МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ОТЧЕТ

по лабораторной работе №12

«Индексные файлы»

**Выполнил:** Антонюк Н.А.

**Группа:** ПО-11

**Проверил:** Войцехович О. Ю.

Брест 2023

**Цель работы**: Изучить основные принципы программирования с использованием индексных файлов

Ход работы

**Вариант 2**

**Задание:**

**Список товаров, имеющихся на складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести список товаров, стоимость которых превышает 100 000 рублей.**

**В программу разработанную в лабораторной работе 12 внести следующие изменения и дополнения:**

**1. При запуске программы данные читаются в массив структур из файла, при добавлении новой записи в массив структур в файл должна дописываться новая запись, без изменения остальных записей.**

**2. Все изменения (изменения полей записи, удаление записи) – сохраняются в файле при помощи перезаписи содержимого всего файла.**

**3. Сортировка должна выполняться по двум полям на выбор при помощи создания индексных файлов. Содержимое индексного файла переписывается в случае изменения значения ключевого поля (поля, по которому выполняется сортировка) или в случае удаления, добавления записей.**

**Код С++:**

**#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;**

**};**

**struct Index {**

**string fieldName;**

**string indexFile;**

**};**

**void CreateIndex(Production\* Base, int N, const Index& index) {**

**// Создание индексного файла на основе указанного поля с использованием сортировки**

**if (index.fieldName == "name") {**

**// Сортировка массива Base по полю "name"**

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

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

**if (strcmp(Base[j].name, Base[j + 1].name) > 0) {**

**Production temp = Base[j];**

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

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

**}**

**}**

**}**

**}**

**else if (index.fieldName == "PricePerUnit") {**

**// Сортировка массива Base по полю "PricePerUnit"**

**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;**

**}**

**}**

**}**

**}**

**else {**

**cout << "Некорректное поле для создания индекса." << endl;**

**return;**

**}**

**// Запись индексов в индексный файл**

**ofstream indexFile(index.indexFile, ios::binary);**

**if (!indexFile) {**

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

**return;**

**}**

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

**indexFile.write(reinterpret\_cast<const char\*>(&i), sizeof(int));**

**}**

**indexFile.close();**

**}**

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

**cout << "Введите наименование товара: ";**

**cin >> Base[N].name;**

**cout << "Введите количество товара: ";**

**cin >> Base[N].amount;**

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

**cin >> Base[N].PricePerUnit;**

**cout << "Введите дату: ";**

**cin >> Base[N].Date;**

**cout << "";**

**N++;**

**}**

**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, const string& fieldName) {**

**if (fieldName == "name") {**

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

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

**if (strcmp(Base[j].name, Base[j + 1].name) > 0) {**

**Production temp = Base[j];**

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

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

**}**

**}**

**}**

**}**

**else if (fieldName == "PricePerUnit") {**

**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;**

**}**

**}**

**}**

**}**

**else {**

**cout << "Некорректное поле для сортировки." << endl;**

**}**

**}void SortData(Production\* Base, int N, const string& fieldName, vector<Index>& indexes) {**

**if (fieldName == "name") {**

**}**

**else if (fieldName == "PricePerUnit") {**

**}**

**else {**

**cout << "Некорректное поле для сортировки." << endl;**

**return;**

**}**

**Index index;**

**index.fieldName = fieldName;**

**index.indexFile = fieldName + ".index";**

**CreateIndex(Base, N, index);**

**indexes.push\_back(index);**

**}**

**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 | ios::app); // Добавлен флаг ios::app для дописывания в файл**

**if (!outFile) {**

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

**return;**

**}**

**outFile.write(reinterpret\_cast<char\*>(Base), N \* 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;**

**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 = 0;**

**Production\* Base = new Production[100];**

**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:**

**{**

**string fieldName;**

**cout << "Выберите поле для сортировки (name, PricePerUnit): ";**

**cin >> fieldName;**

**SortData(Base, N, fieldName);**

**}**

**break;**

**case 4:**

**DeleteElements(Base, N);**

**break;**

**case 5:**

**SaveData(Base, N);**

**break;**

**case 0:**

**break;**

**default:**

**cout << "Некорректный ввод. Попробуйте снова." << endl;**

**break;**

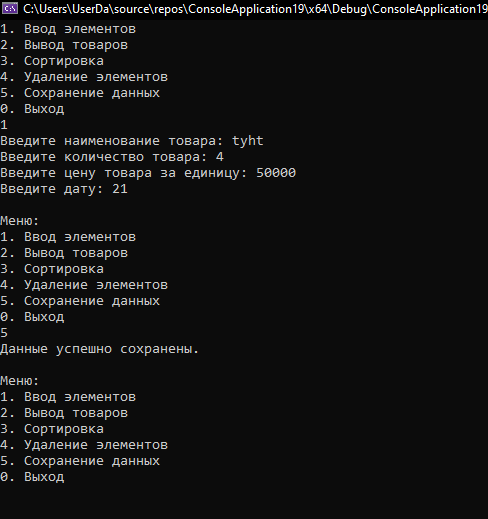
**}**

**} while (choice != 0);**

**delete[] Base;**

**return 0;**

**}**

****

**Вывод: : Изучил принципы работы с индекс файлами. Переделал программу согласно заданию**