МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ОТЧЕТ

по лабораторной работе №11

«Бинарные и текстовые файлы»

**Выполнил:** Антонюк Никита

**Группа:** ПО-11

**Проверил:** Войцехович О. Ю.

Брест 2023

**Цель работы**: Изучить принципы программирования с использованием бинарных файлов. Ознакомиться с основными функциями в Си для работы с бинарными файлами.

Ход работы

**Вариант 9**

**Задание:**

В программу разработанную в лабораторной работе 10 добавить

чтение и сохранение данных массива структур при помощи

бинарных файлов следующим образом:

1.При первом запуске программы должен создаваться бинарный

или текстовый файл на выбор пользователя для хранения данных

из массива структур.

2.При добавлении новой записи в массив структур в файл должна

дописываться новая запись, без изменения остальных записей.

3.При повторном запуске программы, если файл уже существует,

то информация в массив структур должна читаться из этого

файла. Если файл отсутствует, то он должен создаваться (см.

Пункт 1).

4.Все изменения (сортировка, изменения полей записи, удаление

записи) – сохраняются в файле при помощи полной перезаписи

содержимого.

5.Сделать вывод о том, какие преимущества использования

конкретного типа файлов (бинарные или текстовые) в решаемой

вами задаче.

#include <iostream>

#include <fstream>

#include <cstring>

#include <string>

#include <vector>

using namespace std;

enum class ItemType {

Food,

Technics,

Clothes

};

union Sum {

int Amount;

int Price;

};

struct Production {

char name[100];

int amount;

int PricePerUnit;

string Date;

int sum;

};

void GetElement(Production\* Base, int N) {

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

cout << "Введите наименование товара: ";

cin >> Base[i].name;

cout << "Введите количество товара: ";

cin >> Base[i].amount;

cout << "Введите цену товара за единицу: ";

cin >> Base[i].PricePerUnit;

cout << "Введите дату: ";

cin >> Base[i].Date;

cout << "";

}

}

void ShowElement(Production\* Base, int N) {

char tovar[100];

cout << "Введите товар: ";

cin >> tovar;

cout << "Вывод данных" << endl;

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

if (!strcmp(tovar, Base[i].name)) {

Base[i].sum = Base[i].amount \* Base[i].PricePerUnit;

if (Base[i].sum > 100000) {

cout << "Этот товар купили на сумму: " << endl;

cout << Base[i].sum;

}

else {

cout << "Этот товар купили на сумму меньше 100000" << endl;

}

}

}

}

void SortData(Production\* Base, int N) {

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

for (int j = 0; j < N - i - 1; j++) {

if (Base[j].PricePerUnit > Base[j + 1].PricePerUnit) {

Production temp = Base[j];

Base[j] = Base[j + 1];

Base[j + 1] = temp;

}

}

}

}

void DeleteElements(Production\* Base, int& N) {

char tovar[100];

cout << "Введите товар: " << endl;

cin >> tovar;

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

if (!strcmp(tovar, Base[i].name)) {

for (int j = i; j < N - 1; j++) {

Base[j] = Base[j + 1];

}

N--;

cout << "Запись удалена!" << endl;

return;

}

}

cout << "Запись не найдена." << endl;

}

void SaveData(Production\* Base, int N) {

ofstream outFile("data.bin", ios::binary);

if (!outFile) {

cout << "Ошибка при открытии файла для записи." << endl;

return;

}

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

outFile.write(reinterpret\_cast<char\*>(&Base[i]), sizeof(Production));

}

outFile.close();

cout << "Данные успешно сохранены." << endl;

}

void ReadData(Production\* Base, int& N) {

ifstream inFile("data.bin", ios::binary);

if (!inFile) {

cout << "Файл не найден. Создание нового файла." << endl;

SaveData(Base, N);

return;

}

N = 0;

while (inFile.read(reinterpret\_cast<char\*>(&Base[N]), sizeof(Production))) {

N++;

}

inFile.close();

cout << "Данные успешно загружены." << endl;

}

int main() {

setlocale(LC\_ALL, "ru");

int N;

cout << "Введите количество записей: ";

cin >> N;

Production\* Base = new Production[N];

ReadData(Base, N);

int choice;

do {

cout << endl;

cout << "Меню:" << endl;

cout << "1. Ввод элементов" << endl;

cout << "2. Вывод товаров" << endl;

cout << "3. Сортировка" << endl;

cout << "4. Удаление элементов" << endl;

cout << "5. Сохранение данных" << endl;

cout << "0. Выход" << endl;

cin >> choice;

switch (choice) {

case 1:

GetElement(Base, N);

break;

case 2:

ShowElement(Base, N);

break;

case 3:

SortData(Base, N);

break;

case 4:

DeleteElements(Base, N);

break;

case 5:

SaveData(Base, N);

break;

case 0:

break;

default:

cout << "Некорректный ввод. Попробуйте снова." <<

endl;

break;

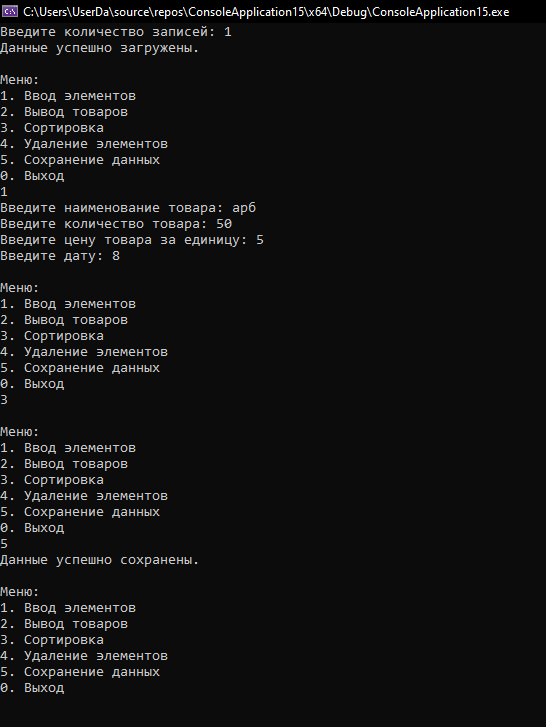
}

} while (choice != 0);

delete[] Base;

return 0;

}



Создался бинарный файл:



Создался текстовый файл:



Вывод: Изучил принципы программирования с использованием

бинарных файлов. Ознакомилась с основными функциями в Си для

работы с бинарными файлами.

**Текстовый файл**

Любой файл по умолчанию является текстовым файлом. Текстовые файлы могут быть легко перенесены из одной компьютерной системы в другую. Эти файлы легко читаются, т.к. содержимое, записанное в текстовых файлах, человек может прочитать.

**Бинарный файл**

Бинарные файлы не могут быть легко перенесены из одной компьютерной системы в другую из-за различий во внутреннем представлении, которое варьируется от компьютера к компьютеру. Он хранит данные в двоичном формате. Это экономит память, потому что данные любого типа будут храниться в памяти в соответствии с размером памяти. Содержимое, записанное в бинарных файлах, не читается человеком.