МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения информационных систем и технологий

**КУРСОВАЯ РАБОТА**

по дисциплине: ”Основы алгоритмизации и программирования”

на тему: ***”*** ***Разработка программы для информационной системы ассортимента обуви в магазине”***

Выполнил**:** ст. гр. 10701320 Ковалёв Н. Д.

Приняла**:** ст**.** преподаватель И. М. Борисова

Минск 2021

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра программного обеспечения информационных систем и технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе**

**по дисциплине**

**«Основы алгоритмизации**

**и программирования»**

Тема: Разработка программы для информационной системы ассортимента обуви в магазине

**Исполнитель**: Ковалёв Н. Д.

(подпись)

**Студент 1 курса 10701320 группы**

**Руководитель**: Борисова И. М.

(подпись)

Минск 2021

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 6](#_Toc71133898)

[1. Техническое задание 7](#_Toc71133899)

[1.1. Постановка задачи 7](#_Toc71133900)

[1.2. Исходные данные 7](#_Toc71133901)

[2. Проектирование структуры программы 8](#_Toc71133902)

[2.1. Разработка модульной структуры программы 8](#_Toc71133903)

[2.2. Выбор способа организации данных 9](#_Toc71133904)

[2.3. Разработка перечня пользовательских функций 10](#_Toc71133905)

[3. Разработка алгоритмов работы программы 11](#_Toc71133906)

[3.1. Алгоритм функции main 11](#_Toc71133907)

[3.2. Алгоритм функции push\_back 12](#_Toc71133908)

[3.3. Алгоритм функции pop\_back 13](#_Toc71133909)

[4. Описание работы программы 14](#_Toc71133912)

[ЗАКЛЮЧЕНИЕ 20](#_Toc71133913)

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 20

ПРИЛОЖЕНИЕ 21

**ВВЕДЕНИЕ**

Цель курсовой работы - закрепление и углубление знаний, полученных при изучении курса «Основы алгоритмизации и программирования» посредством разработки программного обеспечения для компьютера. Для этого необходимо разработать программу ассортимента обуви в магазине с возможностью добавления и удаления обуви, сортировки и поиска, а также просмотра информации о всей обуви.

**1. Техническое задание**

**1.1 Постановка задачи**

Разработка программы **ассортимента обуви в магазине.** Сведения об ассортименте обуви в магазине включают: артикул, наименование, цена, изготовитель, Бренд, размер. Артикул начинается с буквы Ж для женской обуви, М – для мужской, Д – для детской. Индивидуальное задание: вывести список обуви артикула x и размера y (x, y вводятся с клавиатуры). Общее для всех вариантов задание: реализовать авторизацию для входа в систему, функционал администратора и функционал пользователя (см. более подробно в функциональных требованиях к курсовой работе).

**1.2 Исходные данные**

* Язык программирования С++;
* Среда разработки Microsoft Visual Studio;
* Вид приложения – консольное;
* Парадигма программирования – процедурная;
* Способ организации данных – структуры (struct);
* Способ хранения данных – файлы;
* Каждая логически завершенная подзадача программы должна быть реализована в виде отдельной функции;
* Допускается использовать классы string и vector, библиотеку algorithm.

Необходимо предусмотреть:

* Возможность возврата назад (навигация);
* Запрос на выполнение необратимых действий;
* Обратную связь с пользователем, например, вывод сообщения об успешности удаления/редактирования записи и т.д. [1]

**2. Проектирование структуры программы**

**2.1 Разработка модульной структуры программы**

На рисунке 2.1 показано графическое представление структуры программы.

**РАБОТА С ДАННЫМИ**

**РЕЖИМ РЕДАКТИРОВАНИЯ**

**ПРОСМОТР**

**ДОБАВЛЕНИЕ**

**УДАЛЕНИЕ**

**РЕЖИМ ОБРАБОТКИ**

**ИНД. ЗАДАНИЕ**

**ПОИСК**

**СОРТИРОВКА**

Рисунок 2.1 – Структура программы

**2.2 Разработка способа организации данных**

enum Brand

{

BRAND\_ADIDAS,

BRAND\_NIKE,

BRAND\_ECCO,

BRAND\_BARKER,

BRAND\_CROCS,

BRAND\_COLUMBIA,

BRAND\_CONVERSE,

BRAND\_LACOSTE,

BRAND\_VANS,

BRAND\_PORRONET,

};

struct Shoe

{

string article\_number;

string nameOfShue;

float priseOfShue;

int sizeOfShue;

Brand brand\_name;

};

**2.3 Разработка перечня пользовательских функций программы**

void readFile(Shoes\*, int); Функция считывает записи из файла.

void saveShoes(Shoes\*, int); Функция сохраняет данные в файл.

void show\_list(Shoes\*, int); Функция выводит всю обувь на экран.

void showRecording(Shoes\*); Функция выводит определенную запись структуры на экран.

void push\_back(int&, Shoes\*&); Функция добавляет запись в массив.

void pop\_back(int&, Shoes\*&); Функция удаляет запись из массива.

void individual\_task(Shoes\*, int); Функция выводит обувь артикула x и размера y.

void sorting(Shoes\*, int); Некое меню сортировки обуви по различным параметрам.

void search\_Shoe(Shoes\*, int); Функция выводит меню поиска обуви по 3 критериям (размер, цена, название бренда).

**3. Разработка алгоритмов работы программы**

**3.1 Алгоритм функции main**

В функции main() после инициализации данных (считывания из файла) вызывается главное меню, в котором реализован функционал пользователя и администратора, а также их обратное взаимодействие, по окончанию работы которого, данные сохраняются, и программа завершает свою работу. Алгоритм приводится на рисунке 3.1.

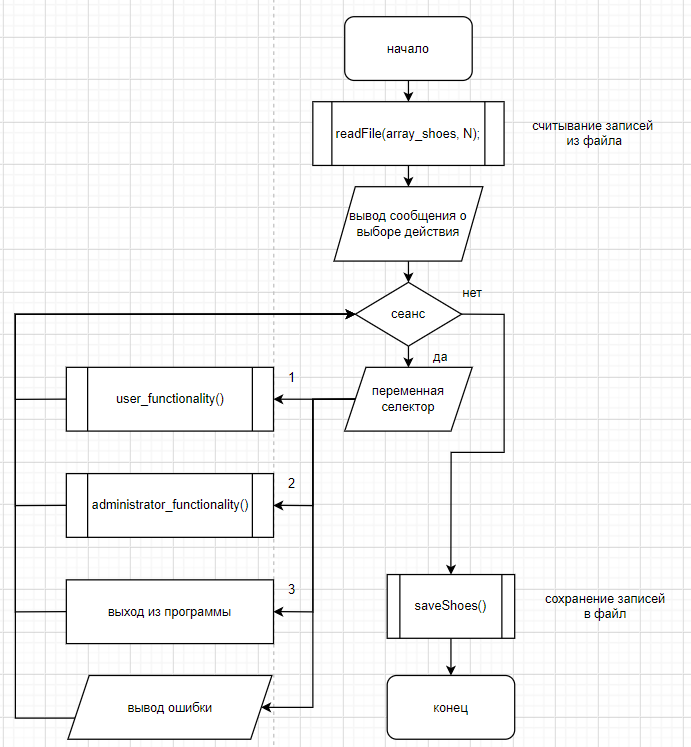


Рисунок 3.1 – Блок-схема функции main

**3.2 Алгоритм функции push\_back**

Функция push\_back() добавляет обувь в динамический массив структур.

Алгоритм приводится на рисунке 3.2.

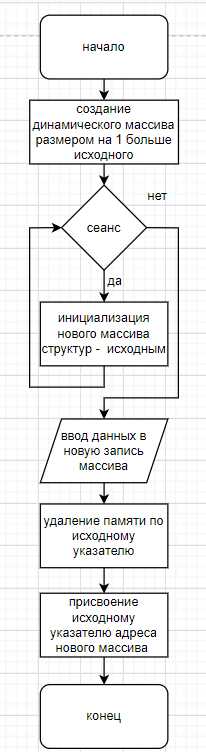


Рисунок 3.2 – Блок-схема функции push\_back

**3.3 Алгоритм функции pop\_back**

Функция pop\_back() удаляет обувь из динамического массива структур.

Алгоритм приводится на рисунке 3.3.



Рисунок 3.3 – Блок-схема функции pop\_back

1. **Описание работы программы**

Начало работы программы, считывание данных из файла и вызов меню, в котором доступны 2 режима (пользователь и администратор) приводится на рисунке 4.1.

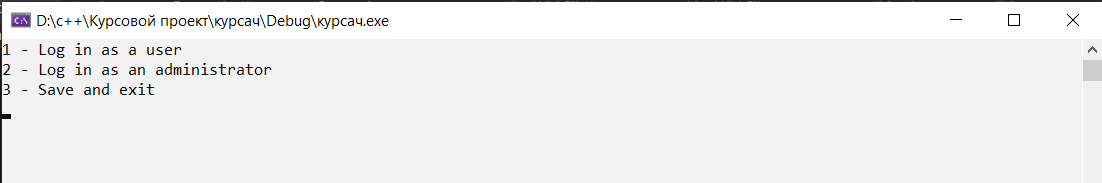


Рисунок 4.1

Меню пользователя, в котором можно просмотреть всю обувь, отсортировать обувь по критериям (2 параметра), осуществить поиск обуви (3 параметра), просмотреть обувь артикула x и размера y (x и y вводятся с клавиатуры) приводится на рисунке 4.2.

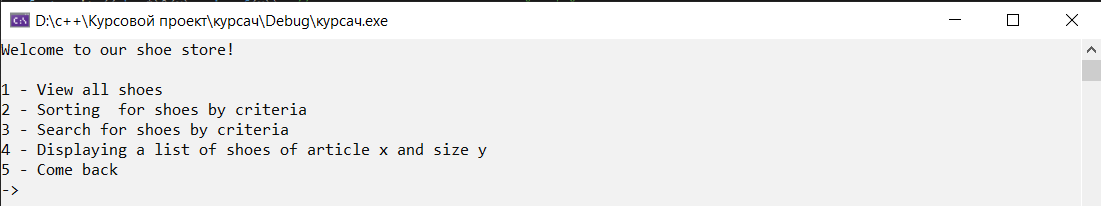


Рисунок 4.2

Меню администратора, в котором можно просмотреть всю обувь, добавить обувь, удалить обувь, а также сменить пароль, который нужно ввести при входе от лица администратора, приводится на рисунке 4.3.

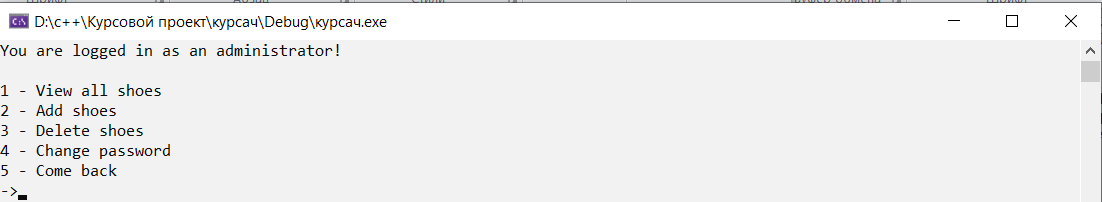


Рисунок 4.3

Вызов функции для просмотра всей обуви приводится на рисунке 4.4.

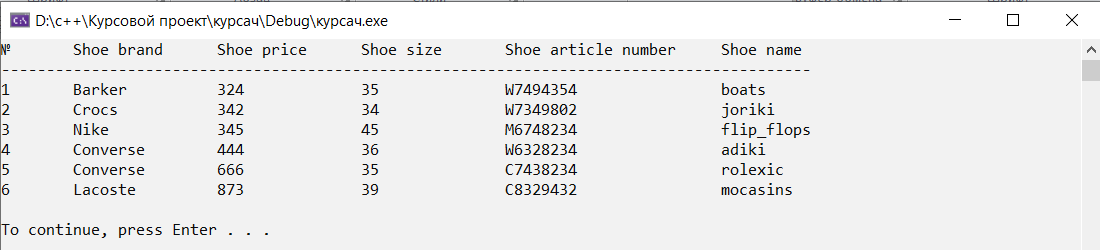


Рисунок 4.4

Меню функции сортировки обуви приводится на рисунке 4.5.

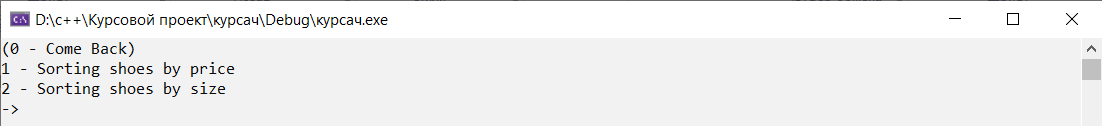


Рисунок 4.5

Вызов функции сортировки обуви по цене приводится на рисунке 4.6.

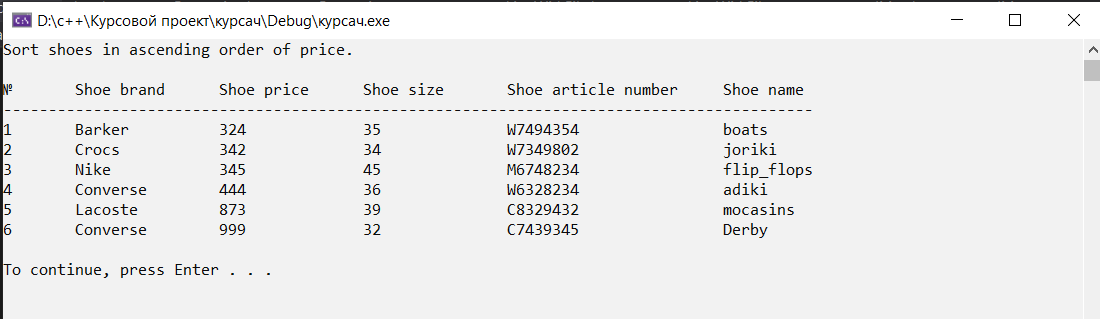


Рисунок 4.6

Вызов функции сортировки обуви по размеру приводится на рисунке 4.7.

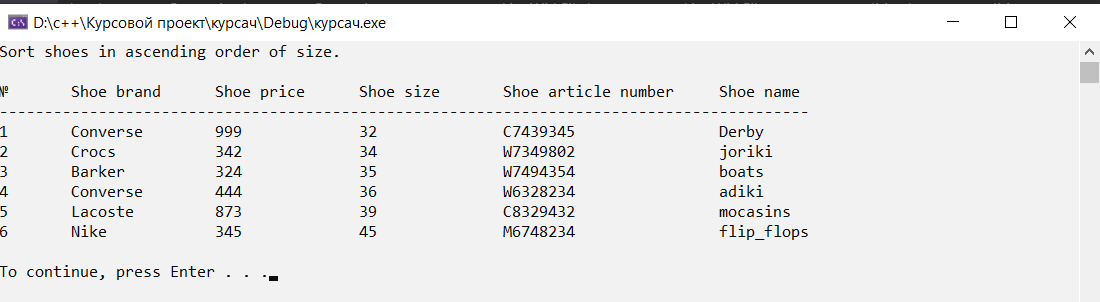


Рисунок 4.7

Меню функции поиска обуви приводится на рисунке 4.8.

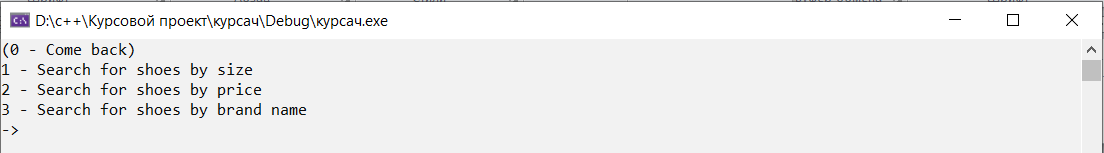


Рисунок 4.8

Вызов функции поиска обуви по размеру приводится на рисунках 4.9. и 4.10.



Рисунок 4.9

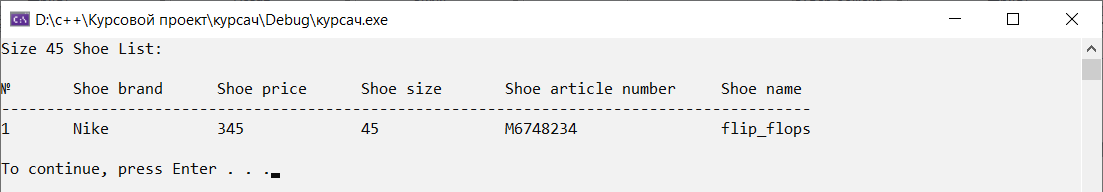


Рисунок 4.10

Вызов функции поиска обуви по цене приводится на рисунках 4.11. и 4.12.

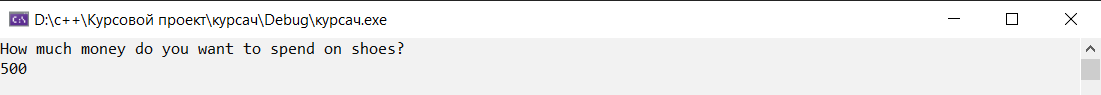


Рисунок 4.11

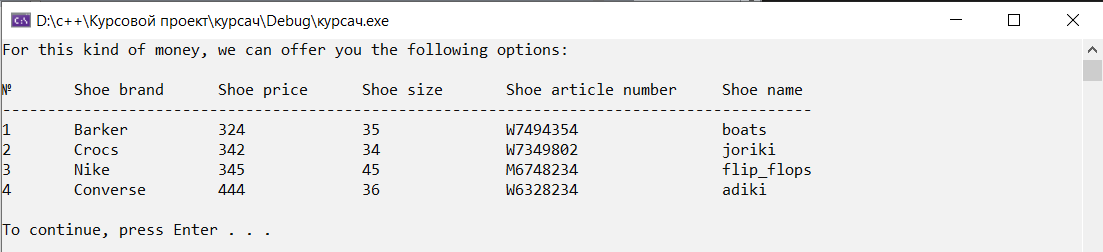


Рисунок 4.12

Вызов функции поиска обуви по названию бренда приводится на рисунках 4.13. и 4.14.

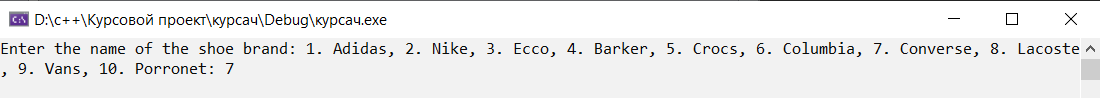


Рисунок 4.13

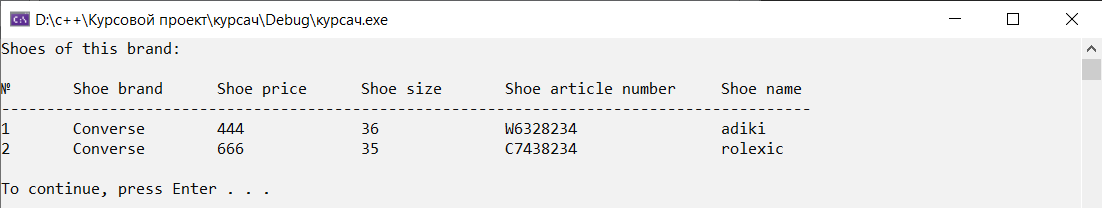


Рисунок 4.14

Вызов функции, выводящей всю обувь артикула x и размера y (x, y – вводятся с клавиатуры) приводится на рисунках 4.15., 4.16.

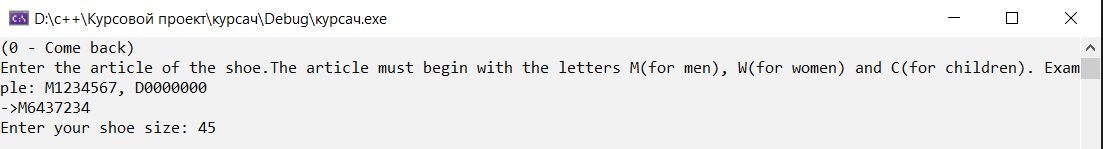


Рисунок 4.15

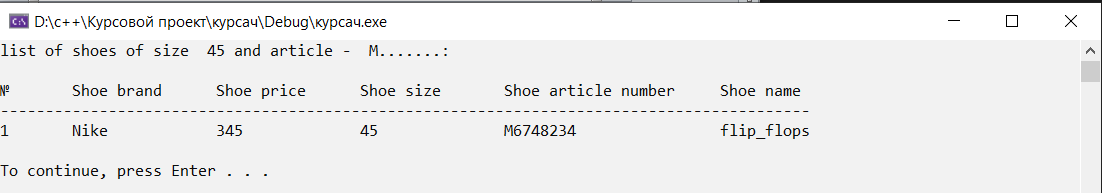


Рисунок 4.16

Вызов функции добавления приводится на рисунках 4.17. и 4.18.

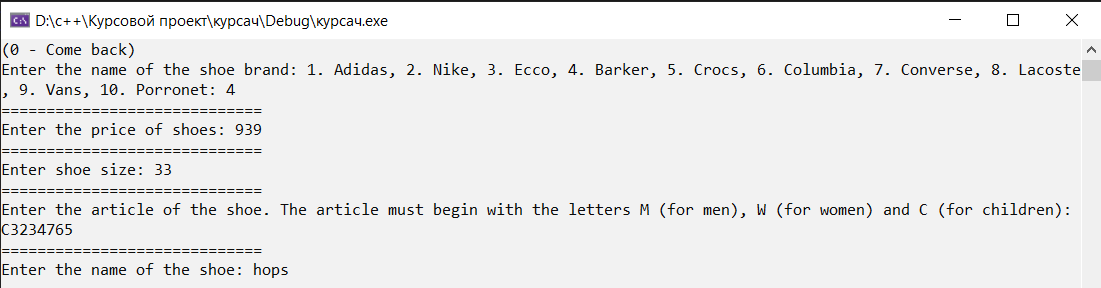


Рисунок 4.17

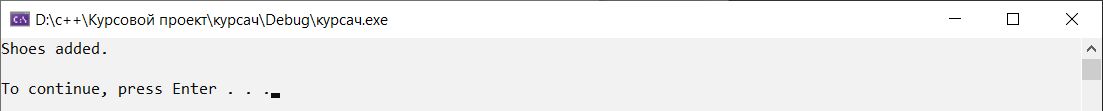


Рисунок 4.18

Вызов функции удаления обуви приводится на рисунках 4.19. и 4.20.

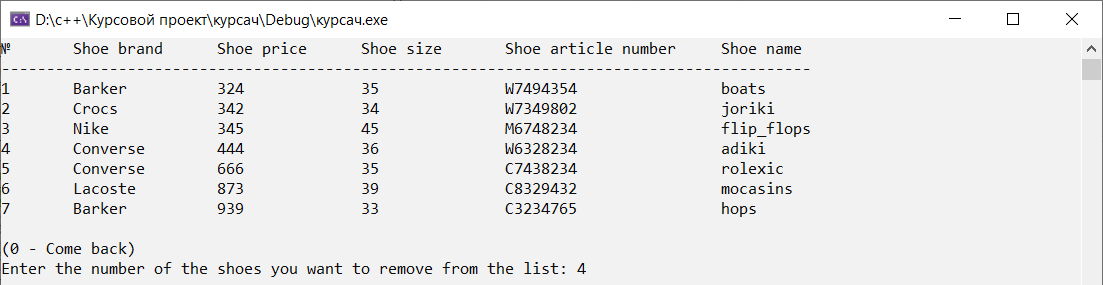


Рисунок 4.19

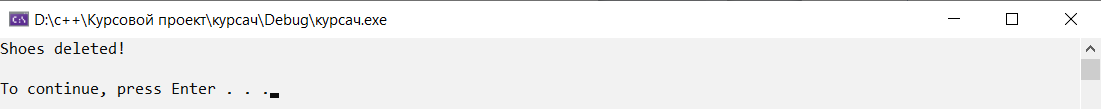


Рисунок 4.20

Вызов функции смены пароля приводится на рисунках 4.21., 4.22.

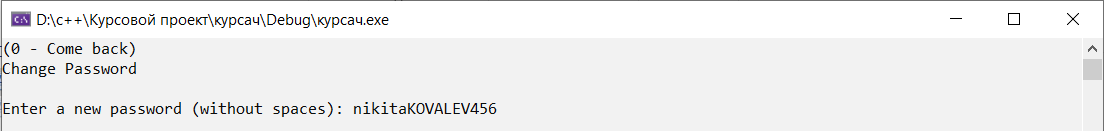


Рисунок 4.21

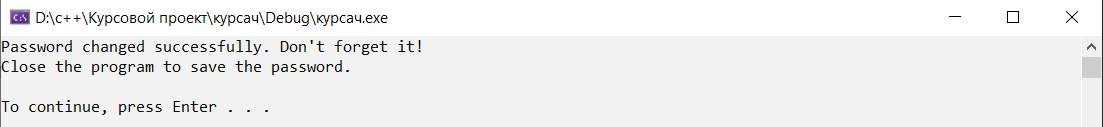


Рисунок 4.22

Выход из программы, сохранение данных в файл и завершение работы программы приводится на рисунке 4.23.

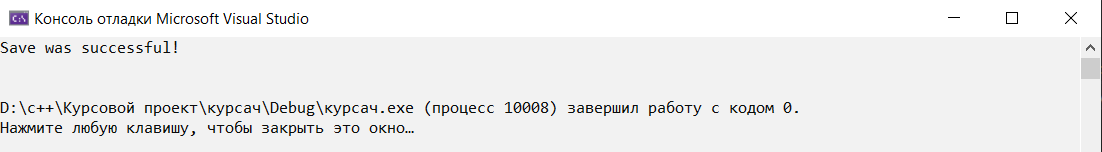


Рисунок 4.23

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы была разработана программа ассортимента обуви в магазине. В ней были реализованы возможность добавления и удаления обуви, сортировки и поиска, а также просмотр всей информации об обуви

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Бондарев В.М. Программирование на С++ / 2-е издание. – Харьков: Компания «Смит», 2005. – 284 с.

2 Борисова И.М. – Минск: БНТУ, 2021 / Методические указания и задания для выполнения курсовой работы по дисциплине: «Основы алгоритмизации и программирования»

3 Васильев А.Н. – Москва, 2019 / Программирование на C++ в примерах и задачах

4 Довбуш Г.Ф. Visual C++ на примерах. / Г.Ф.Довбуш; под ред. проф.А.Д.Хомоненко: - СПб.: БХВ-Петербург, 2007. - 528 с.

5 Подбельский В.В. Язык Си++. Учебное пособие / В.В. Подбельский. - М.: Финансы и статистика, 2003. – 560 с.

6 Шиманович Е.Л. С/С++ в примерах и задачах / Е.Л.Шиманович – Минск: Новое знание, 2004. – 528 с.

**ПРИЛОЖЕНИЕ Листинг исходного кода**

**Файл shoe.h**

#pragma once

#include <string>

using namespace std;

enum Brand

{

BRAND\_ADIDAS = 1,

BRAND\_NIKE = 2,

BRAND\_ECCO = 3,

BRAND\_BARKER = 4,

BRAND\_CROCS = 5,

BRAND\_COLUMBIA = 6,

BRAND\_CONVERSE = 7,

BRAND\_LACOSTE = 8,

BRAND\_VANS = 9,

BRAND\_PORRONET = 10,

};

struct Shoes //объявляем структуру, содержащую следующие поля:

{

string article\_number; //артикул

string nameOfShue; //название обуви

float priseOfShue; // поиск, сортировка //цена обуви

int sizeOfShue; // поиск, сортировка //размер обуви

Brand brand\_name; // поиск //название бренда обуви.

};

**Файл Processing.h**

#pragma once

#include "shoe.h"; // подключаем заголовочный файл shoe.h, потому что там хранится наша структура, которую мы передаем в функции.

#include "editing.h"; // подключаем заголовочный файл shoe.h, потому что там хранятся функции, которые испльзуются в соответствующем cpp файле

void individual\_task(Shoes\*, int); //функция выводит обувь артикула x и размера y.

void sorting(Shoes\*, int); // некое меню сортировки обуви по различным параметрам

int minimum\_fromPrice(Shoes\*, int, int); // возвращает индекс структуры с минимальной ценой обуви

void selectionSortPrice(Shoes\*, int); // сортировка выбором по цене обуви

int minimum\_fromSize(Shoes\*, int, int); // возвращает индекс структуры с минимальным размером обуви

void selectionSortSize(Shoes\*, int); // сортировка выбором по размеру обуви

void search\_Shoe(Shoes\*, int); // функция выводит меню поиска обуви по 3 критериям (размер, цена, название бренда).

void search\_shoeSize(Shoes\*, int); //функция выводит обувь размера, введеннго с клавиатуры.

void search\_shoePrise(Shoes\*, int); //функция выводит всю обувь ценой ниже, чем ввведенная с клавиатуры.

void search\_shoeBrand\_name(Shoes\*, int); //функция выводит обувь бренда, введенного с клавиатуры.

**Файл Processing.cpp**

#include "Processing.h";

void individual\_task(Shoes\* array\_shoes, int N)

{

system("CLS");

string articl;

int size, kol = 0;

if (!isEmpty(N))

{

cout << "(0 - Come back)\nEnter the article of the shoe.The article must begin with the letters M(for men), W(for women) and C(for children). Example: M1234567, D0000000\n->";

while (true)

{

cin >> articl;

if (articl[0] == '0') { system("CLS"); return; }

if (articl[0] == 'M' or articl[0] == 'W' or articl[0] == 'C') // если артикул начинается с M, W, C идем дальше, иначе нужно ввести артикул заново.

break;

else system("CLS"); cout << "(0 - Come back)\nThe article must start with M, W or C! Try entering the article number again.\n->";

}

while (true) // проверка на принадлежноть вводимого типа с типом переменной

{

cout << "Enter your shoe size: ";

cin >> size;

if (size == 0) { system("CLS"); return; }

if (Test(size)) break;

else { system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; }

}

system("CLS");

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

{

if (array\_shoes->article\_number[0] == articl[0] and array\_shoes->sizeOfShue == size)

{

kol++;

if (kol == 1)

{

cout << "list of shoes of size " << size << " and article - " << articl[0] << "......." << ":" << endl << endl;

cout << "№\tShoe brand\tShoe price\tShoe size\tShoe article number\tShoe name" << endl;

cout << "------------------------------------------------------------------------------------------" << endl;

}

cout << kol << "\t"; showRecording(array\_shoes);

}

array\_shoes++;

}

if (kol == 0)

cout << "Shoes article " << articl << " and size " << size << " - no." << endl;

cout << endl;

}

else

cout << "In the array no records." << endl << endl; pause(); system("CLS");

}

int minimum\_fromPrice(Shoes\* array\_shoes, int position, int N)

{

int min\_index = position;

for (int count = position + 1; count < N; count++)

{

if ((array\_shoes + count)->priseOfShue < (array\_shoes + min\_index)->priseOfShue) min\_index = count;

}

return min\_index;

}

void selectionSortPrice(Shoes\* array\_shoes, int N)

{

for (int count = 0; count < N - 1; count++)

{

swap(\*(array\_shoes + count), \*(array\_shoes + minimum\_fromPrice(array\_shoes, count, N)));

}

}

int minimum\_fromSize(Shoes\* array\_shoes, int position, int N)

{

int min\_index = position;

for (int count = position + 1; count < N; count++)

{

if ((array\_shoes + count)->sizeOfShue < (array\_shoes + min\_index)->sizeOfShue) min\_index = count;

}

return min\_index;

}

void selectionSortSize(Shoes\* array\_shoes, int N)

{

for (int count = 0; count < N - 1; count++)

{

swap(\*(array\_shoes + count), \*(array\_shoes + minimum\_fromSize(array\_shoes, count, N)));

}

}

void sorting(Shoes\* array\_shoes, int N)

{

system("CLS");

if (!isEmpty(N))

{

int answer;

while (true)

{

while (true) // проверка на принадлежноть вводимого типа с типом переменной

{

cout << "(0 - Come Back)\n1 - Sorting shoes by price\n2 - Sorting shoes by size\n->";

cin >> answer;

if (answer == 0) { system("CLS"); return; }

if (Test(answer)) break;

else { system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; }

}

if (answer == 1) {

selectionSortPrice(array\_shoes, N);

system("CLS");

cout << "Sort shoes in ascending order of price." << endl << endl;

show\_list(array\_shoes, N);

pause(); system("CLS");

break;

}

else if (answer == 2)

{

selectionSortSize(array\_shoes, N);

system("CLS");

cout << "Sort shoes in ascending order of size." << endl << endl;

show\_list(array\_shoes, N);

pause(); system("CLS");

break;

}

else system("CLS"); cout << "You entered incorrect data! Try again:" << endl << endl;

}

}

else

cout << "In the array no records." << endl << endl;

}

void search\_Shoe(Shoes\* array\_shoes, int N)

{

system("CLS");

if (!isEmpty(N))

{

bool flag = true;

int answer;

while (flag)

{

while (true) // проверка на принадлежноть вводимого типа с типом переменной

{

cout << "(0 - Come back)\n1 - Search for shoes by size\n2 - Search for shoes by price\n3 - Search for shoes by brand name\n->";

cin >> answer;

if (answer == 0) { system("CLS"); return; }

if (Test(answer)) break;

else { system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; }

}

switch (answer)

{

case 1: search\_shoeSize(array\_shoes, N); flag = false;

break;

case 2: search\_shoePrise(array\_shoes, N); flag = false;

break;

case 3: search\_shoeBrand\_name(array\_shoes, N); flag = false;

break;

default: system("CLS"); cout << "You entered incorrect data! Try again:" << endl << endl;

break;

}

}

}

else

cout << "In the array no records." << endl << endl; pause(); system("CLS");

}

void search\_shoeSize(Shoes\* array\_shoes, int N)

{

int size, kol = 0;

system("CLS");

cout << "Enter the shoe size you need: ";

while (true) // проверка на принадлежноть вводимого типа с типом переменной

{

cin >> size;

if (Test(size)) break;

else { system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; }

}

system("CLS");

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

{

if (array\_shoes->sizeOfShue == size)

{

kol++;

if (kol == 1)

{

cout << "Size " << size << " Shoe List:" << endl << endl;

cout << "№\tShoe brand\tShoe price\tShoe size\tShoe article number\tShoe name" << endl;

cout << "------------------------------------------------------------------------------------------" << endl;

}

cout << kol << "\t";

showRecording(array\_shoes);

}

array\_shoes++;

}

if (kol == 0)

cout << "There are no shoes of this size!" << endl << endl;

cout << endl;

}

void search\_shoePrise(Shoes\* array\_shoes, int N)

{

int geld, kol = 0;

system("CLS");

cout << "How much money do you want to spend on shoes? " << endl;

while (true) // проверка на принадлежноть вводимого типа с типом переменной

{

cin >> geld;

if (Test(geld)) break;

else { system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; }

}

system("CLS");

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

{

if (array\_shoes->priseOfShue <= geld)

{

kol++;

if (kol == 1)

{

cout << "For this kind of money, we can offer you the following options:" << endl << endl;

cout << "№\tShoe brand\tShoe price\tShoe size\tShoe article number\tShoe name" << endl;

cout << "------------------------------------------------------------------------------------------" << endl;

}

cout << kol << "\t";

showRecording(array\_shoes);

}

array\_shoes++;

}

if (kol == 0)

cout << "Unfortunately, we don't have shoes for that kind of money." << endl << endl;

cout << endl;

}

void search\_shoeBrand\_name(Shoes\* array\_shoes, int N)

{

int brandd, kol = 0;

system("CLS");

while (true)

{

while (true) // проверка на принадлежноть вводимого типа с типом переменной

{

cout << "Enter the name of the shoe brand: 1. Adidas, 2. Nike, 3. Ecco, 4. Barker, 5. Crocs, 6. Columbia, 7. Converse, 8. Lacoste, 9. Vans, 10. Porronet: ";

cin >> brandd;

if (Test(brandd)) break;

else { system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; }

}

if (brandd > 0 and brandd <= 10)

break;

else

system("CLS"); cout << "You can only enter a number from 1 to 10. Try again!" << endl << endl;

}

system("CLS");

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

{

if (array\_shoes->brand\_name == brandd)

{

kol++;

if (kol == 1)

{

cout << "Shoes of this brand:" << endl << endl;

cout << "№\tShoe brand\tShoe price\tShoe size\tShoe article number\tShoe name" << endl;

cout << "------------------------------------------------------------------------------------------" << endl;

}

cout << kol << "\t";

showRecording(array\_shoes);

}

array\_shoes++;

}

if (kol == 0)

cout << "There are no shoes of this brand!" << endl << endl;

cout << endl;

}

**Файл editing.h**

#pragma once

#include <iostream>

#include "shoe.h"

void show\_list(Shoes\*, int); //функция выводит всю обувь на экран.

void showRecording(Shoes\*); //функция выводит определенную запись структуры на экран.

void push\_back(int&, Shoes\*&); //функция добавляет запись в массив структур.

void pop\_back(int&, Shoes\*&); //функция удаляет запись из массива структур.

void pause(); // функция ждет, пока мы не нажмем enter.

bool isEmpty(int); // проверяет, пуст ли массив

bool Test(int); // проерка на принадлежноть вводимого типа с типом переменной

**Файл editing.cpp**

#include "editing.h";

void show\_list(Shoes\* array\_shoes, int N)

{

if (!isEmpty(N))

{

cout << "№\tShoe brand\tShoe price\tShoe size\tShoe article number\tShoe name" << endl;

cout << "------------------------------------------------------------------------------------------" << endl;

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

{

cout << i + 1 << "\t";

showRecording(array\_shoes);

array\_shoes++;

}

cout << endl;

}

else

cout << "No shoes in stock!" << endl;

}

void showRecording(Shoes\* array\_shoes)

{

switch (array\_shoes->brand\_name)

{

case 1: cout << "Adidas\t\t"; break;

case 2: cout << "Nike\t\t"; break;

case 3: cout << "Ecco\t\t"; break;

case 4: cout << "Barker\t\t"; break;

case 5: cout << "Crocs\t\t"; break;

case 6: cout << "Columbia\t"; break;

case 7: cout << "Converse\t"; break;

case 8: cout << "Lacoste\t\t"; break;

case 9: cout << "Vans\t\t"; break;

case 10: cout << "Porronet\t"; break;

}

cout << array\_shoes->priseOfShue << "\t\t";

cout << array\_shoes->sizeOfShue << "\t\t";

cout << array\_shoes->article\_number << "\t\t";

cout << array\_shoes->nameOfShue << endl;

}

void push\_back(int& N, Shoes\*& array\_shoes)

{

system("CLS");

Shoes\* newArray = new Shoes[N + 1]; // создаём массив размером на 1 больше исходного.

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

{

(newArray + i)->article\_number = (array\_shoes + i)->article\_number; // заполняем новый массив данными исходного массива.

(newArray + i)->nameOfShue = (array\_shoes + i)->nameOfShue;

(newArray + i)->priseOfShue = (array\_shoes + i)->priseOfShue;

(newArray + i)->sizeOfShue = (array\_shoes + i)->sizeOfShue;

(newArray + i)->brand\_name = (array\_shoes + i)->brand\_name;

}

// вводим значения добавленной записи массива структур.

cout << "(0 - Come back)\nEnter the name of the shoe brand: 1. Adidas, 2. Nike, 3. Ecco, 4. Barker, 5. Crocs, 6. Columbia, 7. Converse, 8. Lacoste, 9. Vans, 10. Porronet: ";

int answer;

while (true)

{

cin >> answer;

if (answer == 0) { system("CLS"); return; }

switch (answer)

{

case 1: (newArray + N)->brand\_name = BRAND\_ADIDAS; break;

case 2: (newArray + N)->brand\_name = BRAND\_NIKE; break;

case 3: (newArray + N)->brand\_name = BRAND\_ECCO; break;

case 4: (newArray + N)->brand\_name = BRAND\_BARKER; break;

case 5: (newArray + N)->brand\_name = BRAND\_CROCS; break;

case 6: (newArray + N)->brand\_name = BRAND\_COLUMBIA; break;

case 7: (newArray + N)->brand\_name = BRAND\_CONVERSE; break;

case 8: (newArray + N)->brand\_name = BRAND\_LACOSTE; break;

case 9: (newArray + N)->brand\_name = BRAND\_VANS; break;

case 10: (newArray + N)->brand\_name = BRAND\_PORRONET; break;

}

cout << "=============================" << endl;

if ((newArray + N)->brand\_name > 0 and (newArray + N)->brand\_name < 11) break;

else cout << "You entered incorrect data! Try again: " << endl;

}

cout << "Enter the price of shoes: ";

cin >> (newArray + N)->priseOfShue;

cout << "=============================" << endl;

cout << "Enter shoe size: ";

cin >> (newArray + N)->sizeOfShue;

cout << "=============================" << endl;

cout << "Enter the article of the shoe. The article must begin with the letters M (for men), W (for women) and C (for children): ";

cin >> (newArray + N)->article\_number;

cout << "=============================" << endl;

cout << "Enter the name of the shoe: ";

cin >> (newArray + N)->nameOfShue;

cout << "=============================" << endl;

N++;

system("CLS");

cout << "Shoes added." << endl << endl;

pause(); system("CLS");

delete[] array\_shoes; // освобождаем память, в которой хранятся данные исходного массива.

array\_shoes = newArray; // теперь наш указатель указывает на новый массив.

}

void pop\_back(int& N, Shoes\*& array\_shoes)

{

if (!isEmpty(N))

{

system("CLS");

int answer;

show\_list(array\_shoes, N);

cout << "(0 - Come back)\nEnter the number of the shoes you want to remove from the list: ";

while (true)

{

cin >> answer;

if (answer == 0) { system("CLS"); return; }

if (Test(answer) and answer > 0 and answer <= N)

break;

else

cout << "You entered incorrect data! Try again: ";

}

swap(\*(array\_shoes + answer - 1), \*(array\_shoes + N - 1)); // меняем местами запись, которую мы выбрали для удаления с последней.

Shoes\* newArr = new Shoes[--N]; // создали массив на единицу меньше исходного.

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

{

(newArr + i)->article\_number = (array\_shoes + i)->article\_number; // заполняем новый массив данными исходного массива.

(newArr + i)->nameOfShue = (array\_shoes + i)->nameOfShue;

(newArr + i)->priseOfShue = (array\_shoes + i)->priseOfShue;

(newArr + i)->sizeOfShue = (array\_shoes + i)->sizeOfShue;

(newArr + i)->brand\_name = (array\_shoes + i)->brand\_name;

}

delete[] array\_shoes; // освобождаем память, в которой хранятся данные исходного массива.

array\_shoes = newArr; // теперь наш указатель указывает на новый массив.

system("CLS");

cout << "Shoes deleted!" << endl << endl;

pause(); system("CLS");

}

else

{

system("CLS");

cout << "You cannot delete anything, as there is no stock." << endl << endl;

pause(); system("CLS");

}

}

bool isEmpty(int N)

{

bool flag;

(N > 0) ? flag = false : flag = true;

return flag;

}

bool Test(int ind)

{

if (cin.good()) return true; // // проерка на принадлежноть вводимого типа с типом переменной

else

{

cin.clear(); // Очищает флаг ошибки на cin (чтобы будущие операции ввода-вывода работали правильно)

cin.ignore(32767, '\n'); // Переход к следующей строке

return false;

}

}

**Файл workingWithFile.h**

#pragma once

#include <fstream>

#include "shoe.h"; // подключаем заголовочный файл shoe.h, потому что там хранится наша структура.

#include "editing.h"; // подключаем заголовочный файл editing.h, потому что там хранятся функции, которые мы используем в соответствующем cpp файле

int numberOfRecords(); // функция, возвращающая из файла количество записей в массиве структур.

void readFile(Shoes\*, int); // считываем записи из файла Shoes.dat в нашу структуру.

void saveShoes(Shoes\*, int); // сохраняем записи в файл Shoes.dat.

void keyOfAdminWrite(); // функция изменения пароля.

string keyOfAdminRead(); // функция, которая возвращает пароль.

**Файл workingWithFile.cpp**

#include "workingWithFile.h";

void readFile(Shoes\* array\_shoes, int N)

{

string path = "Shoes.dat"; // путь к файлу.

ifstream fin; // отвечает за ввод данных из файла.

fin.open(path, ios::binary);

int MUSOR; // переменная для считывания колиечества записей ( она нам не нужна, так как у нас отдельная функция, которая возвращает количество записей)

if (!fin.is\_open())

cout << "File opening error!\n";

else

{

fin.read((char\*)&(MUSOR), sizeof(MUSOR));

for (int i = 0; i < N; i++) // ctrl+alt

{

fin.read((char\*)&(array\_shoes->brand\_name), sizeof(array\_shoes->brand\_name)); //считываем переменные типа int и Brand (enum).

fin.read((char\*)&(array\_shoes->priseOfShue), sizeof(array\_shoes->priseOfShue));

fin.read((char\*)&(array\_shoes->sizeOfShue), sizeof(array\_shoes->sizeOfShue));

int len; //считываем строку, чтобы ее считать нужно сначала считать длину строки в переменную len,

fin.read((char\*)&len, sizeof(len)); //а потом саму строку в поле структуры размера len.

array\_shoes->article\_number.resize(len);

fin.read((char\*)&(array\_shoes->article\_number[0]), len);

fin.read((char\*)&len, sizeof(len));

array\_shoes->nameOfShue.resize(len);

fin.read((char\*)&(array\_shoes->nameOfShue[0]), len);

array\_shoes++;

}

}

fin.close();

}

int numberOfRecords()

{

string path = "Shoes.dat";

ifstream fin; // отвечает за ввод данных из файла.

fin.open(path, ios::binary);

int number;

if (!fin.is\_open())

cout << "File opening error!\n";

else

fin.read((char\*)&(number), sizeof(number)); // считываем в переменную number количество записей массива структур.

fin.close();

return number;

}

void saveShoes(Shoes\* array\_shoes, int N)

{

string path1 = "Shoes.dat";

ofstream fout; // отвечает за вывод данных в файл.

fout.open(path1, ios::binary);

if (!fout.is\_open())

cout << "Error opening file!" << endl;

else

{

fout.write((char\*)&(N), sizeof(N)); // записываем количество записей в файл.

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

{

fout.write((char\*)&(array\_shoes->brand\_name), sizeof(array\_shoes->brand\_name)); // записываем переменные типа int и Brand (enum).

fout.write((char\*)&(array\_shoes->priseOfShue), sizeof(array\_shoes->priseOfShue));

fout.write((char\*)&(array\_shoes->sizeOfShue), sizeof(array\_shoes->sizeOfShue));

int len = array\_shoes->article\_number.size();

fout.write((char\*)&len, sizeof(len)); // записываем сначала длину строки в файл, а потом саму строку.

fout.write(array\_shoes->article\_number.c\_str(), len);

len = array\_shoes->nameOfShue.size();

fout.write((char\*)&len, sizeof(len));

fout.write(array\_shoes->nameOfShue.c\_str(), len);

array\_shoes++;

}

cout << "Save was successful!" << endl << endl;

}

fout.close();

}

void keyOfAdminWrite()

{

string password;

string path1 = "Password.dat";

ofstream fout; // отвечает за вывод данных в файл.

system("CLS");

cout << "Change Password\n\nEnter a new password (without spaces): " << endl; // меню.

cin >> password;

fout.open(path1, ios::binary);

if (!fout.is\_open())

cout << "Error opening file!" << endl;

else

{

int len = password.size();

fout.write((char\*)&len, sizeof(len));

fout.write(password.c\_str(), len);

}

system("CLS");

cout << "Password changed successfully. Don't forget it!\nClose the program to save the password." << endl << endl;

pause();

system("CLS");

}

string keyOfAdminRead()

{

string path = "Password.dat";

ifstream fin; // отвечает за ввод данных из файла

string password;

fin.open(path, ios::binary);

if (!fin.is\_open())

cout << "File opening error!\n";

else

{

int len;

fin.read((char\*)&len, sizeof(len));

password.resize(len);

fin.read((char\*)&(password[0]), len);

}

return password;

}

**Файл dataEditing.h**

#pragma once

#include "train.h"

#include "ticket.h"

void buyTicket(Train trainArr[], Ticket ticketArr[], int ticketsNum, int trainsNum); // Functiom adds entries to the ticket array

void deleteData(Ticket ticketArr[], Train trainArr[], int ticketNum, int trainsNum); // Functiom deletes entries in the ticket array

**Файл main.cpp**

#include "editing.h"; // редактирование.

#include "Processing.h"; // обработка.

#include "workingWithFile.h"; // работа с файлами.

#include "shoe.h"; // структура обуви

void user\_functionality(Shoes\*& , int&); // функция выводит функционал пользователя.

void administrator\_functionality(Shoes\*& , int&); // функция выводит функционал администратора.

bool checkPassword(string); // проверяет пароль на правильность

int main()

{

setlocale(LC\_ALL, "Rus");

int N = numberOfRecords();

Shoes\* array\_shoes = new Shoes[N]; // объявляем массив структур размера N.

readFile(array\_shoes, N);

//keyOfAdminWrite();

string PASSWORDFile = keyOfAdminRead(); // пароль администратора.

int answer;

bool flag = true;

while (flag)

{

while (true) // проверка на принадлежноть вводимого типа с типом переменной

{

cout << "1 - Log in as a user\n2 - Log in as an administrator\n3 - Save and exit\n->";

cin >> answer;

if (Test(answer)) break;

else { system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; }

}

switch (answer)

{

case 1: user\_functionality(array\_shoes, N); break;

case 2: system("CLS"); cout << "Enter the password: ";

if (checkPassword(PASSWORDFile)) { administrator\_functionality(array\_shoes, N); break; }

else { cout << "Invalid password!" << endl << endl; pause(); system("CLS"); break; }

case 3: flag = false; break;

default: system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; break;

}

}

system("CLS");

saveShoes(array\_shoes, N);

delete[] array\_shoes; // очистка памяти

array\_shoes = NULL;

return 0;

}

void user\_functionality(Shoes\*& array\_shoes, int &N)

{

system("CLS");

cout << "Welcome to our shoe store!" << endl << endl;

int answer;

bool flag = true;

while (flag)

{

while (true) // проверка на принадлежноть вводимого типа с типом переменной

{

cout << "1 - View all shoes\n2 - Sorting for shoes by criteria\n3 - Search for shoes by criteria\n4 - Displaying a list of shoes of article x and size y\n5 - Come back\n->";

cin >> answer;

if (Test(answer)) break;

else { system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; }

}

switch (answer)

{

case 1: system("CLS"); show\_list(array\_shoes, N); pause(); system("CLS"); break;

case 2: sorting(array\_shoes, N); pause(); system("CLS"); break;

case 3: search\_Shoe(array\_shoes, N); break;

case 4: individual\_task(array\_shoes, N); break;

case 5: flag = false; system("CLS"); break;

default: system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; break;

}

}

}

void administrator\_functionality(Shoes\*& array\_shoes, int &N)

{

system("CLS");

cout << "You are logged in as an administrator!" << endl << endl;

int answer;

bool flag = true;

while (flag)

{

while (true) // проверка на принадлежноть вводимого типа с типом переменной

{

cout << "1 - View all shoes\n2 - Add shoes\n3 - Delete shoes\n4 - Change password\n5 - Come back\n->";

cin >> answer;

if (Test(answer)) break;

else { system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; }

}

switch (answer)

{

case 1: system("CLS"); show\_list(array\_shoes, N); pause(); system("CLS"); break;

case 2: push\_back(N, array\_shoes); break;

case 3: pop\_back(N, array\_shoes); break;

case 4: keyOfAdminWrite(); break;

case 5: flag = false; system("CLS"); break;

default:system("CLS"); cout << "You entered incorrect data! Try again: " << endl << endl; break;

}

}

}

bool checkPassword(string PASSWORDFile)

{

string password;

cin >> password;

if (password == PASSWORDFile) return true;

else return false;

}