**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**Ордена Трудового Красного Знамени государственное образовательное учреждение высшего профессионального образования**

**Московский технический университет связи и информатики**

Кафедра информатики

**КУРСОВАЯ РАБОТА**

По дисциплине **«Языки программирования»**

Выполнил:

студент 1 курса,

по специальности 10.03.01

Информационная безопасность,

группы БИБ2203,

Блёсткин Михаил Сергеевич

Руководитель:

М. К. Скрыпникова,

кандидат технических наук,

доцент

Москва 2023

ОГЛАВЛЕНИЕ

[ЗАДАНИЕ 3](#_Toc134365559)

[ПОЯСНИТЕЛЬНАЯ ЗАПИСКА 4](#_Toc134365560)

[СХЕМА ВЗАИМОСВЯЗЕЙ ВСЕХ ФУНКЦИЙ 6](#_Toc134365561)

[ТЕКСТ ЗАГОЛОВОЧНОГО ФАЙЛА 7](#_Toc134365562)

[**Header.h** 7](#_Toc134365563)

[ТЕКСТЫ ПРОГРАММНЫХ МОДУЛЕЙ 8](#_Toc134365564)

[**funcs.cpp:** 8](#_Toc134365565)

[**main.cpp:** 15](#_Toc134365566)

[**tabl.cpp:** 16](#_Toc134365567)

[РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ 18](#_Toc134365568)

[**1)** **Чтение всех записей** 18](#_Toc134365569)

[**2)** **Добавить запись** 19](#_Toc134365570)

[**3)** **Пометить на удаление** 20](#_Toc134365571)

[**4)** **Снять пометки на удаление** 21](#_Toc134365572)

[**5)** **Чтение записи с N номером** 22](#_Toc134365573)

[**6)** **Список стран** 22](#_Toc134365574)

[**7)** **Удалить выбранную запись** 23](#_Toc134365575)

[**8)** **Закрытие программы** 24](#_Toc134365576)

[ВЫВОД 25](#_Toc134365577)

# ЗАДАНИЕ

Министерство торговли маленькой африканской страны заказало программу обработки экспортируемых товаров. В файле о каждом из товаров заводится запись следующего содержания: наименование товара, страна, куда экспортируется, объем поставляемой партии в штуках, цена единицы товара. Программа составляет список стран, куда экспортируется каждый из экспортируемых товаров, с указанием общего объема экспорта и общей стоимости. Программа называет товар, который приносит наибольший доход в казну государства.

# ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Курсовая работа включает в себя:

* Заголовочный файл: **header.h**
* Программные модули (файлы):

**funcs.cpp –** файл, содержащий вычислительные функции

**main.cpp –** главный файл

**tabl.cpp –** файл, строящий элементы таблицы записей

Разработанные функции в файле funcs.cpp:

1. **void write(char\* fname1, Product zap, int num)** – запись одного запроса в конец файла или в указанною строку (стр. 7);
2. **void print(Product zap, int i)** – вывод в консоль номер i и всей структуры;
3. **void print2(Product zap, int i, int volum\_p, double price\_p, double max\_p)** – вывод в консоль часть структуры (zap.country и zap.title) и значений переменных volum\_p, price\_p и max\_p в заданном порядке;
4. **Product vvod()** – ввод данных структуры (стр. 7);
5. **void writeNzap(int n)** – запись n строк в файл;
6. **fstream& read(Product& zap, int num)** – чтение num-й строки из файла;
7. **void readAll()** – чтение всех строк из файла (стр. 8);
8. **void readN(int N)** – чтение N строки из файла (стр. 8);
9. **int metkaDel(char\* title)** – ставит метку на удаление: меняет метку выбранной записи с '+' на '-';
10. **void UNmetkaDel()** – снимает все метки на удаление: меняет метку записи с '-' на '+';
11. **int Delete()** – удаление всех записей из файла с отрицательной меткой;
12. **double max\_profit(char country[], bool f = false)** – вычисляет максимальную прибыль от продажи всей партии товара по заданной цене среди всех товаров;
13. **int all\_profit(char country[], double& price, int i)** – считает общий объём экспортируемых товаров и общую стоимость экспортируемых товаров для конкретной страны. Функция считает не все запросы в файле, а только те, которые начинаются с номера i;
14. **void find\_title(char country[], int& kol, int volume\_p, double price\_p, double max\_p, bool f = false)** – находит товар, который приносит наибольший доход в казну государства, т.е. имеет наибольшую прибыль. Функция может считать как все запросы, так и только те, которые принадлежат конкретной стране;
15. **void spiski()** – составляет список стран, куда экспортируется каждый из экспортируемых товаров, с указанием общего объема экспорта и общей стоимости, а также выводит самый прибыльный товар среди всех экспортируемых товаров;

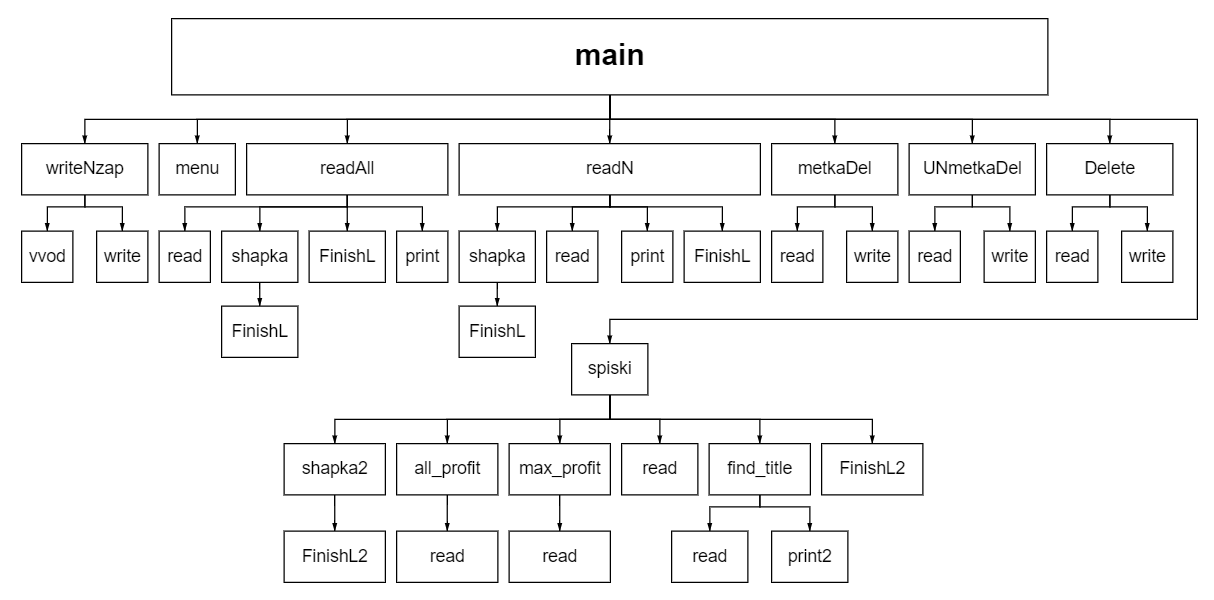
Разработанные функции в файле main.cpp:

1. **char menu()** – выводит меню выбора действий в консоль;
2. **void main()** – главная функция;

Разработанные функции в файле tabl.cpp:

1. **ostream& FinishL(ostream& stream)** – формирования верхней и нижней границ для первой таблицы (стр. 13);
2. **ostream& FinishL2(ostream& stream)** – формирования верхней и нижней границ для второй таблицы (стр. 13);
3. **ostream& shapka(ostream& stream)** – формирование «шапки» первой таблицы (стр. 13);
4. **ostream& shapka2(ostream& stream)** – формирование «шапки» второй таблицы (стр. 13).

# СХЕМА ВЗАИМОСВЯЗЕЙ ВСЕХ ФУНКЦИЙ



# ТЕКСТ ЗАГОЛОВОЧНОГО ФАЙЛА

# **Header.h**

**#include <conio.h>**

**#include <iostream>**

**#include <fstream>**

**#include <iomanip>**

**#include <string.h>**

**using namespace std;**

**extern int len;**

**extern char fname[20]; // переменная, имя файла**

**struct Product // структура, записывающая информацию для одного товара**

**{**

**char title[11]; // наименование товара (без пробелов)**

**char country[11]; // страна экспорта (без пробелов)**

**int volume; // объем поставляемой партии в штуках**

**double price; // цена единицы товара**

**char metka[2]; // метка**

**};**

**ostream& FinishL(ostream& stream);**

**ostream& FinishL2(ostream& stream);**

**ostream& shapka(ostream& stream);**

**ostream& shapka2(ostream& stream);**

**void write(char\* fname1, Product zap, int num = 0);**

**void print(Product zap, int i);**

**void print2(Product zap, int i, int volum\_p, double price\_p, double max\_p);**

**Product vvod();**

**void writeNzap(int n);**

**fstream& read(Product& zap, int num = 0);**

**void readAll();**

**void readN(int N);**

**int metkaDel(char\* shifr);**

**void UNmetkaDel();**

**void spiski();**

**int Delete();**

# ТЕКСТЫ ПРОГРАММНЫХ МОДУЛЕЙ

## **funcs.cpp:**

**#include "Header.h"**

**void write(char\* fname1, Product zap, int num)**

**{**

**fstream p(fname1, ios::app); // создание потока**

**if (p) // проверка файла на существование**

**{**

**p.close();**

**p.open(fname1, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**if (num)**

**{**

**p.seekg((num - 1) \* len, ios::beg);**

**}**

**p.write((char\*)&zap, len);**

**p.close();**

**}**

**void print(Product zap, int i)**

**{**

**cout << endl << setw(2) << i ;**

**cout << setw(12) << zap.title;**

**cout << setw(12) << zap.country;**

**cout << setw(12) << zap.volume;**

**cout << setw(17) << zap.price;**

**cout << setw(5) << zap.metka << setw(5) << '|';**

**}**

**void print2(Product zap, int i, int volum\_p, double price\_p, double max\_p)**

**{**

**cout << endl << setw(2) << i;**

**cout << setw(12) << zap.country;**

**cout << setw(12) << volum\_p;**

**cout << setw(12) << price\_p;**

**cout << setw(12) << zap.title;**

**cout << setw(12) << max\_p << setw(5) << '|';**

**}**

**Product vvod()**

**{**

**Product new\_zap;**

**cout << "\nВведите наименование товара:\n";**

**cin >> new\_zap.title;**

**cout << "Введите страну, куда товар экпортируется:\n";**

**cin >> new\_zap.country;**

**cout << "Введите число - объем поставляемой партии в штуках:\n";**

**cin >> new\_zap.volume;**

**cout << "Выберите число - цену за единицу товара:\n";**

**cin >> new\_zap.price;**

**strcpy\_s(new\_zap.metka, "+");**

**return new\_zap;**

**}**

**void writeNzap(int n)**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

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

**{**

**Product zap;**

**zap = vvod();**

**write(fname, zap);**

**}**

**p.close();**

**}**

**fstream& read(Product& zap, int num)**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**if (num)**

**{**

**p.seekg((num - 1) \* len, ios::beg);**

**p.read((char\*)&zap, len);**

**}**

**p.close();**

**return p;**

**}**

**void readAll()**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**p.seekg(0, ios::beg);**

**Product zap;**

**cout << shapka;**

**for (int i = 1; read(zap, i); i++)**

**print(zap, i);**

**cout << endl << FinishL;**

**p.close();**

**}**

**void readN(int N)**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**p.seekg(0, ios::beg);**

**Product zap;**

**if (read(zap, N))**

**{**

**cout << shapka;**

**print(zap, N);**

**cout << endl << FinishL;**

**}**

**else cout << endl << "Такой записи нет";**

**\_getch();**

**p.close();**

**}**

**int metkaDel(char\* title)**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**Product zap;**

**p.seekg(0, ios::beg);**

**int kol = 0;**

**for (int i = 1; read(zap, i); i++)**

**{**

**if (strcmp(zap.title, title) == 0)**

**{**

**strcpy\_s(zap.metka, "-");**

**write(fname, zap, i);**

**kol++;**

**}**

**}**

**p.close();**

**return kol;**

**}**

**void UNmetkaDel()**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**Product zap;**

**p.seekg(0, ios::beg);**

**for (int i = 1; read(zap, i); i++)**

**{**

**if (strcmp(zap.metka, "-") == 0)**

**{**

**strcpy\_s(zap.metka, "+");**

**write(fname, zap, i);**

**}**

**}**

**\_getch();**

**p.close();**

**}**

**int Delete()**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**char fname1[10] = "temp.bin";**

**fstream p1(fname1, ios::app);**

**if (p1)**

**{**

**p1.close();**

**p1.open(fname1, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**Product zap;**

**p.seekg(0, ios::beg);**

**int kol = 0;**

**for (int i = 1; read(zap, i); i++)**

**{**

**if (strcmp(zap.metka, "+") == 0)**

**{**

**write(fname1, zap);**

**kol++;**

**}**

**}**

**p.close();**

**p1.close();**

**remove(fname);**

**rename("temp.bin", fname);**

**return kol;**

**}**

**double max\_profit(char country[], bool f = false)**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**Product zap;**

**p.seekg(0, ios::beg);**

**// max\_price - максимальная прибыль,**

**// p\_price - прибыль от данного товара**

**double max\_price = 0, p\_price;**

**// цикл проходит по всем записям**

**for (int i = 1; read(zap, i); i++)**

**{**

**// прибыль для данного товара**

**p\_price = zap.volume \* zap.price;**

**// находим новую максимальную прибыль**

**if (p\_price > max\_price &&**

**(strcmp(zap.country, country) == 0 || f))**

**{**

**// запоминаем новую максимальную прибыль**

**max\_price = p\_price;**

**}**

**}**

**p.close();**

**return max\_price;**

**}**

**int all\_profit(char country[], double& price, int i)**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**Product zap;**

**p.seekg(0, ios::beg);**

**int sum = 0;**

**price = 0;**

**for (int j = i; read(zap, j); j++)**

**{**

**if (strcmp(zap.country, country) == 0) // проверка страны**

**{**

**sum += zap.volume;**

**price += zap.price \* zap.volume;**

**}**

**}**

**p.close();**

**return sum;**

**}**

**void find\_title(char country[], int& kol, int volume\_p,double price\_p, double max\_p, bool f = false)**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**Product zap;**

**p.seekg(0, ios::beg);**

**double p\_price;**

**for (int i = 1; read(zap, i); i++)**

**{**

**// проверка на страну**

**if (strcmp(zap.country, country) == 0 || f)**

**{**

**p\_price = zap.volume \* zap.price;**

**// проверка на максимальную прибыль**

**if (p\_price == max\_p)**

**{**

**print2(zap, kol, volume\_p, price\_p, max\_p);**

**kol++; // увеличение номера на единицу**

**}**

**}**

**}**

**p.close();**

**}**

**void spiski()**

**{**

**fstream p(fname, ios::app);**

**if (p)**

**{**

**p.close();**

**p.open(fname, ios::in | ios::out | ios::ate |**

**ios::binary);**

**}**

**Product zap;**

**cout << shapka2;**

**p.seekg(0, ios::beg);**

**double max\_p, price\_p, max\_p\_all;**

**int volume\_p, volume\_max, kol = 1;**

**char best\_title[11];**

**for (int i = 1; read(zap, i); i++)**

**{**

**// подсчёт общего объема начиная с первого запроса**

**volume\_max = all\_profit(zap.country, price\_p, 1);**

**// подсчёт общего объма начиная с i-го запроса**

**volume\_p = all\_profit(zap.country, price\_p, i);**

**// проверка того, что данный запрос первый, включающий в // себя данную страну, и что данная страна ещё не была // включена в список:**

**if (volume\_max == volume\_p)**

**{**

**max\_p = max\_profit(zap.country);**

**find\_title(zap.country, kol, volume\_p, price\_p,max\_p);**

**}**

**max\_p\_all = max\_profit(zap.country, true);**

**if (zap.volume \* zap.price == max\_p\_all)**

**strcpy\_s(best\_title, zap.title);**

**}**

**cout << endl << FinishL2 << endl;**

**cout << "Товар, который приносит наибольший доход в казну государства, это: " << best\_title << ", приносит прибыль в размере: " << max\_p\_all;**

**p.close();**

**}**

## **main.cpp:**

**#include "Header.h"**

**// глобальные переменные**

**char fname[20];**

**int len = sizeof(Product);**

**char menu()**

**{**

**setlocale(LC\_ALL, "Russian");**

**cout << endl << "\n\t1-> Добавить запись"**

**<< endl << "\t2-> Чтение всех записей"**

**<< endl << "\t3-> Чтение записи с N номером"**

**<< endl << "\t4-> Пометить на удаление"**

**<< endl << "\t5-> Снять пометки на удаление"**

**<< endl << "\t6-> Удалить выбранную запись"**

**<< endl << "\t7-> Список стран"**

**<< endl << "\t0-> EXIT" << endl;**

**char s;**

**cout << "\n\t-> ";**

**cin >> s;**

**return s;**

**}**

**void main()**

**{**

**setlocale(LC\_ALL, "Russian"); // вывод букв кириллицы**

**cout << "Введите имя файла:\n";**

**cin >> fname;**

**int n;**

**do**

**{**

**char s = menu();**

**switch (s)**

**{**

**case '1':**

**cout << "Введите кол-во записей: ";**

**cin >> n;**

**writeNzap(n);**

**break;**

**case '2':**

**readAll();**

**\_getch();**

**break;**

**case '3':**

**int nom;**

**cout << "\nВведите номер записи: ";**

**cin >> nom;**

**readN(nom);**

**\_getch();**

**break;**

**case '4':**

**int a;**

**char name[10];**

**cout << "\n\tВведите наименование товара: ";**

**cin >> name;**

**a = metkaDel(name);**

**break;**

**case '5':**

**UNmetkaDel();**

**system("cls");**

**break;**

**case '6':**

**Delete();**

**cout << ("\n\tDone!");**

**break;**

**case '7':**

**spiski();**

**break;**

**case '0':**

**exit(0);**

**}**

**} while (1);**

**system("pause");**

**}**

## **tabl.cpp:**

**#include "Header.h"**

**#include <iomanip>**

**using namespace std;**

**ostream& FinishL(ostream& stream);**

**ostream& FinishL2(ostream& stream);**

**ostream& shapka(ostream& stream);**

**ostream& shapka2(ostream& stream);**

**ostream& shapka(ostream& stream)**

**{**

**setlocale(LC\_ALL, "Russian");**

**stream << endl << FinishL << endl;**

**stream << setw(2) << '№';**

**stream << setw(12) << "товар";**

**stream << setw(12) << "страна";**

**stream << setw(12) << "объём";**

**stream << setw(17) << "цена за единицу";**

**stream << setw(7) << "Метка" << setw(3) << '|' << endl;**

**stream << setw(65) << '|' << endl << FinishL;**

**return stream;**

**}**

**ostream& FinishL(ostream& stream)**

**{**

**stream << setfill('-') << setw(65) << '|' << setfill(' ');**

**return stream;**

**}**

**ostream& FinishL2(ostream& stream)**

**{**

**stream << setfill('-') << setw(67) << '|' << setfill(' ');**

**return stream;**

**}**

**ostream& shapka2(ostream& stream)**

**{**

**setlocale(LC\_ALL, "Russian");**

**stream << endl << FinishL2 << endl;**

**stream << setw(2) << '№';**

**stream << setw(12) << "страна";**

**stream << setw(14) << "общий объём";**

**stream << setw(12) << "общая цена";**

**stream << setw(10) << "лучший";**

**stream << setw(14) << "прибыль";**

**stream << setw(3) << '|' << endl;**

**stream << setw(50) << "товар";**

**stream << setw(14) << "с товара" << setw(3**

**) << '|';**

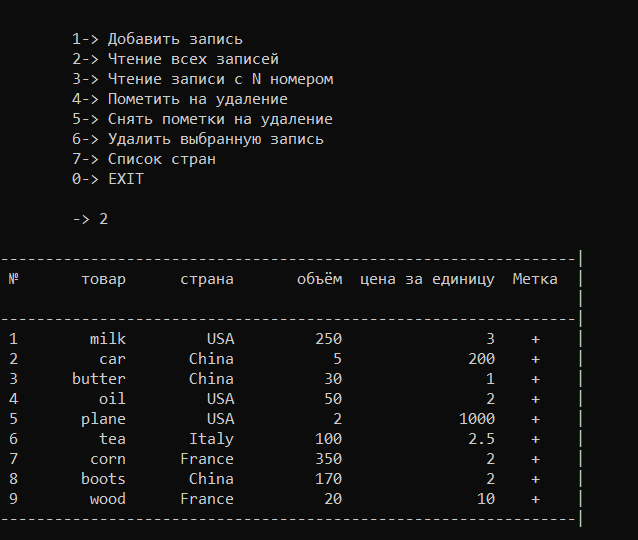
**stream << endl << FinishL2;**

**return stream;**

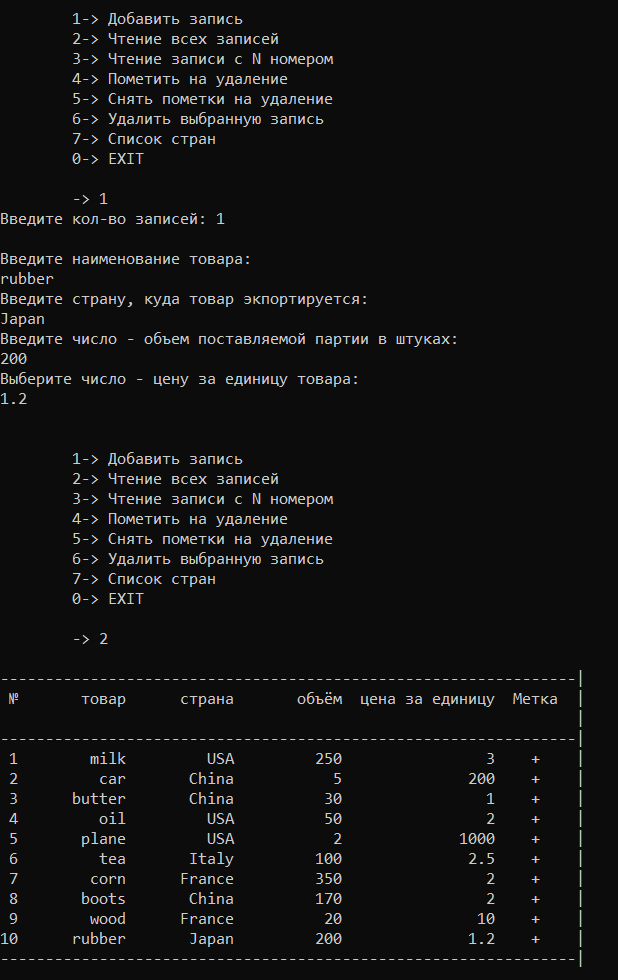
**}**

# РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

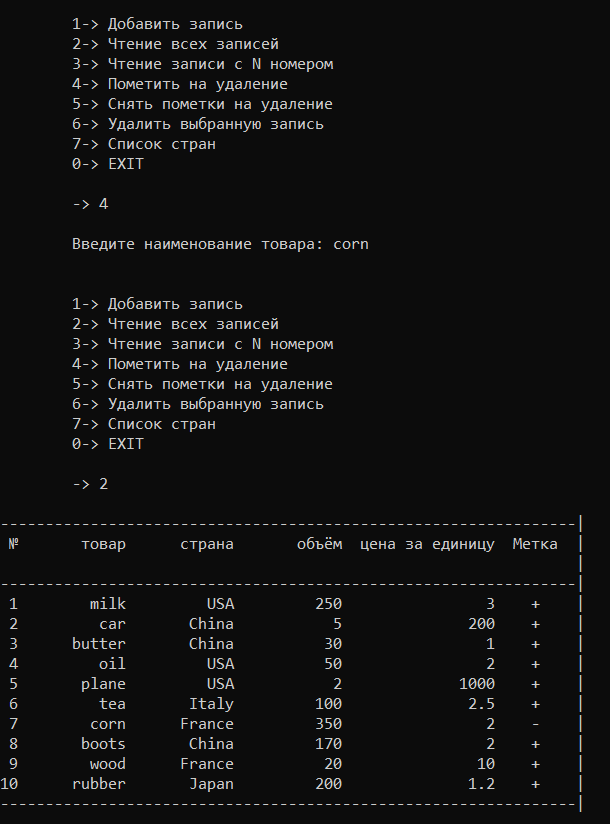
## **Чтение всех записей**



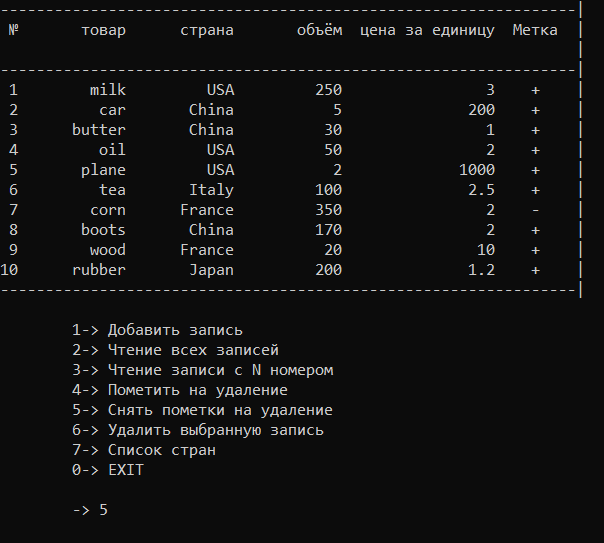
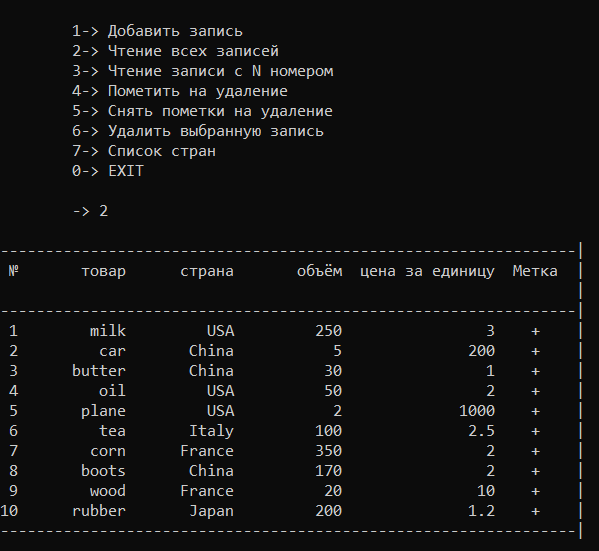
## **Добавить запись**

****

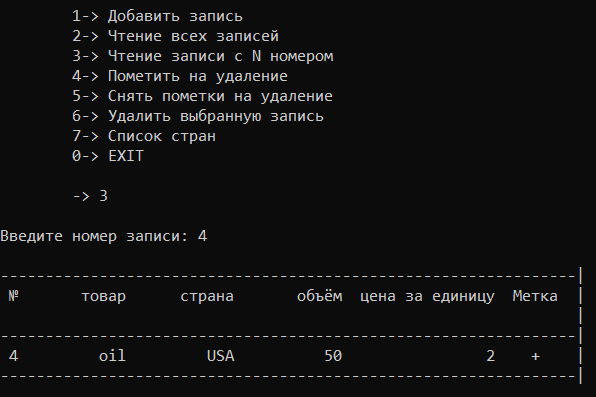
## **Пометить на удаление**

****

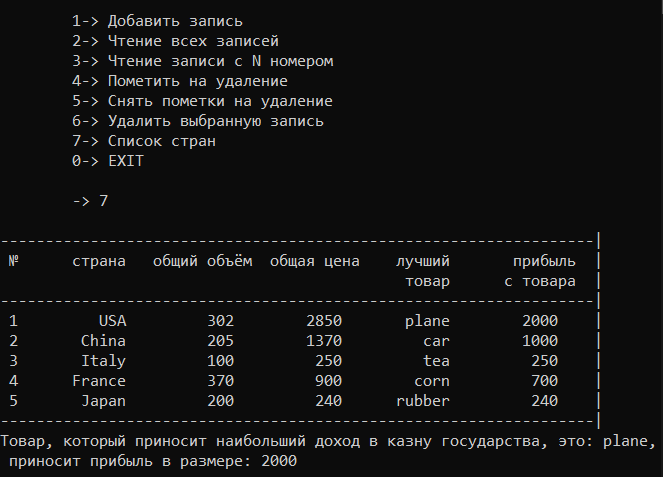
## **Снять пометки на удаление**

****

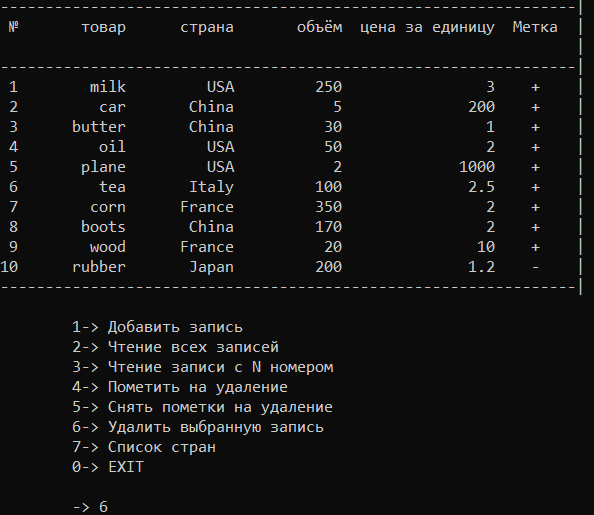
## **Чтение записи с N номером**

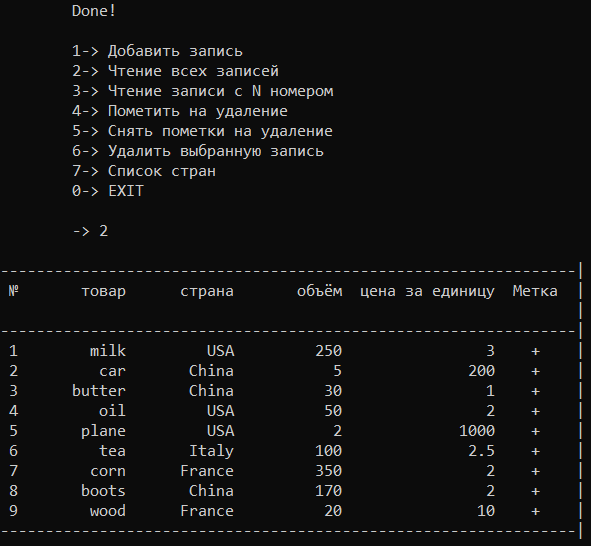
****

## **Список стран**

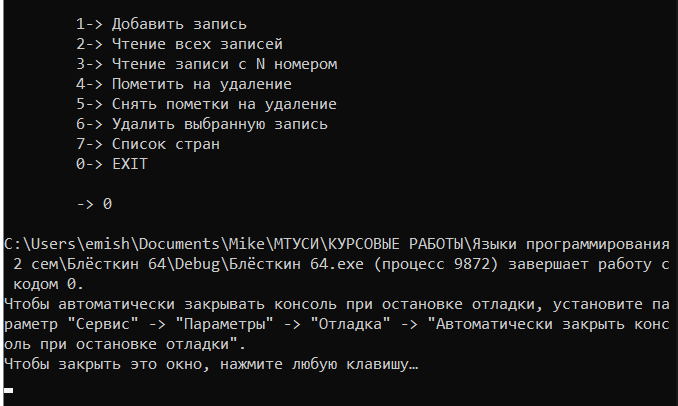
****

## **Удалить выбранную запись**

****

****

## **Закрытие программы**

****

# ВЫВОД

В результате выполнения курсовой работы по дисциплине «Языки программирования» поставленная задача успешно выполнена. Были достигнуты следующие результаты:

1. Разработан алгоритм для решения поставленной задачи;
2. Составлена пояснительная записка, в которой кратко описаны разработанные функции и их значение в общем алгоритме;
3. Составлена схема взаимодействия всех функций в программе;
4. Создана полностью рабочая и отлаженная программа в среде Microsoft Visual Studio на языке программирования C++, реализующая разработанный алгоритм и функции, которая выполнена в виде проекта, имеющий четыре файла:
   * **main.cpp** – главный файл,
   * **funcs.cpp** – файл, содержащий вычислительные функции**,**
   * **tabl.cpp** – файл, строящий элементы таблицы записей,
   * **Header.h** – заголовочный файл;
5. Реализована возможность работы (создание, чтение и запись) с двоичными файлами расширения типа «.bin»;
6. Созданная программа проверена на тестовых данных;
7. Продемонстрирован результат работы программы, а именно, продемонстрирована возможность:
   * добавлять новую запись в конец файла,
   * читать и выводить на экран все имеющиеся записи в файле в виде таблицы, а также выводить любую выбранную запись,
   * помечать выбранные записи на удаления,
   * убирать пометки на удаления с записей,
   * удалять помеченные записи из файла,
   * выводить на экран таблицу стран и прибыли с экспортируемых в них товаров,
   * выводить на экран самый прибыльный товар и общую прибыль с него от экспорта,
   * завершать программу, сохраняя данные в используемом для работы файле;
8. Проведён анализ результатов выполнения курсовой работы и составлен перечень всех результатов;
9. Составлен и распечатан отчёт о курсовой работе;
10. Отчёт сдан на проверку руководителю;
11. Курсовая работа полностью проверена и оценена опытным руководителем.