

Практическая работа № 8

ФАЙЛЫ

Постановка задачи

Создать программные модули с операциями над двоичными и текстовыми файлами для выполнения задания варианта. Двоичный файл состоит из записей определенной структуры. Записи имеют ключ, уникальный в пределах файла. Для выполнения задания использовать потоковые файлы C++.

1 - Преобразовать данные в двоичный файл

2 - Отобразить все записи двоичного файла

3 - Удаление записей по начальному фрагменту ключа

4 - Заменить не ключевой параметр у нескольких записей по ключу

Перечисленные действия оформить в виде самостоятельных режимов работы созданного дека. Выбор режимов производить с помощью пользовательского меню.

Провести полное тестирование (всех режимов работы) программы на стеке, сформированном вводом с клавиатуры. Тест-примеры определить самостоятельно. Результаты тестирования в виде скриншотов экранов включить в отчет по выполненной работе.

Оформить отчет с подробным описанием созданного алгоритма, принципов программной реализации алгоритмов работы с файлами, описанием текста исходного кода и проведенного тестирования программы.

Сделать выводы о проделанной работе, основанные на полученных результатах.

1. Описание алгоритма

Алгоритм программы состоит из функции `main` и вызываемых в ней вспомогательных функций:

- **`void read_file`** – функция считывания товаров файла.
- **`void add_in_file`** – функция добавления товара в файл.
- **`int search_in_list_id`** – функция поиска товара в списке.
- **`int search_in_list_part_id`** – функция поиска по началу ключа.
- **`void del_elems_part_id`** – функция удаления товаров по началу ключа.
- **`void change_elem_id`** – функция изменения товара по ключу.
- **`void change_elems`** – функция именованная нескольких товаров.
- **`void file_update()`** – функция обновления файла.
- **`void out_list()`** – функция вывода всех товаров из файла.

Для работы с файлами в языке `c++` используется класс `fstream`.

`ifstream` для считывания файла, `ofstream` для его записи.

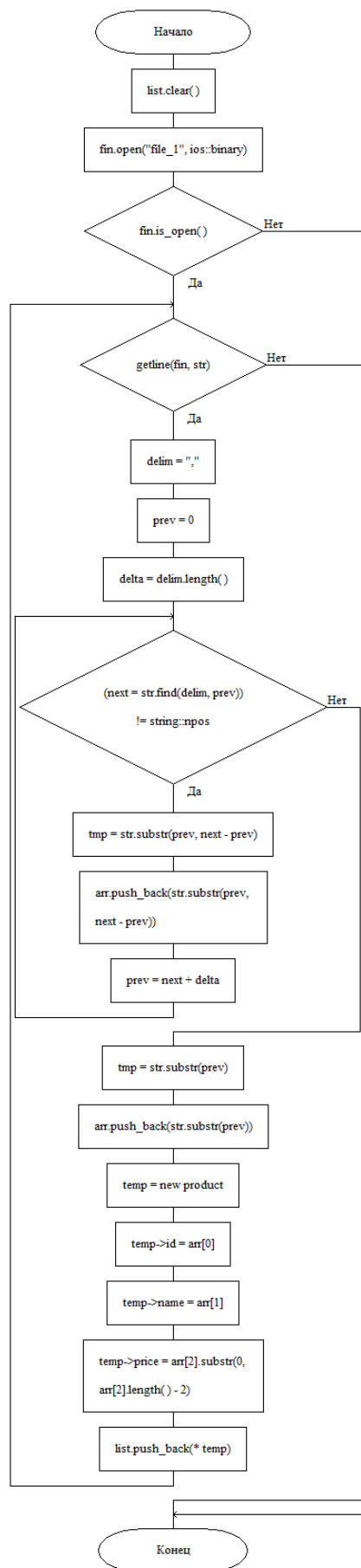


Рис.1 Схема алгоритма функции read_file

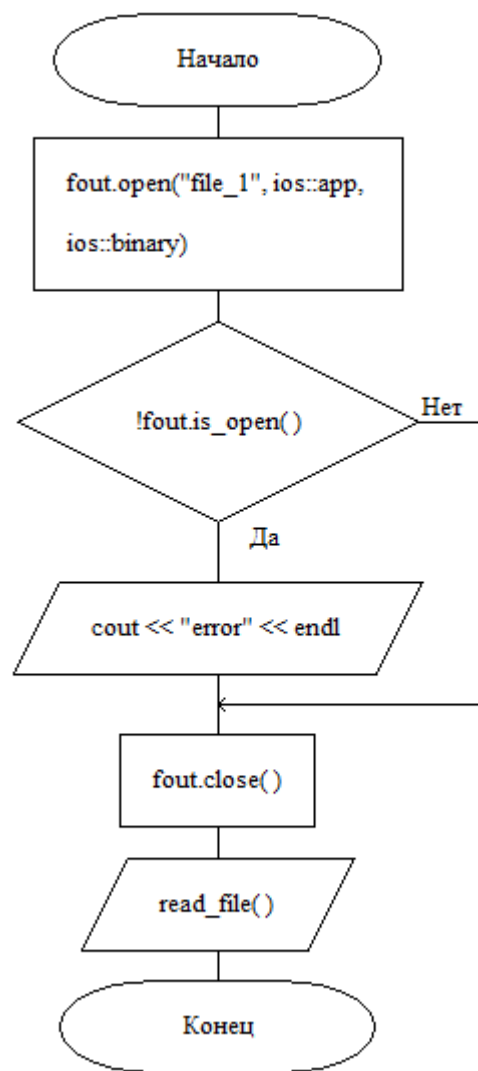


Рис.2 Схема алгоритма функции `add_in_file`

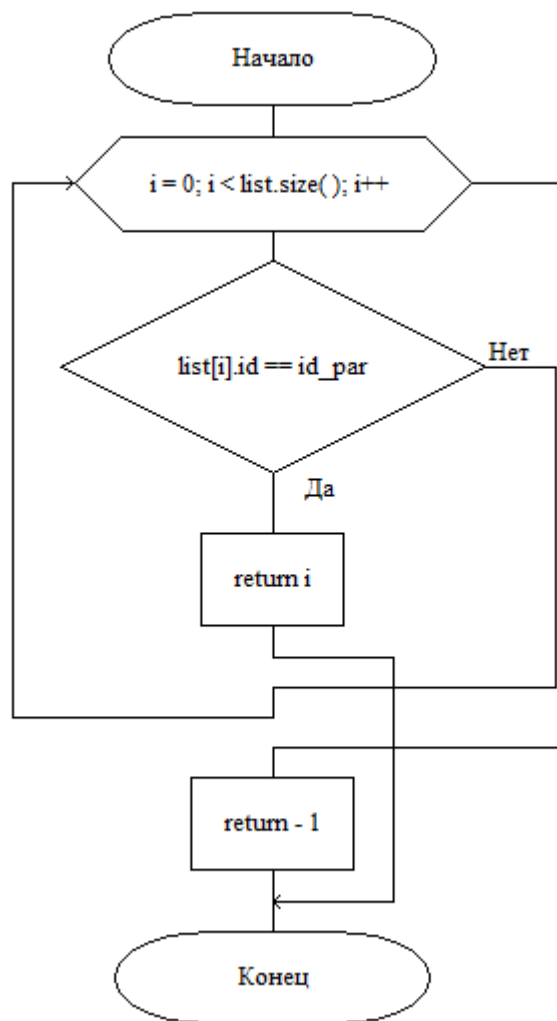


Рис.3 Схема алгоритма функции `search_in_list_id`

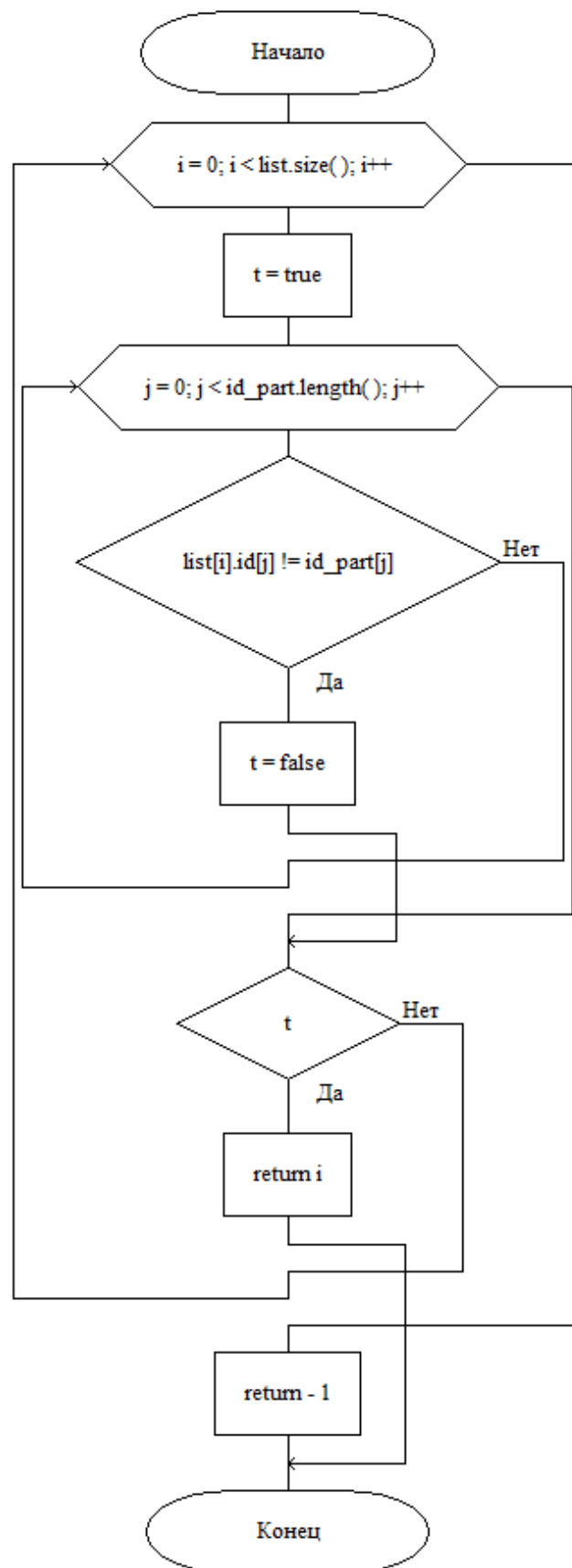


Рис.4 Схема алгоритма функции `search_in_list_part_id`

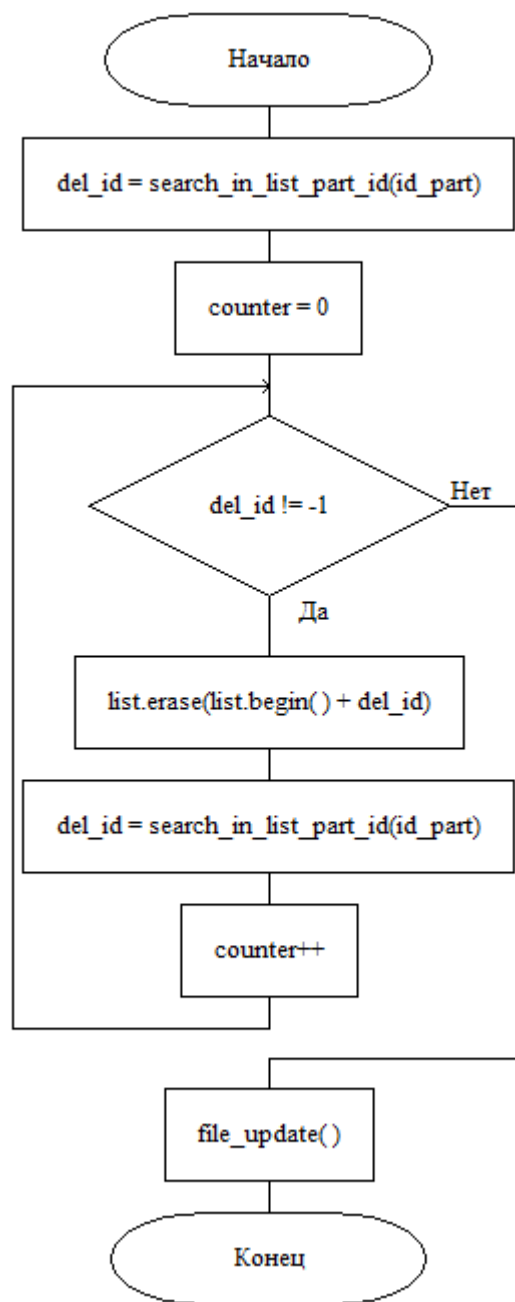


Рис.5 Схема алгоритма функции `del_elems_part_id`

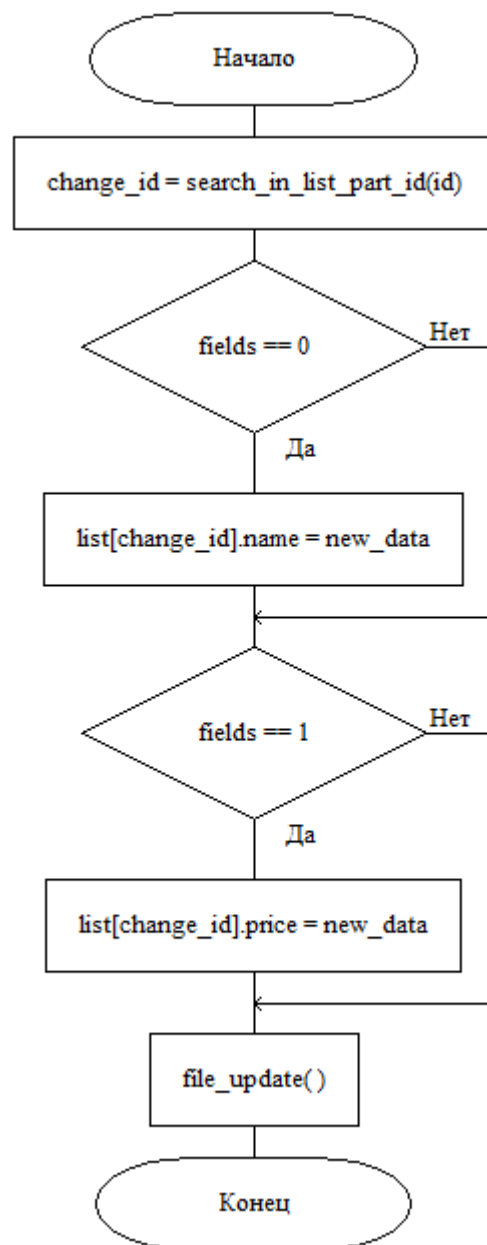


Рис.6 Схема алгоритма функции `change_elem_id`

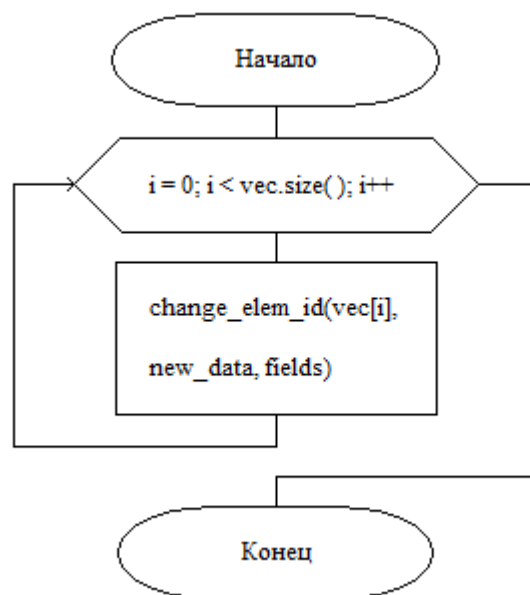


Рис.7 Схема алгоритма функции change_elems

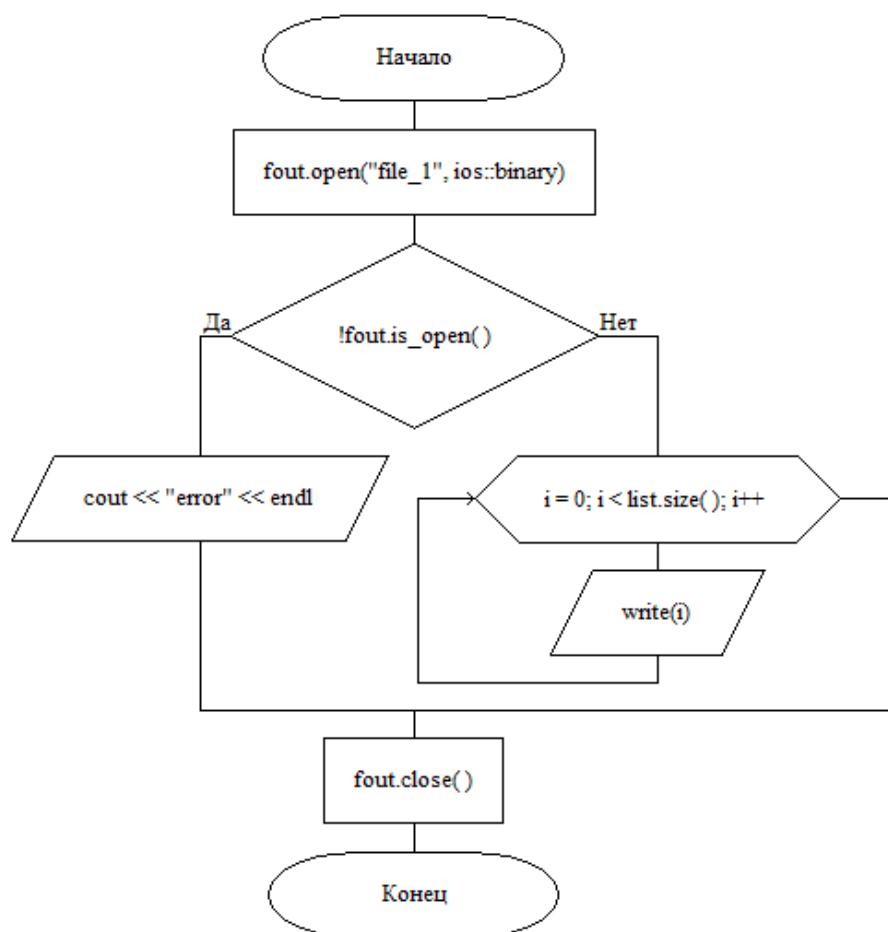


Рис.8 Схема алгоритма функции file_update

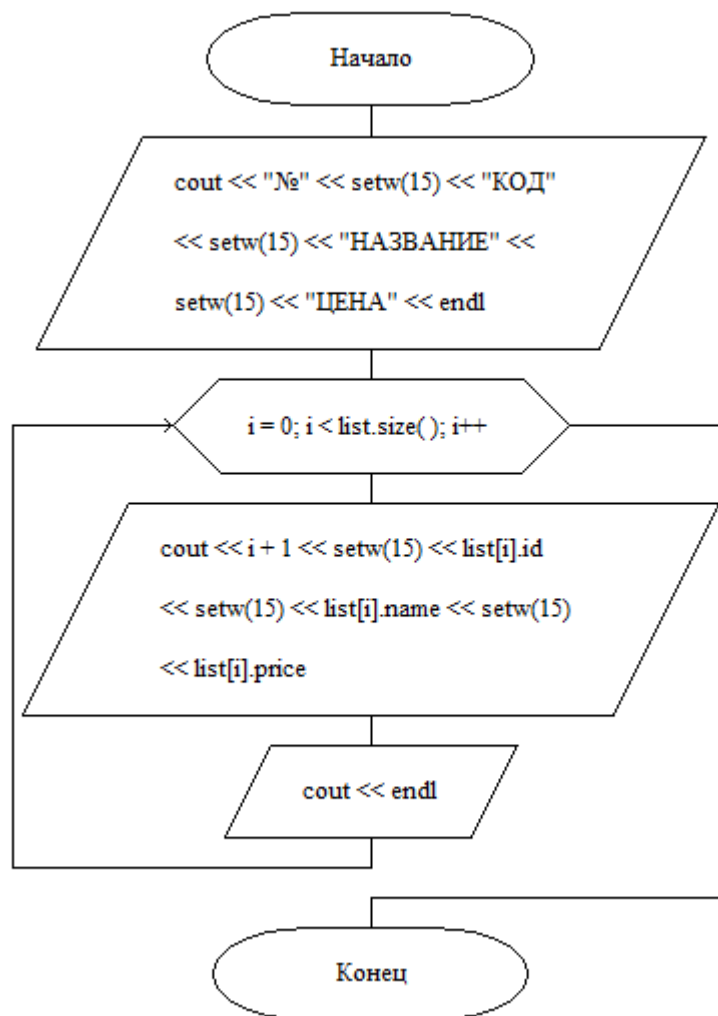


Рис.9 Схема алгоритма функции `out_list()`

Реализация алгоритма

Текст исходного кода программы

main.cpp

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <iomanip>
using namespace std;
void file_update();
struct product{
    string id;
    string name;
    string price;
};
vector<product> list;

void read_file() {
    list.clear();
    string str;
    ifstream fin;
    fin.open("file_1", ios::binary);
    if (fin.is_open()) {
        while (getline(fin, str))
        {
            vector<string> arr;
            size_t next;
            string delim = ",";
            size_t prev = 0;
            size_t delta = delim.length();
            while ((next = str.find(delim, prev)) != string::npos) {
                string tmp = str.substr(prev, next - prev);
                arr.push_back(str.substr(prev, next - prev));
                prev = next + delta;
            }
            string tmp = str.substr(prev);
            arr.push_back(str.substr(prev));
            product* temp = new product;
            temp->id = arr[0];
            temp->name = arr[1];
            temp->price = arr[2].substr(0, arr[2].length()-1);
            list.push_back(*temp);
        }
    }
}

void add_in_file(string s) {
    ofstream fout;
    fout.open("file_1", ios::app, ios::binary);
    if (!fout.is_open()) cout << "error" << endl;
    else {
        fout << s << endl;
    }
    fout.close();
    read_file();
}
```

```

int search_in_list_id(string id_par) {
    for (int i = 0; i < list.size(); i++) {
        if (list[i].id == id_par) {
            return i;
        }
    }
    return -1;
}

int search_in_list_part_id(string id_part) {
    for (int i = 0; i < list.size(); i++) {
        bool t = true;
        for (int j = 0; j < id_part.length(); j++) {
            if (list[i].id[j] != id_part[j]) {
                t = false;
                break;
            }
        }
        if (t) return i;
    }
    return -1;
}

void del_elems_part_id(string id_part) {
    int del_id = search_in_list_part_id(id_part);
    int counter = 0;
    while (del_id != -1) {
        list.erase(list.begin() + del_id);
        del_id = search_in_list_part_id(id_part);
        counter++;
    }
    file_update();
    cout << "Удалено товаров: " << counter << endl;
}

void change_elem_id(string id, string new_data, int fields) {
    int change_id = search_in_list_part_id(id);

    if(fields == 0){
        list[change_id].name = new_data;
    }
    if (fields == 1) {
        list[change_id].price = new_data;
    }
    file_update();
}

void change_elems(vector<string> vec, string new_data, int fields) {
    for (int i = 0; i < vec.size(); i++) {
        change_elem_id(vec[i], new_data, fields);
    }
    cout << "Элементов изменено: " << vec.size() << endl;
}

void file_update() {
    ofstream fout;
    fout.open("file_1", ios::binary);
    if (!fout.is_open()) cout << "error" << endl;
    else {
        for (int i = 0; i < list.size(); i++) {
            fout << list[i].id + "," + list[i].name + "," + list[i].price << endl;
        }
    }
}

```

```

        fout.close();
    }

    void out_list() {
        cout << "№" << setw(15) << "КОД" << setw(15) << "НАЗВАНИЕ" << setw(15) << "ЦЕНА" << endl;
        for (int i = 0; i < list.size(); i++) {
            cout << i+1 << setw(15) << list[i].id << setw(15) << list[i].name << setw(15) << list[i].price;
            cout << endl;
        }
    }

    int main()
    {
        setlocale(LC_ALL, "Russian");
        ofstream fout;
        fout.open("file_1", ios::binary);
        if (!fout.is_open()) cout << "error" << endl;
        else {
            fout << "";
        }
        fout.close();

        while (true) {
            cout << "Выберите команду:" << endl;
            cout << "[1] - Добавить товар." << endl;
            cout << "[2] - Удалить товары по началу ключа." << endl;
            cout << "[3] - Изменить несколько товаров." << endl;
            cout << "[4] - Вывести список товаров." << endl;
            cout << "[5] - Завершить программу." << endl;
            cout << "---->";
            int ch = 0;
            cin >> ch;
            if (ch == 1) {
                cout << "____Введите НОМЕР, НАЗВАНИЕ и ЦЕНУ товара через запятые без пробелов" << endl;
                cout << "----->";
                string new_str;
                cin >> new_str;
                add_in_file(new_str);
                continue;
            }
            if (ch == 2) {
                cout << "____Введите начало ключа" << endl;
                cout << "----->";
                string new_key;
                cin >> new_key;
                del_elems_part_id(new_key);
                continue;
            }
            if (ch == 3) {
                cout << "____Введите через пробел номера товаров (0 для завершения)" << endl;
                cout << "----->";
                string new_id = "";
                vector<string> ids;
                cin >> new_id;
                while (new_id != "0") {
                    ids.push_back(new_id);
                    cin >> new_id;
                }
                cout << "____Введите новые данные и название поля для вставки" << endl;
                cout << "----->";
                string data;
            }
        }
    }
}

```

```
        int field;
        cin >> data >> field;
        change_elems(ids, data, field);
        continue;
    }
    if (ch == 4) {
        out_list();
        continue;
    }
    if (ch == 5) {
        break;
    }
}
}
```

2. Тестирование программы

Ниже представлен результат работы программы с бинарным файлом

```
Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->1
Введите НОМЕР, НАЗВАНИЕ и ЦЕНУ товара через запятые без пробелов
----->248973,AAAA,100
Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->1
Введите НОМЕР, НАЗВАНИЕ и ЦЕНУ товара через запятые без пробелов
----->388923,BBBB,250
Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->1
Введите НОМЕР, НАЗВАНИЕ и ЦЕНУ товара через запятые без пробелов
----->893212,CCCCC,1000
Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->1
Введите НОМЕР, НАЗВАНИЕ и ЦЕНУ товара через запятые без пробелов
----->394324,DDD,15000
```

Рис.8 Скриншот добавления товаров в файл

```

Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->4
№      КОД      НАЗВАНИЕ      ЦЕНА
1      248973      ААААА      100
2      388923      ВВВВ      250
3      893212      СССССС      1000
4      394324      DDD      15000

```

Рис.9 Скриншот вывода списка на экран

```

Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->3
Введите через пробел номера товаров (0 для завершения)
----->248973 394324 0
Введите новые данные и название поля для вставки
----->NEW 0
Элементов изменено: 2
Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->4
№      КОД      НАЗВАНИЕ      ЦЕНА
1      248973      NEW      100
2      388923      ВВВВ      250
3      893212      СССССС      1000
4      394324      NEW      15000

```

Рис.10 Скриншот изменения названия товаров по ключу и вывод


```

Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->3
Введите через пробел номера товаров (0 для завершения)
----->388923 893212 394324 0
Введите новые данные и название поля для вставки
----->666 1
Элементов изменено: 3
Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->4

```

№	КОД	НАЗВАНИЕ	ЦЕНА
1	248973	NEW	100
2	388923	BBBB	666
3	893212	CCCCC	666
4	394324	NEW	666

Рис.10 Скриншот изменения цены товаров по ключу и вывод

```

Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->2
Введите начало ключа
----->3
Удалено товаров: 2
Выберите команду:
[1] - Добавить товар.
[2] - Удалить товары по началу ключа.
[3] - Изменить несколько товаров.
[4] - Вывести список товаров.
[5] - Завершить программу.
---->4

```

№	КОД	НАЗВАНИЕ	ЦЕНА
1	248973	NEW	100
2	893212	CCCCC	666

Рис.10 Скриншот удаления товаров по началу ключа

Выводы

1. В ходе работы была создана программа для работы с файлами.
2. Также были реализованы функции записи, чтения, удаления и изменения данных в файле.
3. Были изучены преимущества и недостатки хранения данных в файле:
4. Преимущества: данные в файлах могут храниться даже если устройство отключено от сети. В файлы можно записать гораздо больший объем данных чем в оперативную память
5. Недостатки: для работы необходимо постоянно обращаться к файловой системе.
6. Таким образом, была изучена работа алгоритмов обработки файлов

Список используемых информационных источников

1. Сыромятников В.П. Структуры и алгоритмы обработки данных, лекции, РТУ МИРЭА, Москва, 2020/2021 уч./год.
2. Документация по языку программирования C++, интернет-ресурс: <https://en.cppreference.com/w/> (Дата обращения – 02.11.2020)
3. Интегрированная среда разработки для языков программирования C и C++, разработанная компанией JetBrains - CLion / Copyright © 2000-2020 JetBrains s.r.o., интернет-ресурс: <https://www.jetbrains.com/clion/learning-center/> (Дата обращения – 02.11.2020).
4. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. Интернет-ресурс: <http://docs.cntd.ru/document/gost-19-701-90-espd> (Дата обращения – 02.11.2020).
5. Описание файлов. интернет-ресурс: <https://ru.wikipedia.org/wiki/Файл> (Дата обращения – 02.11.2020).