Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №12

за 2 семестр

По дисциплине: «ОАиП»

Тема: «Индексные файлы»

Выполнила:

Студентка 1 курса

Группы ПО-3(1)

Гаврилюк Р.И.

Проверил:

Хацкевич М. В.

2019

Лабораторная работа №12

Индексные файлы

Цель работы: изучить принципы программирования с использованием индексных файлов.

Вариант 24

**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 <fstream>

#include <iomanip>

using namespace std;

enum Continent { Europe = 1, Africa, North\_America, South\_America, Australia, Asia };

union Sea {

int num;

char access;

};

struct Country {

char name[20],

capital[20],

currency[20];

int population,

num;

Continent continent;

Sea sea;

unsigned goverment : 2;

};

void menu(); //организация меню

void input(Country \*&array, int &N); //ввод данных массива структуры

void output(Country \*&array, int N, int key, char \*name); //вывод данных массива структур

void sort(Country \*array, int N, int key, char \*name); //сортировка во алфавиту по фамилии

int search(Country \*array, int N); //поиск по заданному параметру

void deleted(Country \*&array, int &N, int key, char \*name); //удаление из массива структур

void read(Country \*&array, int &N, int key, char \*name);

void write(Country \*array, int N, int key, char \*name);

void add(Country \*&array, int &N, int key, char \*name);

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

int main() {}

void menu() {}

void input(Country \*&array, int &N) {}

void output(Country \*&array, int N, int key, char \*name) {}

void sort(Country \*array, int N, int key, char \*name) {

cout << "Выберите пункт:\n";

cout << "1 - Сортировка по названию страны\n";

cout << "2 - Сортировка по количеству населения\n";

int choise;

cin >> choise;

int \*index = new int[N];

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

index[i] = i;

}

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

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

if (choise == 1) {

if (strcmp(array[index[j]].name, array[index[i]].name) < 0) {

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

}

}

else if (choise == 0) {

if (array[index[j]].population < array[index[i]].population) {

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

}

}

}

}

index\_file(N, index);

write(array, N, key, name);

output(array, N, key, name);

cin.ignore();

}

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

ofstream fout("Indexes.txt");

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

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

if (i < N - 1) {

fout << endl;

}

}

fout.close();

}

int search(Country \*array, int N){}

void deleted(Country \*&array, int &N, int key, char \*name) {

char Name[20];

cout << "Введите название страны, которую хотите удалить.\n";

gets\_s(Name);

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

if (strcmp(array[i].name, Name) == 0) {

N--;

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

array[j] = array[j + 1];

}

i--;

}

}

Country\* tempArr = new Country[N];

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

tempArr[i] = array[i];

}

array = new Country[N];

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

array[i] = tempArr[i];

}

delete[] tempArr;

system("cls");

int \*tempArray = new int[N];

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

tempArray[i] = i;

}

index\_file(N, tempArray);

delete[] tempArray;

write(array, N, key, name);

output(array, N, key, name);

}

void write(Country \*array, int N, int key, char \*name) {

ofstream file;

if (key == 1) {

file.open(name);

}

else if (key == 0) {

file.open(name, ios\_base::binary);

}

ifstream index\_file;

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

if (key == 1) {

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

int k;

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

index\_file >> k;

index\_file >> k;

}

else k = i;

file << setw(15) << left << array[k].name;

file << setw(15) << left << array[k].capital;

file << setw(15) << left << array[k].currency;

switch (array[k].continent) {

case Europe:

file << setw(20) << left << "Европа";

break;

case Africa:

file << setw(20) << left << "Африка";

break;

case North\_America:

file << setw(20) << left << "Северная\_Америка";

break;

case South\_America:

file << setw(20) << left << "Южная\_Америка";

break;

case Australia:

file << setw(20) << left << "Австралия";

break;

case Asia:

file << setw(20) << left << "Азия";

break;

}

file << setw(15) << left << array[k].sea.access;

file << setw(15) << left << array[k].population;

if (array[k].goverment == 1) {

file << left << "монархия";

}

else if (array[k].goverment == 2) {

file << left << "республика";

}

if (i < N - 1) {

file << endl;

}

}

}

else if (key == 0) {

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

int k;

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

index\_file >> k;

index\_file >> k;

}

else k = i;

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

}

}

index\_file.close();

file.close();

}

void read(Country \*&array, int &N, int key, char \*name){}

void add(Country \*&array, int &N, int key, char \*name) {

ofstream file;

if (key == 1) {

file.open(name, ios\_base::app);

}

else if (key == 0) {

file.open(name, ios\_base::app | ios\_base::binary);

}

input(array, N);

file << endl;

write(array, N, key, name);

output(array, N, key, name);

file.close();

int \*tempArr = new int[N];

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

tempArr[i] = array[i].num;

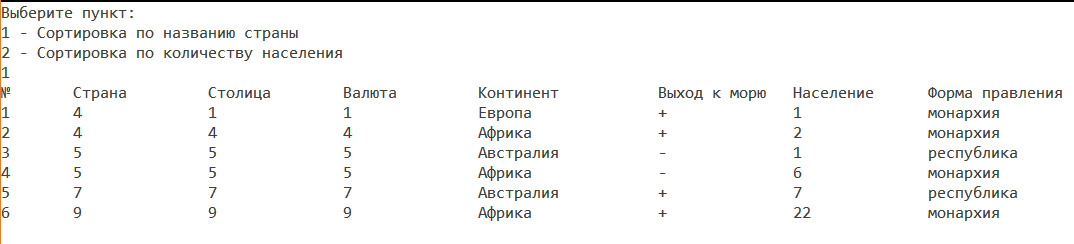
}

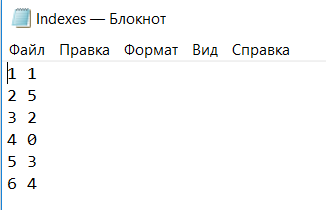
index\_file(N, tempArr);

delete[] tempArr;

}

Результаты выполнения программы:





Вывод: изучила принципы программирования с использованием индексных файлов.