Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Программная инженерия»

Учебная дисциплина «Основы алгоритмизации и программирования»

Отчет

по лабораторной работе №7

«Структуры и файлы»

Подготовила: Студент гр. 410901

Бондаржевская А.А.

Проверил: Усенко Ф.В

Минск 2024

Цель: 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.

Листинг кода:

#include <iostream>

#include <string>

#include <Windows.h>

#include <fstream>

using namespace std;

enum class Vacancies

{

Boss,

Manager,

Clerk,

Salesman,

};

struct Employee

{

string name;

Vacancies vacancy{ Vacancies::Clerk };

int salary = 0;

int data[3] = { 0, 0, 0 };

};

Vacancies VacancyChoice();

Employee information();

string VacancyToString(Vacancies vacancy);

int main()

{

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

int n;

cout << "Введите количество сотрудников: ";

cin >> n;

Employee\* employees = new Employee[n];

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

{

cout << "\nСотрудник " << i + 1;

employees[i] = information();

}

ofstream fin;

fin.open("Employees.txt");

if (!fin.is\_open())

{

cout << "Не удалось открыть файл." << endl;

return 1;

}

fin << "-------------------------Информация о сотрудниках------------------------" << endl;

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

{

fin << "\nСотрудник " << i + 1 << ":" << endl;

fin << "Имя: " << employees[i].name << endl;

fin << "Должность: " << VacancyToString(employees[i].vacancy) << endl;

fin << "Зарплата: " << employees[i].salary << endl;

fin << "Дата приема на работу: " << employees[i].data[0] << "." << employees[i].data[1] << "." << employees[i].data[2] << endl;

}

fin.close();

ifstream fout;

fout.open("Employees.txt");

string line;

while (getline(fout, line))

{

cout << line << endl;

}

fout.close();

return 0;

}

Vacancies VacancyChoice()

{

int num;

cout << "1. Директор \n2. Менеджер \n3. Офисный работник \n4. Продавец" << endl;

cout << "\nВведите номер должности: ";

while (true)

{

cin >> num;

if ((num < 1) || (num > 4))

cout << "Некорректный ввод. Попробуйте еще раз." << endl;

else

break;

}

switch (num)

{

case 1: return Vacancies::Boss;

case 2: return Vacancies::Manager;

case 3: return Vacancies::Clerk;

case 4: return Vacancies::Salesman;

}

return Vacancies::Clerk;

}

Employee information()

{

Employee emp;

cout << "\nВведите имя сотрудника:" << endl;

cin.ignore();

getline(cin, emp.name);

cout << "\nВыберите должность сотрудника:" << endl;

emp.vacancy = VacancyChoice();

cout << "\nВведите зарплату сотрудника:" << endl;

cin >> emp.salary;

cout << "\nВведите дату приема сотрудника на работу:" << endl;

while (true)

{

cin >> emp.data[0];

if ((emp.data[0] < 1) || (emp.data[0] > 31))

cout << "Некорректный ввод. Попробуйте еще раз." << endl;

else

break;

}

while (true)

{

cin >> emp.data[1];

if ((emp.data[1] < 1) || (emp.data[1] > 12))

cout << "Некорректный ввод. Попробуйте еще раз." << endl;

else

break;

}

while (true)

{

cin >> emp.data[2];

if ((emp.data[2] < 1965) || (emp.data[2] > 2024))

cout << "Некорректный ввод. Попробуйте еще раз." << endl;

else

break;

}

return emp;

}

string VacancyToString(Vacancies vacancy)

{

switch (vacancy)

{

case Vacancies::Boss: return "Директор";

case Vacancies::Manager: return "Менеджер";

case Vacancies::Clerk: return "Офисный работник";

case Vacancies::Salesman: return "Продавец";

}

return "Офисный работник";

}

A screenshot of a computer

Description automatically generated

Рисунок 1 – Результат выполнения программы

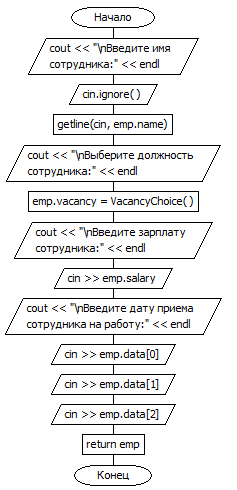
Построение блок-схемы:

Main:

A diagram of a program

Description automatically generated

Information:



Вывод: В ходе выполнения задания успешно создана программа, в которой пользователь вводит количество сотрудников, заполняет информацию о них. Данные, введенные пользвателем, записываются в файл и считываются из него.