Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №12

за 2 семестр

По дисциплине: «ОАиП»

Тема: «Индексные файлы»

Выполнил:

Студенты 1 курса

Группы АС-59

Левоцкий Н.

Цыганов Е.

Проверил:

Гирель Т.Н.

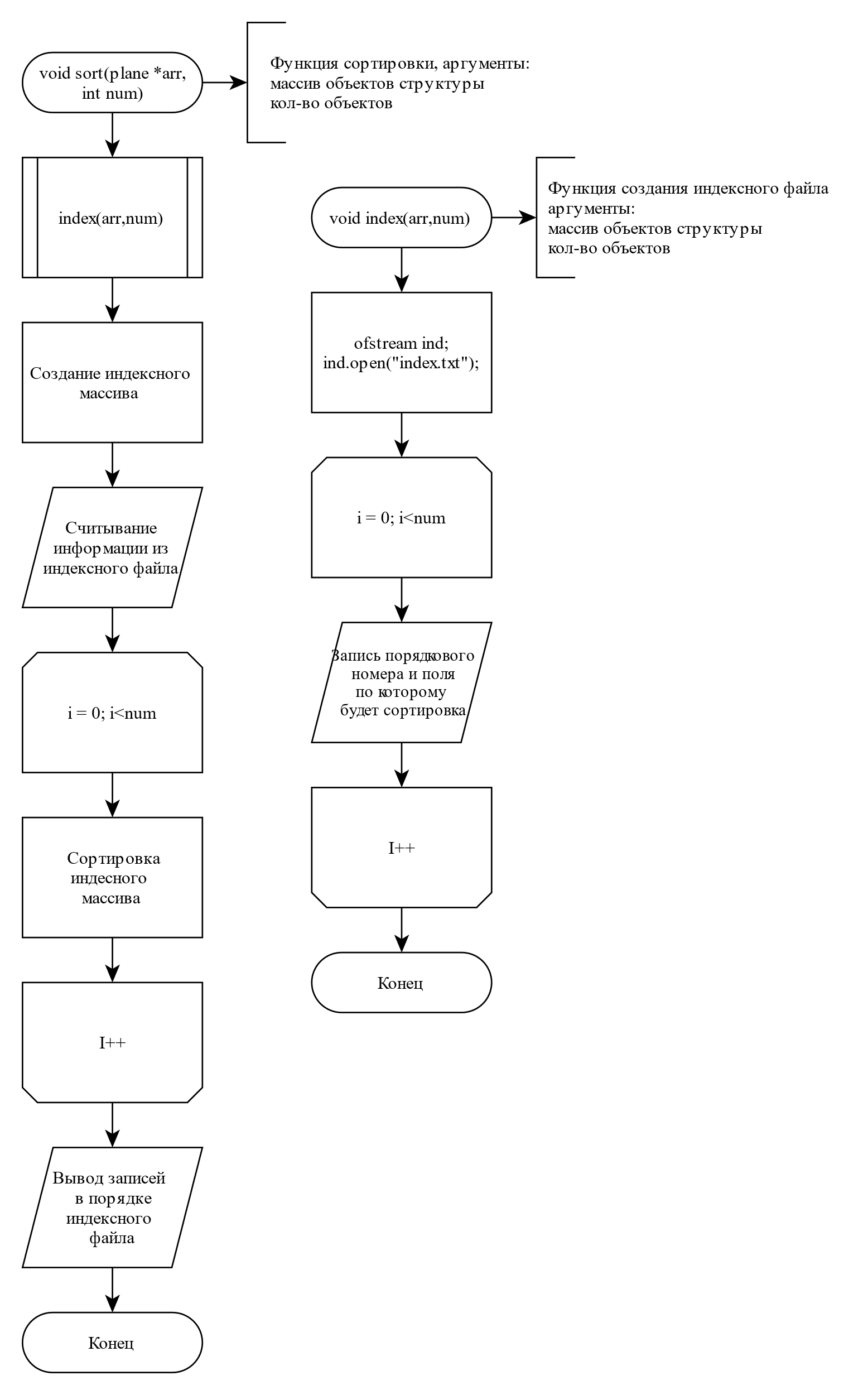
2021

Лабораторная работа №12

Индексные файлы

Цель работы: изучить принципы программирования с использованием индексных файлов.

**Блок-схема:**



**Задание:**

В программу, разработанную в лабораторной работе 11, 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.

Текст программы:

#include<iostream>

#include<Windows.h>

#include<iomanip>

#include<stdlib.h>

#include<fstream>

using namespace std;

struct plane

{

int num;

union inFlight //объединение

{

int key;

char ans[4];

}inf;

enum Type //перечисление

{

pass = 1,

trans,

mail,

cour

};

char typedes[15];

short hour;

short min;

char loc[30];

short time;

};

void menu(plane\* arr, int size); //меню

void input(plane\*& arr, int &size); //ввод

void output(plane\* arr, int size); //вывод

void del(plane\*& arr, int& size); //удаление

void sort(plane\* arr, int size); //сортировка

void task(plane\* arr, int size); //задание

void fileadd(plane\*& arr, int& size); //добавление

void filecreate(plane\* arr, int size);

void fileread(plane\*& arr, int& size);

void bincreate(plane\* arr, int size);

void binread(plane\*& arr, int& size);

void binadd(plane\*& arr, int& size);

void index(plane\* arr, int size,int key);

void sort(plane\* arr, int size)

{

int key;

cout << "1 - по времени\n2 - по пункту назначения\n";

cin >> key;

if (key == 1)

{

index(arr, size, key);

size = 0;

struct sub

{

int index;

int sHour, sMin;

};

sub\* mass = new sub[size];

ifstream ind;

ind.open("index.txt");

for (int i = 0; !ind.eof(); i++) {

size++;

sub\* tempArr = new sub[size - 1];

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

tempArr[j] = mass[j];

}

mass = new sub[size];

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

mass[j] = tempArr[j];

}

delete[] tempArr;

ind >> mass[i].index;

ind >> mass[i].sHour;

ind >> mass[i].sMin;

}

size -= 1;

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

{

for (int j = i; j < size; j++)

{

if (mass[i].sHour > mass[j].sHour||

mass[i].sHour == mass[j].sHour

&& mass[i].sMin >mass[j].sMin)

{

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

}

}

}

cout << left << setw(8) << "|Номер" << setw(14) << left << "|Тип самолёта" << setw(19) << left << "|Пункт назначения" << setw(14) << left << "|Время вылета" << setw(10) << left << "|В полёте|" << endl;

for (int j = 0; j < size; j++)

{

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

{

if(mass[j].index == arr[i].num)

cout << left << setw(9) << arr[i].num << setw(15) << left << arr[i].typedes << setw(20) << left << arr[i].loc << setw(2) << left << arr[i].hour << ":" << setw(12) << left << arr[i].min << setw(10) << left << arr[i].inf.ans << endl;

}

}

}

if (key == 2)

{

index(arr, size, key);

size = 0;

struct sub

{

int index;

char sLoc[30];

};

sub\* mass = new sub[size];

ifstream ind;

ind.open("index.txt");

for (int i = 0; !ind.eof(); i++) {

size++;

sub\* tempArr = new sub[size - 1];

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

tempArr[j] = mass[j];

}

mass = new sub[size];

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

mass[j] = tempArr[j];

}

delete[] tempArr;

ind >> mass[i].index;

ind >> mass[i].sLoc;

}

size -= 1;

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

{

for (int j = i; j < size; j++)

{

if (strcmp(mass[i].sLoc, mass[j].sLoc) > 0)

{

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

}

}

}

cout << left << setw(8) << "|Номер" << setw(14) << left << "|Тип самолёта" << setw(19) << left << "|Пункт назначения" << setw(14) << left << "|Время вылета" << setw(10) << left << "|В полёте|" << endl;

for (int j = 0; j < size; j++)

{

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

{

if (mass[j].index == arr[i].num)

cout << left << setw(9) << arr[i].num << setw(15) << left << arr[i].typedes << setw(20) << left << arr[i].loc << setw(2) << left << arr[i].hour << ":" << setw(12) << left << arr[i].min << setw(10) << left << arr[i].inf.ans << endl;

}

}

}

}

void index(plane\* arr, int size, int key)

{

ofstream ind;

ind.open("index.txt");

if (key == 1)

{

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

ind << arr[i].num << " " << arr[i].hour << " " << arr[i].min << endl;

}

if (key == 2)

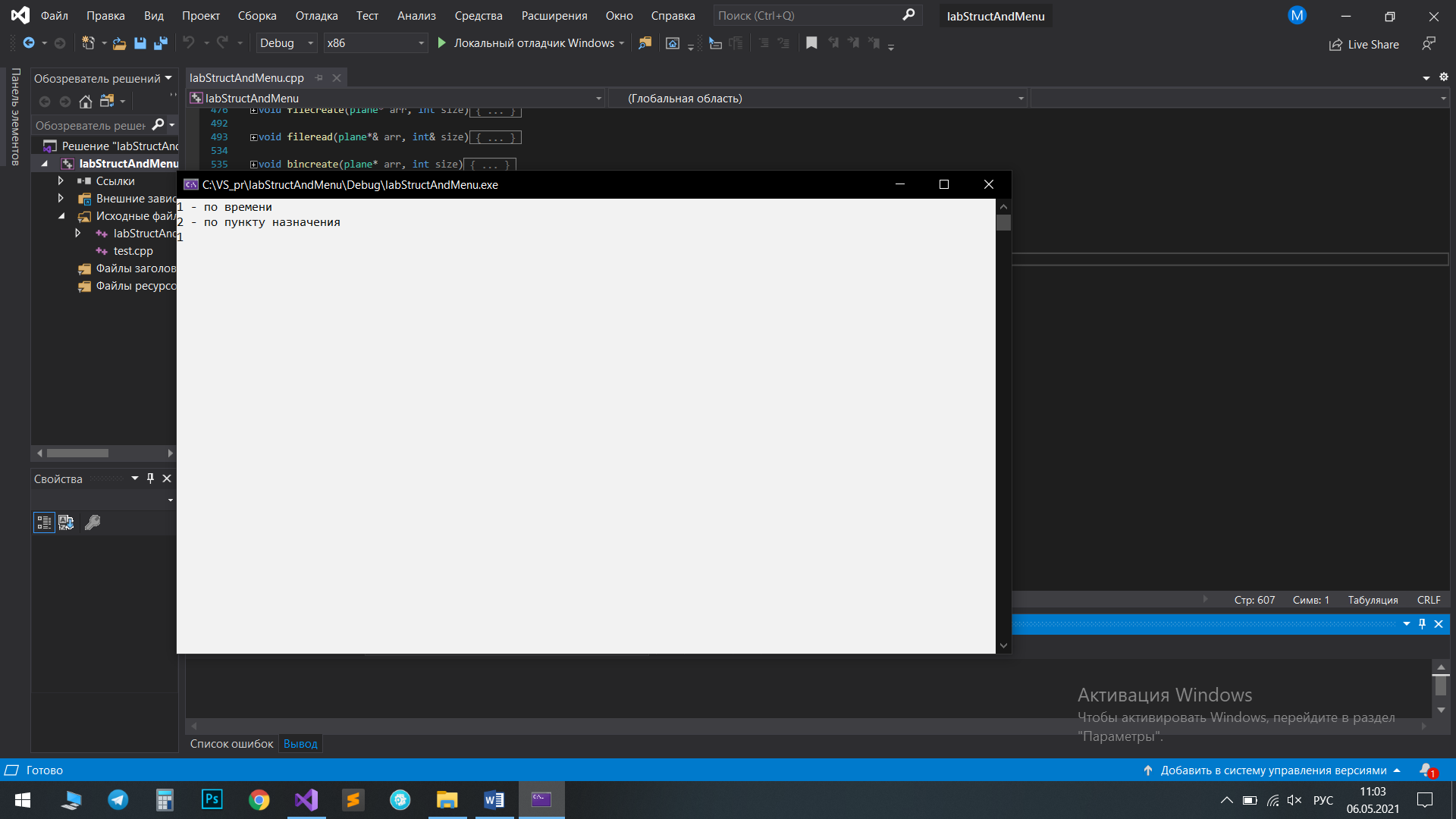
{

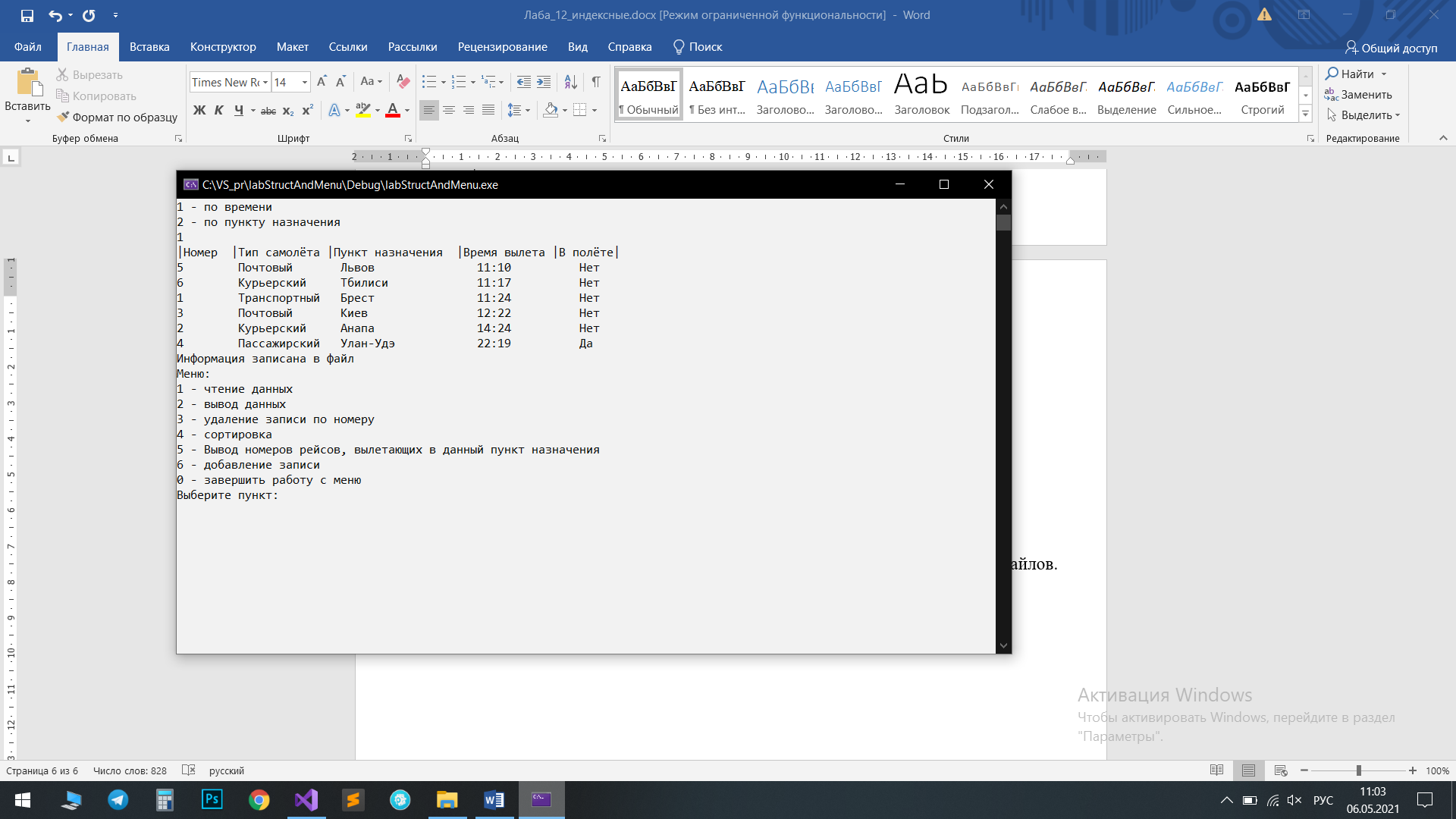
for (int i = 0; i < size; i++)

ind << arr[i].num << " " << arr[i].loc << endl;

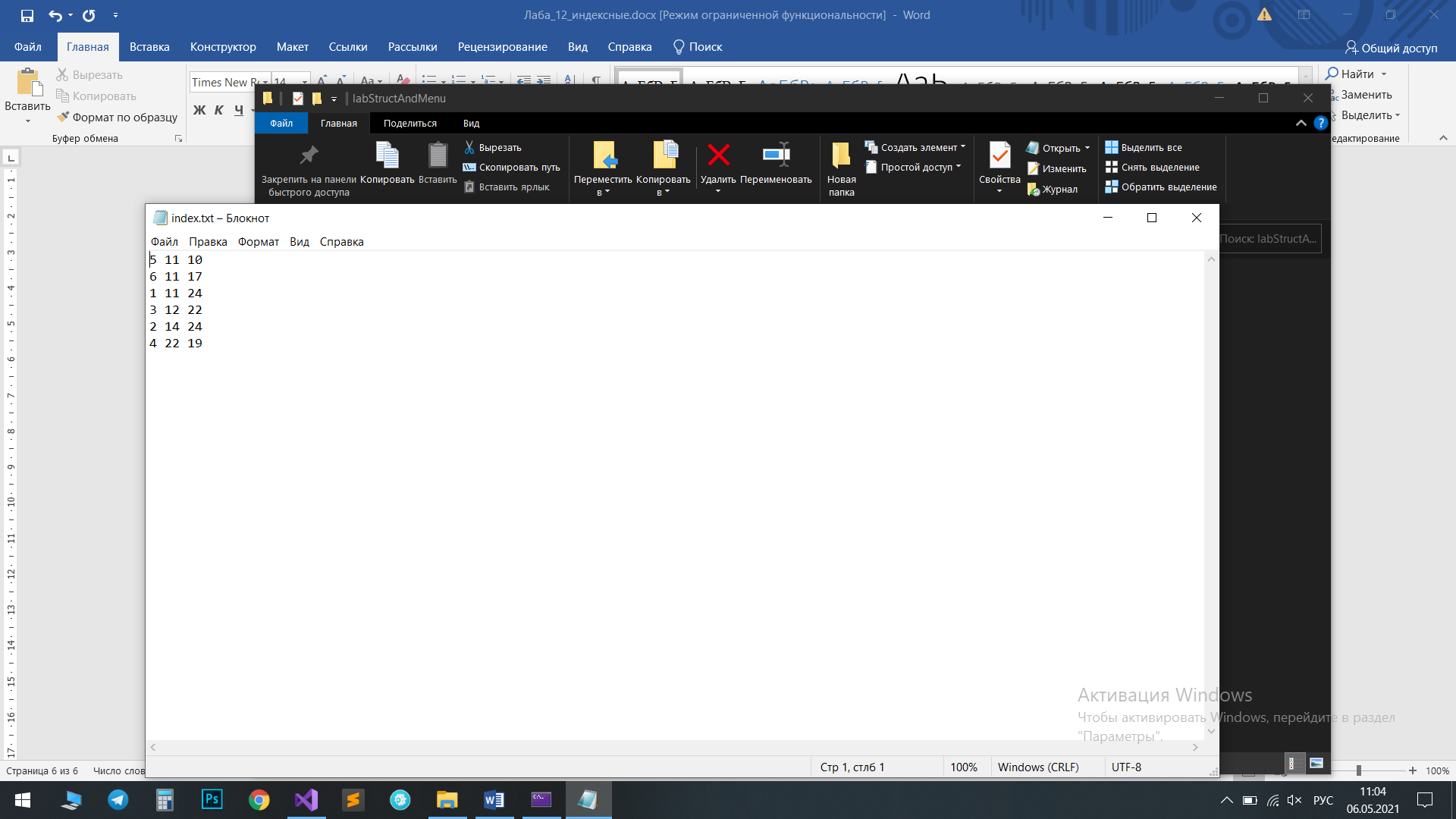
}

}





Индексный файл



Вывод: изучил принципы программирования с использованием индексных файлов.