Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине: «ОАиП»

За 2 семестр

**Тема:** «Индексные файлы»

Выполнила:

Студентка 1 курса

Группы ПО-6(2)

Хурсина Е. П.

Проверил:

Хацкевич М. В.

Брест, 2021

Лабораторная работа №4

Индексные файлы

**Цель:** изучить принципы программирования с использованием индексных файлов.

**Ход работы**

**Задание:**

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.

**Блок-схема:**

Описание функции Void poisk(date\* date\_p, radio\* (&radio\_t), int& MAX). С помощью этой функции производиться поиск элемента массива до году сдачи в ремонт



**Текст программы:**

#include <iostream>

#include <string>

#include <Windows.h>

#include <locale.h>

#include <iomanip>

#include <fstream> // функционал для считывания данных из файла и для записи в файл

using namespace std;

enum texnika {

Бытовая\_техника = 1, Медицинская\_техника, Офисная\_техника, Вычислительная\_техника, Видеотехника

};

struct date {

int day;

char month[20];

int year;

};

struct radio {

char naim[25];

struct date;

int d;

};

void input(date\* date\_p, radio\* (&radio\_t), int& MAX);// ввод данных массива

void list(date\* date\_p, radio\* radio\_t, int& MAX); //вывод данныйх массива

void sort(date\* date\_p, radio\* (&radio\_t), int& MAX, char \*name); //сортировка

void poisk(date\* date\_p, radio\* (&radio\_t), int& MAX); //поиск по дате(год)

void delete\_t(date\* date\_p, radio\* (&radio\_t), int& MAX); //удаление структуры из массива по номеру в списке

void write\_tehnica(date\* date\_p, radio\* (&radio\_t), int MAX, int NUM, char \*name);

void read\_tehnica(date\* (&date\_p), radio\* (&radio\_t), int& MAX);

void add\_tehnica(date\* (&date\_p), radio\* (&radio\_t), int& MAX, int k, char \*name);

void index\_file(int MAX, int\* index);

int main() {

setlocale(0, "");

system("color F0");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int ch, key;

int MAX = 0;

int NUM = 0;

char name;

radio\* radio\_t = new radio[100];

date\* date\_p = new date[100];

cout << "Работа с текстовым(1) или с бинарным файлом(0)? :\n";

cin >> key;

while (1) { // организация меню

cout << "Выберите дейсвие:\n";

cout << " 1. Ввод массива\n";

cout << " 2. Вывод массива\n";

cout << " 3. Сортировка массива структур по году сдачи в ремонт\n";

cout << " 4. Поиск\n";

cout << " 5. Удаление\n";

cout << " 6. Выход\n";

cout << "Выбирете номер пункта: ";

cin >> ch;

switch (ch) {

case 1:

write\_tehnica(date\_p, radio\_t, MAX, NUM, name);

input(date\_p, radio\_t, MAX);

break;

case 2:

list(date\_p, radio\_t, MAX);

break;

case 3:

sort(date\_p, radio\_t, MAX, name);

list(date\_p, radio\_t, MAX);

break;

case 4:

poisk(date\_p, radio\_t, MAX);

break;

case 5:

delete\_t(date\_p, radio\_t, MAX);

break;

case 6: return 0;

}

cout << "Выйти из программы? (1-да/ 0-нет): ";

cin >> ch;

if (ch) break;

}

system("pause");

return 0;

}

void input(date\* date\_p, radio\* (&radio\_t), int& MAX) {

cout << "Введите кол-во техники, которую сдали в ремонт: ";

cin >> MAX;

for (int t = 0; t < MAX; t++) {

cout << "\nВыберите категорию техники: \n1 -Бытовая\_техника \n2 - Медицинская\_техника \n3 - Офисная\_техника \n4 - Вычислительная\_техника \n5 - Видеотехника " << endl

<< "Ваш выбор: ";

int m;

cin >> m;

cout << "\nНаименование изделия: ";

cin.ignore(); //очистка входного буфера

cin >> radio\_t[t].naim;

cout << "Дата сдачи в ремонт.\n ";

cout << "Введите день: ";

cin >> date\_p[t].day;

cout << "Введите месяц: ";

cin >> date\_p[t].month;

cout << "Введите год: ";

cin >> date\_p[t].year;

cout << "Состояние готовности заказа(1-выполнено/0- не выполнено): ";

cin >> radio\_t[t].d;

cin.ignore();

}

}

void list(date\* date\_p, radio\* radio\_t, int& MAX) {

int k;

cout << " --------------------------------------------------------------------------------\n";

cout << " | № | Наименование | Дата сдачи | Состояние |\n";

cout << " | | изделия | в ремонт | готовности |\n";

cout << " --------------------------------------------------------------------------------\n";

for (int t = 0; t < MAX; t++) {

k = t + 1;

if (radio\_t[t].d == 0) {

cout << setw(3) << " | " << k << "| " << setw(3) << radio\_t[t].naim << " | " << date\_p[t].day << ", " << setw(5) << date\_p[t].month << setw(5) << date\_p[t].year << " | " << setw(10) << radio\_t[t].d << setw(5) << " | " << endl;

cout << " --------------------------------------------------------------------------------\n";

}

}

}

void sort(date\* date\_p, radio\* (&radio\_t), int& MAX, char \*naim)

{

int choise;

cout << "Выберите пункт:\n";

cout << "1 - Сортировка по году сдачи в ремент.";

cout << "2 - Сортировка по наименованию изделия.";

cin >> choise;

int\* index = new int[MAX];

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

for (int j = i; j < MAX; j++) {

if (choise == 1) {

if (strcmp(radio\_t[index[j]].naim, radio\_t[index[i]].naim) < 0) { // strcmp лексикографическую проверку двух строк, оканчивающихся нулевыми символами, и возвращает целое число со следующим значением

swap(index[i], index[j]); // передаём ссылки

}

}

else if (choise == 0) {

if (radio\_t[index[j]].naim < radio\_t[index[i]].naim) {

swap(index[i], index[j]);

}

}

}

}

index\_file(MAX, index);

}

void poisk(date\* date\_p, radio\* (&radio\_t), int& MAX) {

int year;

cout << "Введите год сдачи в ремонт: ";

cin >> year;

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

if (date\_p[i].year == year) {

i++;

cout << "Наименование изделия: " << radio\_t[i].naim << endl;

cout << "День сдачи в ремонт: " << date\_p[i].day << endl;

cout << "Месяц сдачи в ремонт: " << date\_p[i].month << endl;

cout << "Год сдачи в ремонт: " << date\_p[i].year << endl;

cout << "Состояние готовности: " << radio\_t[i].d << endl;

}

else {

cout << "Введины неверные данные!!!" << endl;}

}

}

void delete\_t(date\* date\_p, radio\* (&radio\_t), int& MAX) {

int n, k;

int NUM = 0;

char name;

cout << "Введите номер записи, которую хотите удалить: ";

cin >> n;

n--; //правильное считывание массива(т.е. не с 0, а с 1)

for (n = 0; n < MAX; n++) {

strcpy\_s(radio\_t[n].naim, radio\_t[n + 1].naim);

date\_p[n].day = date\_p[n+ 1].day;

strcpy\_s(date\_p[n].month, date\_p[n + 1].month);

date\_p[n].year = date\_p[n + 1].year;

radio\_t[n].d = radio\_t[n + 1].d;

}

MAX--;

if (MAX == 1) {

write\_tehnica(date\_p, radio\_t, MAX, NUM, name);

}

else if (MAX == 0) {

}

radio\* new\_radio\_t = new radio[MAX - 1];

int countre = 0;

for (int i = 0; i < MAX; i++) { //перераспределение памяти

if (i != n) {

new\_radio\_t[countre] = radio\_t[countre];

countre++;

}

}

delete[] radio\_t;

radio\_t = new\_radio\_t;

MAX--;

}

void write\_tehnica(date\* date\_p, radio\* (&radio\_t), int MAX, int NUM, char \*name) {

ofstream file; // создаём объект для записи в файл

int k;

if (k == 1) {

file.open(name);

}

else if (k == 0) {

file.open(name, ios\_base::binary); // открытие файла для двоичных операций.

}

ifstream index\_file;

index\_file.open("Indexes.txt");

if (k == 1) {

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

int n;

if (index\_file.is\_open() && !index\_file.eof()) {

index\_file >> n;

index\_file >> n;

}

else n = 1;

file << " --------------------------------------------------------------------------------\n";

file << " | № | Наименование | Дата сдачи | Состояние |\n";

file << " | | изделия | в ремонт | готовности |\n";

file << " --------------------------------------------------------------------------------\n";

for (int t = 0; t < MAX; t++) {

k = t + 1;

if (radio\_t[t].d == 0) {

cout << setw(3) << " | " << k << "| " << setw(3) << radio\_t[t].naim << " | " << date\_p[t].day << ", " << setw(5) << date\_p[t].month << setw(5) << date\_p[t].year << " | " << setw(10) << radio\_t[t].d << setw(5) << " | " << endl;

cout << " --------------------------------------------------------------------------------\n";

}

}

if (i < MAX - 1) {

file << endl;

}

}

}

else if (k == 0) {

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

int n;

if (index\_file.is\_open()) {

index\_file >> n;

index\_file >> n;

}

else n = i;

file.write((char\*)&date\_p[k], sizeof(date\_p[k]));

file.write((char\*)&radio\_t[k], sizeof(radio\_t[k]));

}

}

index\_file.close(); //закрываем файл

file.close();

}

void read\_tehnica(date\* (&date\_p), radio\* (&radio\_t), int& MAX) {

ifstream tehnica\_txt; // открыли файл для чтения

int k, NUM=0;

char name=0;

tehnica\_txt.open("tehnica\_txt.txt");

MAX = 0;

if (!tehnica\_txt.is\_open()) {

list(date\_p, radio\_t, MAX);

write\_tehnica(date\_p, radio\_t, MAX, NUM, name);

}

else

{

date\_p = new date[MAX];

radio\_t = new radio[MAX];

char array[1024];

tehnica\_txt.getline(array, 1024, '\n'); // Методы записи и чтения строк

for (int i = 0; !tehnica\_txt.eof(); i++) {

MAX++;

date\* array\_arr = new date[MAX - 1];

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

array\_arr[j] = date\_p[j];

}

date\_p = new date[MAX];

radio\_t = new radio[MAX];

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

date\_p[j] = array\_arr[j];

}

delete array\_arr;

tehnica\_txt >> date\_p[i].day >> date\_p[i].month >> date\_p[i].year >> radio\_t[i].naim >> radio\_t[i].d;

char array[20];

}

cout << "Данные считаны из файла.\n";

cin.ignore();

tehnica\_txt.close();

}

}

void add\_tehnica(date\* (&date\_p), radio\* (&radio\_t), int& MAX, int k, char \*name) {

ofstream file;

if (k == 1) {

file.open(name, ios\_base::app | ios\_base::binary); //открываем файл для добавления информации к концу файла(ios\_base::app)

}

input(date\_p, radio\_t, MAX);

file << endl;

write\_tehnica(date\_p, radio\_t, MAX, k, name);

list(date\_p, radio\_t, MAX);

file.close();

int\* tempArr = new int[MAX];

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

tempArr[i] = date\_p[i].year;

}

index\_file(MAX, tempArr);

delete[] tempArr;

}

void index\_file(int MAX, int\* index) {

ofstream fout("Indexes.txt");

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

fout << i + 1 << " " << index[i];

if (i < MAX - 1) {

fout << endl;

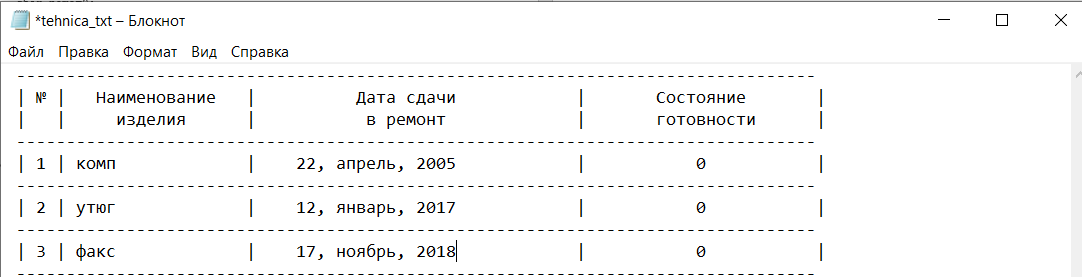
}

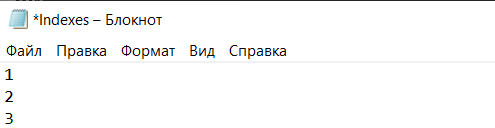
}

fout.close();

}

**Результаты тестирования программы:**





**Вывод:** в ходе лабораторной работы изучили принципы программирования с использованием индексных файлов. Индексный файл (index file) – это файл, в котором хранится информация индекса. Он является файлом, в котором каждая запись

состоит из двух значений: данных и указателя номера записи. Основная цель использования индексных файлов это ускорение процесса извлечения данных.