

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Пермский национальный исследовательский политехнический университет»
(ПНИПУ)
Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
(ИТАС)

Лабораторная работа
на тему
«Блоковый ввод-вывод»

Выполнил
Студент группы ИВТ-23-16
Адаев Даниил Дмитриевич
Проверил
Доцент кафедры ИТАС
Д. В. Яруллин

Постановка задачи

Вариант 2

1.

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

2.

Структура "Сотрудник":

- фамилия, имя, отчество;
- должность
- год рождения;
- заработная плата.

Удалить элемент с указанной фамилией, добавить элемент после элемента с указанным номером.

Словесный алгоритм

Директива `#define _CRT_SECURE_NO_WARNINGS` используется для совместимости с классическими функциями.

Структура «Сотрудник» (`employee`) содержит данные о ФИО и должности в виде массива символов в количестве 100 и данные года рождения и зарплаты в целочисленном виде. Внутренние функции `printer` и `reader` выводят и считывают данные структуры соответственно.

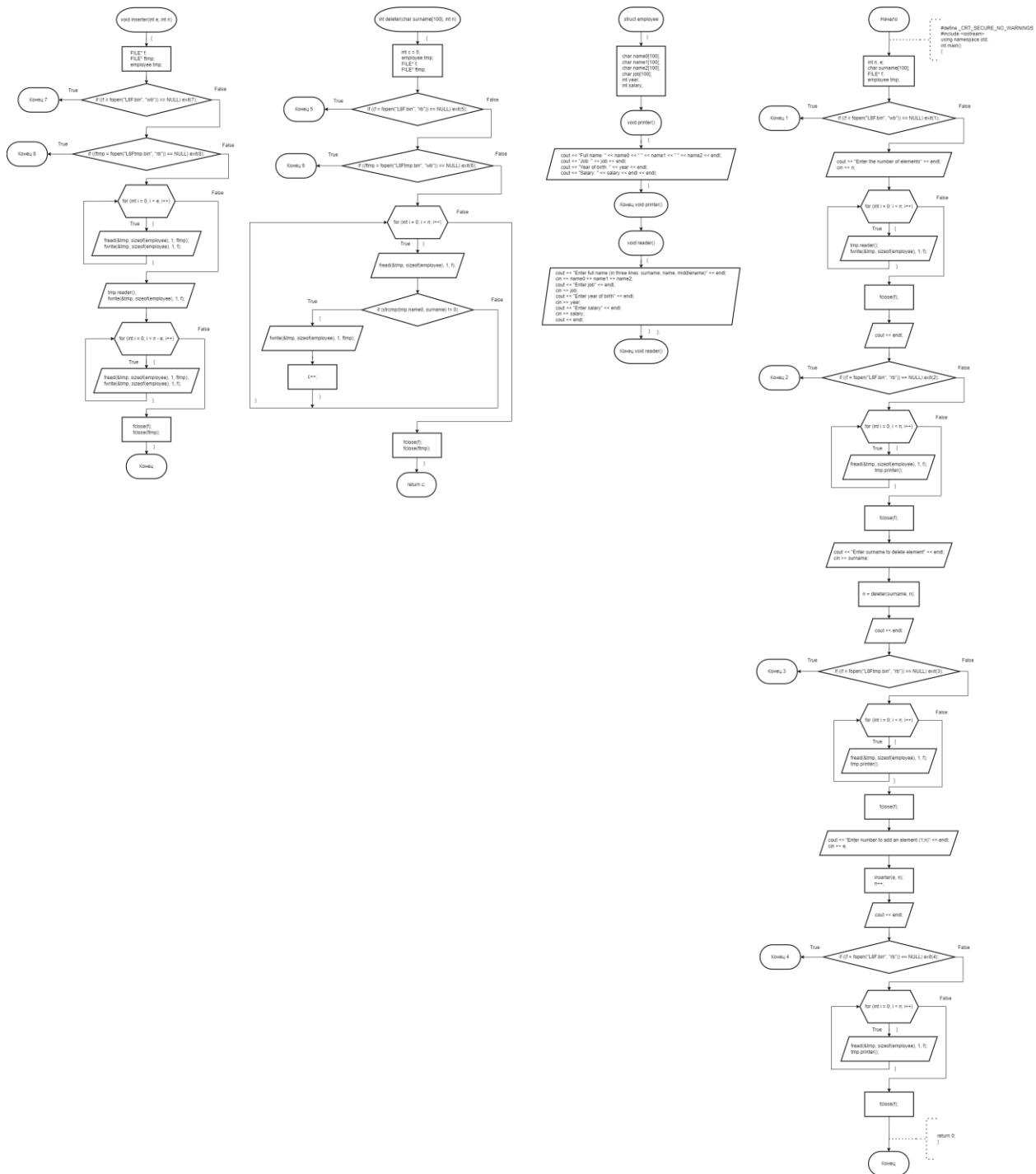
Функция `deleter` выполняет удаление элемента по фамилии. Считывается основной файл, и, если фамилия не совпадает с указанной для удаления, текущая структура записывается в дополнительный файл. Возвращает число структур в файле для дальнейшего считывания.

Функция `inserter` выполняет запись элемента после указанного номера. Элементы из временного файла считываются в основной до указанного номера включительно. Затем в основной файл записывается новый введенный элемент и продолжается запись оставшихся элементов из временного файла.

Главная функция считывает начальное число вводимых элементов, затем цикл столько же раз считывает элементы и записывает в основной файл. Затем выводятся все введенные элементы. Функция запрашивает фамилию для удаления. По указанной фамилии выполняется функция `deleter`, ее значение присваивается числу структур. Выводятся все элементы, записанные во временный файл. Функция запрашивает номер, после которого введется новый элемент. Выполняется функция `inserter`, числа структур увеличивается на 1. Выводятся все структуры из основного файла.

Каждое открытие файла сопровождается проверкой на ошибку. Алгоритм прерывается при ее обнаружении.

Блок-схема



Код программы

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>

using namespace std;

struct employee
{
    char name0[100];
    char name1[100];
    char name2[100];
    char job[100];
    int year;
    int salary;

    void printer()
    {
        cout << "Full name: " << name0 << " " << name1 << " " << name2 << endl;
        cout << "Job: " << job << endl;
        cout << "Year of birth: " << year << endl;
        cout << "Salary: " << salary << endl << endl;
    }

    void reader()
    {
        cout << "Enter full name (in three lines: surname, name, middlename)"
<< endl;
        cin >> name0 >> name1 >> name2;
        cout << "Enter job" << endl;
        cin >> job;
        cout << "Enter year of birth" << endl;
        cin >> year;
        cout << "Enter salary" << endl;
        cin >> salary;
        cout << endl;
    }
};

int deleter(char surname[100], int n)
{
    int c = 0;
    employee tmp;
    FILE* f;
    FILE* ftmp;

    if ((f = fopen("L8F.bin", "rb")) == NULL) exit(5);
    if ((ftmp = fopen("L8Ftmp.bin", "wb")) == NULL) exit(6);

    for (int i = 0; i < n; i++)
    {
        fread(&tmp, sizeof(employee), 1, f);

        if (strcmp(tmp.name0, surname) != 0)
        {
            fwrite(&tmp, sizeof(employee), 1, ftmp);
            c++;
        }
    }

    fclose(f);
    fclose(ftmp);

    return c;
}
```

```

void inserter(int e, int n)
{
    FILE* f;
    FILE* ftmp;
    employee tmp;

    if ((f = fopen("L8F.bin", "wb")) == NULL) exit(7);
    if ((ftmp = fopen("L8Ftmp.bin", "rb")) == NULL) exit(8);

    for (int i = 0; i < e; i++)
    {
        fread(&tmp, sizeof(employee), 1, ftmp);
        fwrite(&tmp, sizeof(employee), 1, f);
    }

    tmp.reader();
    fwrite(&tmp, sizeof(employee), 1, f);

    for (int i = 0; i < n - e; i++)
    {
        fread(&tmp, sizeof(employee), 1, ftmp);
        fwrite(&tmp, sizeof(employee), 1, f);
    }

    fclose(f);
    fclose(ftmp);
}

int main()
{
    int n, e;
    char surname[100];
    FILE* f;
    employee tmp;

    if ((f = fopen("L8F.bin", "wb")) == NULL) exit(1);

    cout << "Enter the number of elements" << endl;
    cin >> n;

    for (int i = 0; i < n; i++)
    {
        tmp.reader();
        fwrite(&tmp, sizeof(employee), 1, f);
    }

    fclose(f);

    cout << endl;

    if ((f = fopen("L8F.bin", "rb")) == NULL) exit(2);

    for (int i = 0; i < n; i++)
    {
        fread(&tmp, sizeof(employee), 1, f);
        tmp.printer();
    }

    fclose(f);

    cout << "Enter surname to delete element" << endl;
    cin >> surname;

    n = deleter(surname, n);
}

```

```

    cout << endl;

    if ((f = fopen("L8Ftmp.bin", "rb")) == NULL) exit(3);

    for (int i = 0; i < n; i++)
    {
        fread(&tmp, sizeof(employee), 1, f);
        tmp.printer();
    }

    fclose(f);

    cout << "Enter number to add an element (1;n)" << endl;
    cin >> e;

    inserter(e, n);
    n++;

    cout << endl;

    if ((f = fopen("L8F.bin", "rb")) == NULL) exit(4);

    for (int i = 0; i < n; i++)
    {
        fread(&tmp, sizeof(employee), 1, f);
        tmp.printer();
    }

    fclose(f);

    return 0;
}

```

Выводы


```
Enter the number of elements
3
Enter full name (in three lines: surname, name, middlename)
1
1
1
Enter job
1
Enter year of birth
1
Enter salary
1

Enter full name (in three lines: surname, name, middlename)
2
2
2
Enter job
2
Enter year of birth
2
Enter salary
2

Enter full name (in three lines: surname, name, middlename)
3
3
3
Enter job
3
Enter year of birth
3
Enter salary
3

Full name: 1 1 1
Job: 1
Year of birth: 1
Salary: 1

Full name: 2 2 2
Job: 2
Year of birth: 2
Salary: 2

Full name: 3 3 3
Job: 3
Year of birth: 3
Salary: 3
```

```
Enter surname to delete element
2

Full name: 1 1 1
Job: 1
Year of birth: 1
Salary: 1

Full name: 3 3 3
Job: 3
Year of birth: 3
Salary: 3

Enter number to add an element (1;n)
2
Enter full name (in three lines: surname, name, middlename)
4
4
4
Enter job
4
Enter year of birth
4
Enter salary
4

Full name: 1 1 1
Job: 1
Year of birth: 1
Salary: 1

Full name: 3 3 3
Job: 3
Year of birth: 3
Salary: 3

Full name: 4 4 4
Job: 4
Year of birth: 4
Salary: 4
```