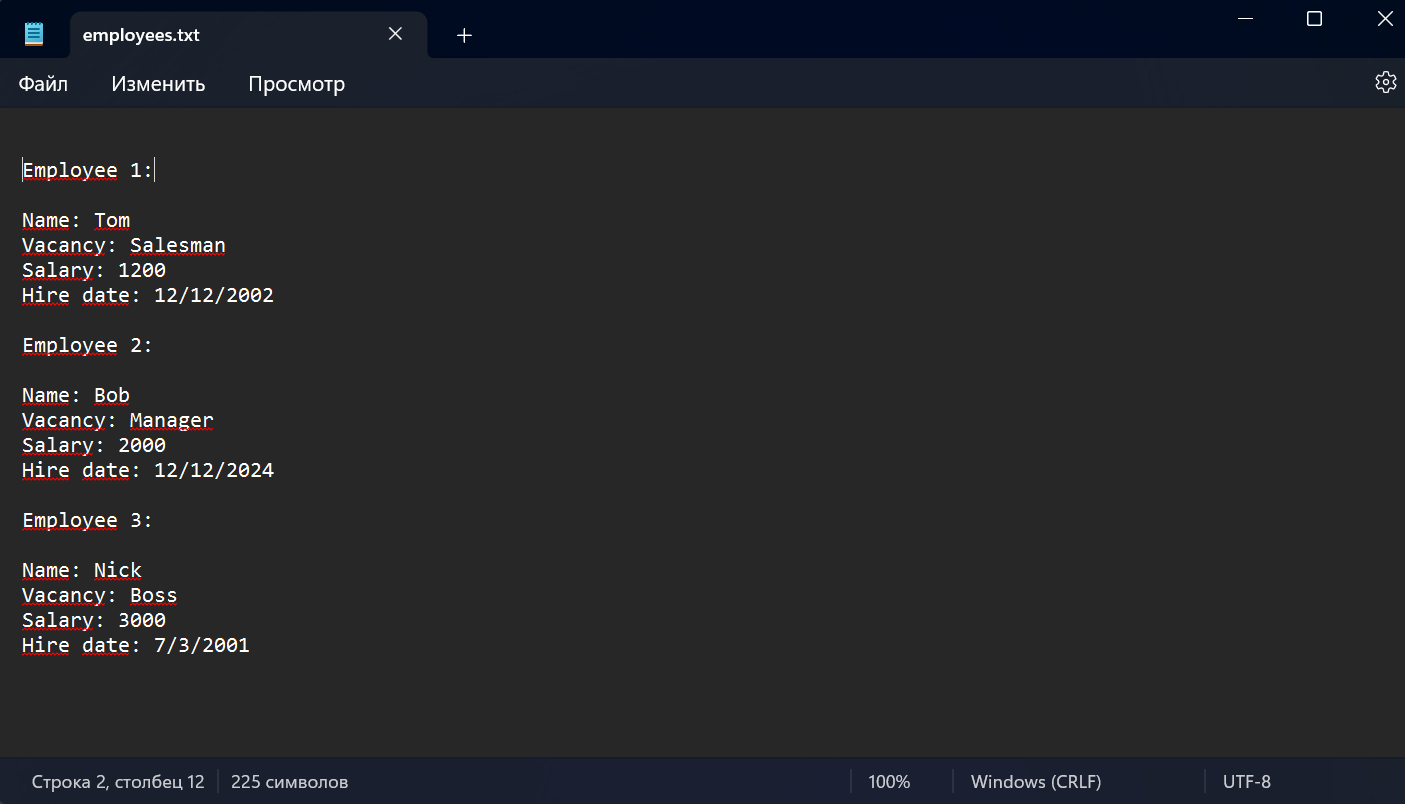
file << "\nSalary: " << employee.salary;

file << "\nHire date: " << employee.hireDate[0] << "/" << employee.hireDate[1] << "/" << employee.hireDate[2];

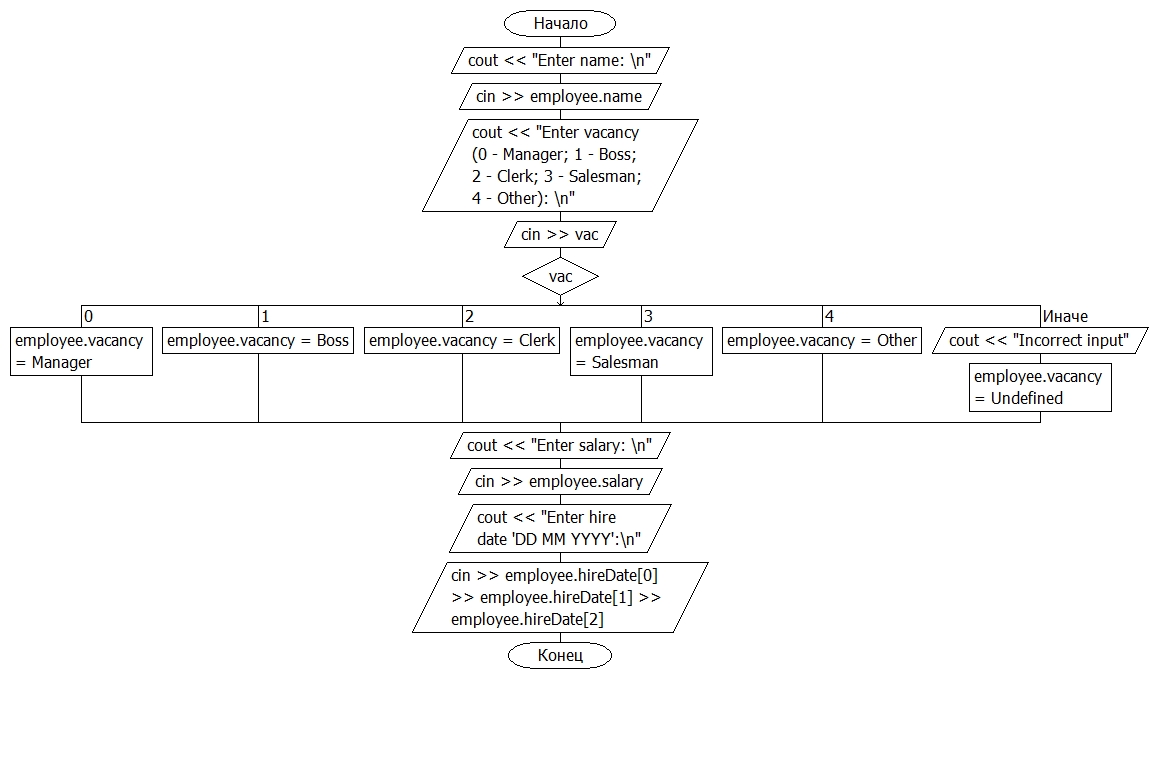
}

Результаты работы программы представлены на рисунках 1-2.

(Рис.1)

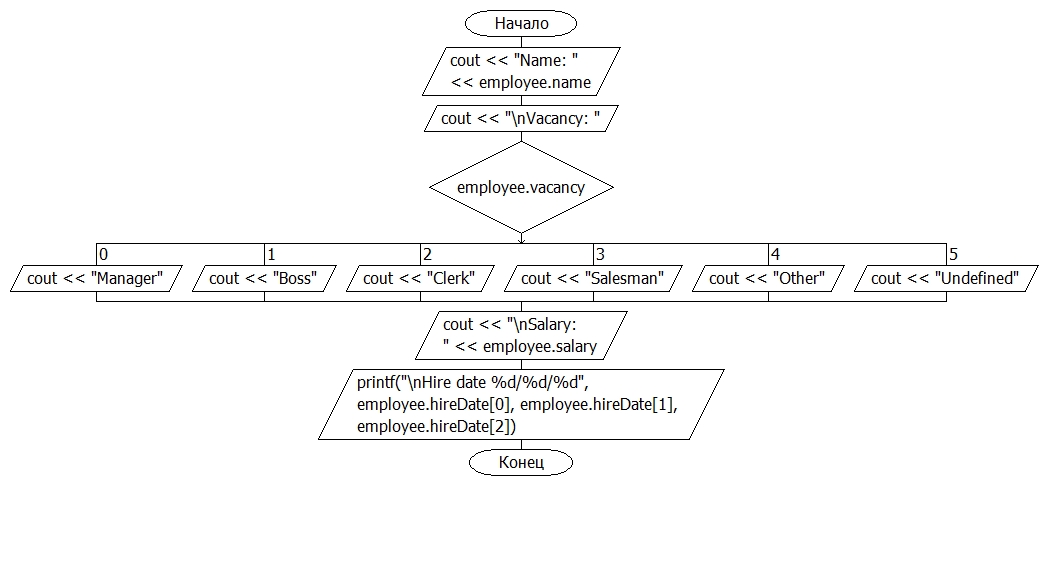
(Рис. 2)

Блок-схема функции для заполнения структуры представлена на рисунке 3.



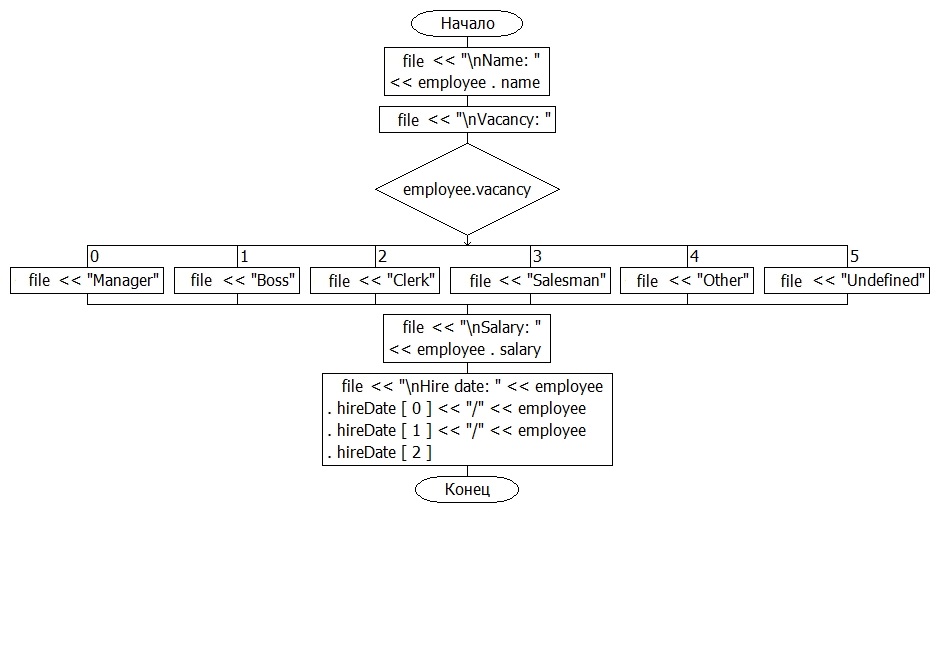
(Рис.3)

Блок-схема функции для вывода содержимого структуры представлена на рисунке 4.

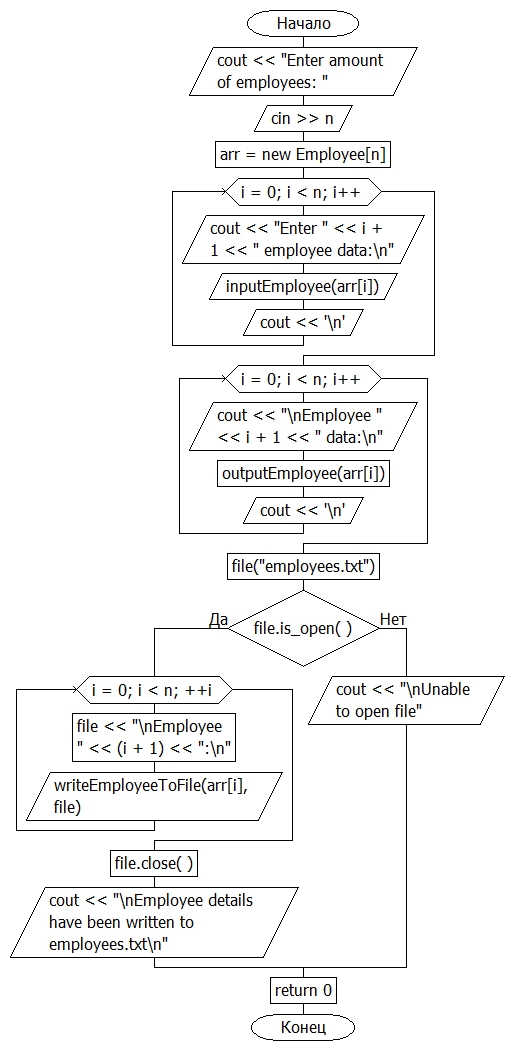


(Рис. 4)

Блок-схема для записи структуры в файл представлена на рисунке 5.

(Рис. 5)

Блок-схема работы программы представлена на рисунке 6.



***(Рис. 6)***

***Вывод***

В ходе выполнения работы была достигнута цель данной лабораторной работы:сформировать навыки и умения обработки структурированных типов данных, организованных в виде структур и файлов.