Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**Отчет по Лабораторной работе № 8**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

Выполнил студент гр. ИВТ-21-1б

Ахунов Руслан Булатович

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Денис Владимирович

(оценка) (подпись)

г. Пермь - 2022

**Постановка задачи:**

Вариант 2

Структура "Государство":

- название;

- столица;

- численность населения;

- занимаемая площадь.

Удалить все элементы, у которых численность меньше

заданной, добавить элемент после элемента с указанным

номером.

**Анализ задачи:**

struct workers

{

    string name;

    string postname;

    string parton;

    string post;

    string birth;

    string money;

};

Структура workers, включает в себя 6 полей типа string.

void formation(int n)

{

    for (int i = 0; i < n; i++)

    {

        cout << "Введите Фамилию: ";

        cin >> line[i].postname;

        cout << "Введите Имя: ";

        cin >> line[i].name;

        cout << "Введите Отчество: ";

        cin >> line[i].parton;

        cout << "Введите Должность: ";

        cin >> line[i].post;

        cout << "Введите Дату Рождения: ";

        cin >> line[i].birth;

        cout << "Введите Заработную плату: ";

        cin >> line[i].money;

        cout << endl;

    }

}

Данная функция нужна для формирования переменных типа workers, где мы вводим все параметры.

cin >> del;

    for (int i = 0; i < n; i++)

    {

        if (line[i].postname == del)

        {

            tmp = i;

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

            {

                line2[j].postname = line[j].postname;

                line2[j].name = line[j].name;

                line2[j].parton = line[j].parton;

                line2[j].post = line[j].post;

                line2[j].birth = line[j].birth;

                line2[j].money = line[j].money;

            }

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

            {

                line2[j - 1].postname = line[j].postname;

                line2[j - 1].name = line[j].name;

                line2[j - 1].parton = line[j].parton;

                line2[j - 1].post = line[j].post;

                line2[j - 1].birth = line[j].birth;

                line2[j - 1].money = line[j].money;

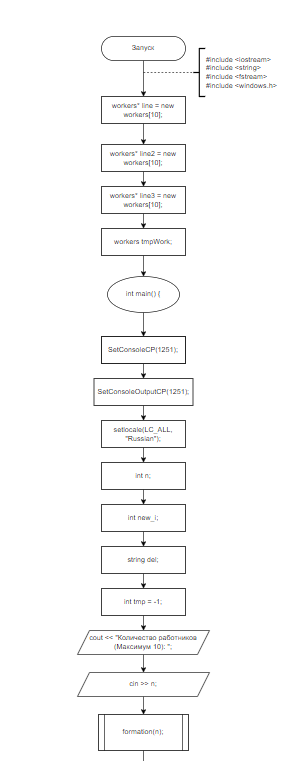
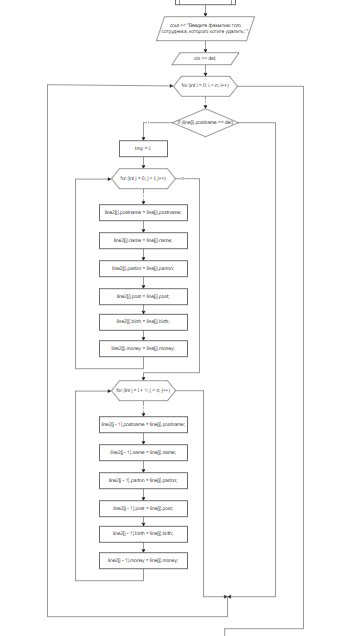
            }

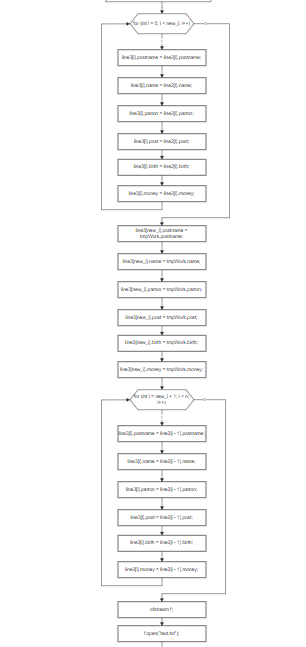
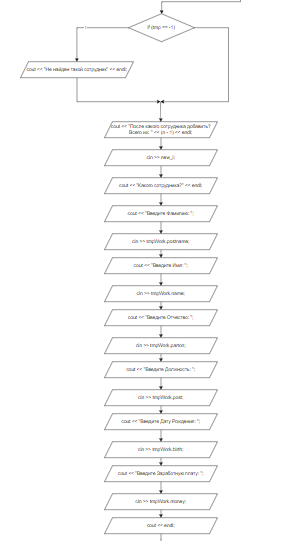
        }

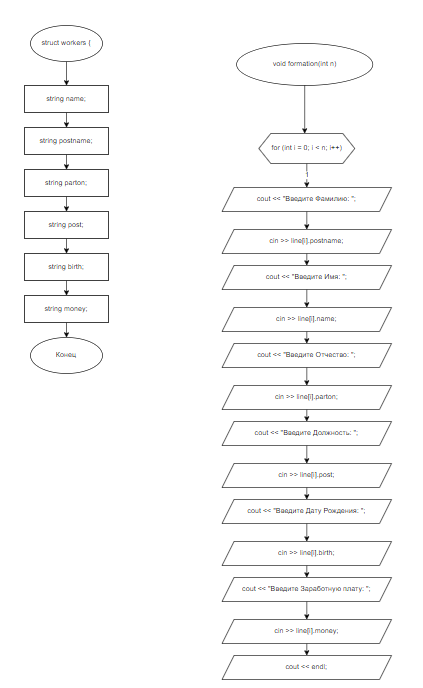
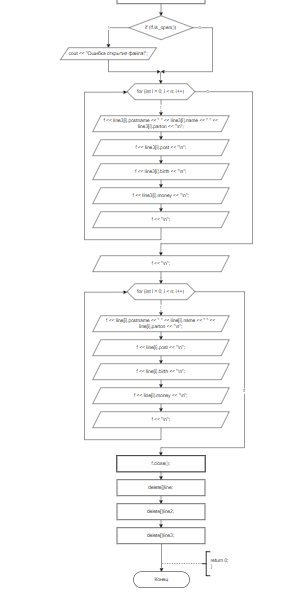
    }

Поиск работника по имени и последующие удаление.

**Блок-схема**

****

****

****

**Код программы**

#include <iostream>

#include <string>

#include <fstream>

#include <windows.h>

using namespace std;

struct workers

{

    string name;

    string postname;

    string parton;

    string post;

    string birth;

    string money;

};

workers\* line = new workers[10];

workers\* line2 = new workers[10];

workers\* line3 = new workers[10];

workers tmpWork;

void formation(int n)

{

    for (int i = 0; i < n; i++)

    {

        cout << "Введите Фамилию: ";

        cin >> line[i].postname;

        cout << "Введите Имя: ";

        cin >> line[i].name;

        cout << "Введите Отчество: ";

        cin >> line[i].parton;

        cout << "Введите Должность: ";

        cin >> line[i].post;

        cout << "Введите Дату Рождения: ";

        cin >> line[i].birth;

        cout << "Введите Заработную плату: ";

        cin >> line[i].money;

        cout << endl;

    }

}

int main()

{

    SetConsoleCP(1251);

    SetConsoleOutputCP(1251);

    setlocale(LC\_ALL, "Russian");

    int n;

    int new\_i;

    string del;

    int tmp = -1;

    cout << "Количество работников (Максимум 10): ";

    cin >> n;

    formation(n);

    cout << "Введите фамилию того сотрудника, которого хотите удалить: ";

    cin >> del;

    for (int i = 0; i < n; i++)

    {

        if (line[i].postname == del)

        {

            tmp = i;

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

            {

                line2[j].postname = line[j].postname;

                line2[j].name = line[j].name;

                line2[j].parton = line[j].parton;

                line2[j].post = line[j].post;

                line2[j].birth = line[j].birth;

                line2[j].money = line[j].money;

            }

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

            {

                line2[j - 1].postname = line[j].postname;

                line2[j - 1].name = line[j].name;

                line2[j - 1].parton = line[j].parton;

                line2[j - 1].post = line[j].post;

                line2[j - 1].birth = line[j].birth;

                line2[j - 1].money = line[j].money;

            }

        }

    }

    if (tmp == -1)

    {

        cout << "Не найден такой сотрудник" << endl;

    }

    cout << "После какого сотрудника добавить? Всего их: " << (n - 1) << endl;

    cin >> new\_i;

    cout << "Какого сотрудника?" << endl;

    cout << "Введите Фамилию: ";

    cin >> tmpWork.postname;

    cout << "Введите Имя: ";

    cin >> tmpWork.name;

    cout << "Введите Отчество: ";

    cin >> tmpWork.parton;

    cout << "Введите Должность: ";

    cin >> tmpWork.post;

    cout << "Введите Дату Рождения: ";

    cin >> tmpWork.birth;

    cout << "Введите Заработную плату: ";

    cin >> tmpWork.money;

    cout << endl;

    for (int i = 0; i < new\_i; i++)

    {

        line3[i].postname = line2[i].postname;

        line3[i].name = line2[i].name;

        line3[i].parton = line2[i].parton;

        line3[i].post = line2[i].post;

        line3[i].birth = line2[i].birth;

        line3[i].money = line2[i].money;

    }

    line3[new\_i].postname = tmpWork.postname;

    line3[new\_i].name = tmpWork.name;

    line3[new\_i].parton = tmpWork.parton;

    line3[new\_i].post = tmpWork.post;

    line3[new\_i].birth = tmpWork.birth;

    line3[new\_i].money = tmpWork.money;

    for (int i = new\_i + 1; i < n; i++)

    {

        line3[i].postname = line2[i - 1].postname;

        line3[i].name = line2[i - 1].name;

        line3[i].parton = line2[i - 1].parton;

        line3[i].post = line2[i - 1].post;

        line3[i].birth = line2[i - 1].birth;

        line3[i].money = line2[i - 1].money;

    }

    ofstream f;

    f.open("text.txt");

    if (!f.is\_open())

    {

        cout << "Ошибка открытия файла!";

    }

    for (int i = 0; i < n; i++)

    {

        f << line3[i].postname << " " << line3[i].name << " " << line3[i].parton << "\n";

        f << line3[i].post << "\n";

        f << line3[i].birth << "\n";

        f << line3[i].money << "\n";

        f << "\n";

    }

    f << "Изначальный вариант: " << "\n";

    f << "\n";

    for (int i = 0; i < n; i++)

    {

        f << line[i].postname << " " << line[i].name << " " << line[i].parton << "\n";

        f << line[i].post << "\n";

        f << line[i].birth << "\n";

        f << line[i].money << "\n";

        f << "\n";

    }

    f.close();

    delete[]line;

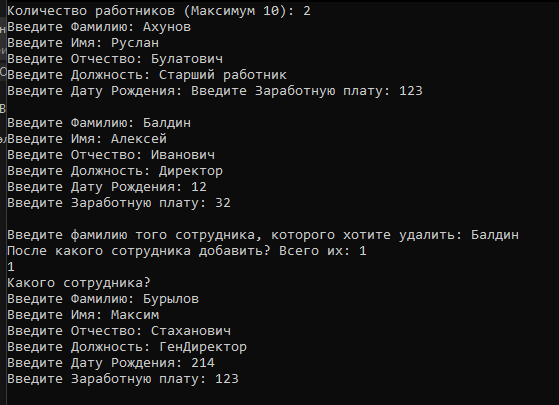
    delete[]line2;

    delete[]line3;

    return 0;

}

**Вывод программы**

****