Министерство образования Республики Беларусь

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Лабораторная работа №5**

**По дисциплине «Основы алгоритмизации и программирования»**

**Тема: «Потоки и файлы»**

Выполнил: Макеёнок Д.И.

Группа 21-ИТ-1

Проверила: Пантелейко А.Ф.

Преподаватель-стажер кафедры ТП

Полоцк 2022

**Цель:** изучить библиотечные функции для работы с файлами. Освоить основные методы обработки файлов.

**Вариант № 1**

1. Функция new\_DB(), в которой будет создаваться новый файл базы данных и открываться для записи. Функция должна возвращать объект потока ввода для нового файла.
2. Функция remove\_DB(), удаляющая файл по названию. В функции должна быть реализована обработка ошибок.
3. Функция clear(), которая будет очищать динамическую память, выделенную для структур базы данных.
4. Функция save\_DB(), которая будет сохранять данные из динамического массива структур в указанный файл.
5. Функция load\_DB(), которая будет выгружать данные из указанного файла в динамический массив структур.

#include <iostream>

#include <cstdio>

#include <iomanip>

#include <fstream>

#include <Windows.h>

using namespace std;

struct registdate {

int day;

int mounth;

int year;

};

struct kp {

char fio[30];

registdate dmy;//вложенная структура

int money;

};

void vvod(kp \*a, int n) {//функция для ввода значений

gets\_s(a[n].fio);

cin >> a[n].dmy.day;

cin >> a[n].dmy.mounth;

cin >> a[n].dmy.year;

cin >> a[n].money;

cout << "данные введены" << endl;

}

void polez(kp \*a, int i) {//функция для вывода значений в формате поле-значение

cout << "ФИО ";

printf("%s", a[i].fio);

cout << "\n дата регистрации - ";

printf("%02d", a[i].dmy.day);

cout << ".";

printf("%02d", a[i].dmy.mounth);

cout << ".";

printf("%d", a[i].dmy.year);

cout << "\n деньги - ";

cout << a[i].money;

}

void vivodtb(kp \*a, int n) {//функция для вывода значений в формате таблицы

printf("%20s %20s %10s\n", "ФИО :", "Дата регистрации :", "Деньги :");

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

printf("%20s %10i%s%i%s%i %10i\n", a[i].fio, a[i].dmy.day, ".", a[i].dmy.mounth, ".", a[i].dmy.year, a[i].money);

}

}

void new\_DB()

{

char fname[80];

cout << "Название создаваемого файла: " << endl;

cin >> fname; //пишется название файла

ofstream fout(fname, ios\_base::out | ios\_base::trunc); //создаётся файла

if (!fout.is\_open()) {

cout << "Файл не может быть открыт или создан\n";

}

}

void remove\_DB()

{

char fname[80];

cout << "Название файла для удаления: " << endl;

cin >> fname;

if (remove(fname)) //удаление файла

{

cout << "Ошибка удаления файла" << endl;

}

}

void save\_DB(kp \*a, const int n)

{

char fname[80];

cout << "Название файла для открытия: " << endl;

cin >> fname;

ofstream fout(fname, ios\_base::app);

if (!fout.is\_open()) {

cout << "Файл не может быть открыт\n";

}

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

fout << a[i].fio << endl;

fout << a[i].dmy.day << endl;

fout << a[i].dmy.mounth << endl;

fout << a[i].dmy.year << endl;

fout << a[i].money << endl << endl;

}

fout.close();

}

void clear(kp\* a, int& n) {

delete[] a;

n = 0;

}

kp\* load\_DB(kp \*a, int& n)

{

char fname[80];

cout << "Введите имя файла: ";

cin >> fname;

int AmountInFile;

cout << "Введите кол-во загружаемых данных: ";

cin >> AmountInFile;

ifstream file;

file.open(fname);

a = new kp[1];

a = (kp\*)realloc(a, (AmountInFile \* sizeof(kp)));// работа с памятью

for (int i = n; i < n + AmountInFile; i++) {

file >> a[i].fio;

file >> a[i].dmy.day;

file >> a[i].dmy.mounth;

file >> a[i].dmy.year;

file >> a[i].money;

}// загрузка данных

n += AmountInFile;

file.close();

return a;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

kp \*a;

int n = 0, i = 1, o = 0;

a = (kp\*)malloc(sizeof(kp));//выделение памяти

cout << "1. ввод значений" << endl;

cout << "2. вывод в виде поле-значение" << endl;

cout << "3. вывод в виде таблицы" << endl;

cout << "4. Создать файл..." << endl;

cout << "5. Удалить файл..." << endl;

cout << "6. Очистить динамическую память..." << endl;

cout << "7. Сохранить данные из динам.памяти в файл..." << endl;

cout << "8. Выгрузить данные из файла в динам.память..." << endl;

cout << "другая цифра-завершение" << endl;

cin >> i;

while (i >= 1 && i <= 8) {

if (i == 1) {

if (n == 0) {

n++;

cin.ignore();//игнорируется последний знак ввода

vvod(a, n - 1);//вызов функции ввода

}

else {

n++;

cin.ignore();//игнорируется последний знак ввода

a = (kp\*)realloc(a, (n \* sizeof(kp)));//перераспределение памяти

vvod(a, n - 1); //вызов функции ввода

}

}

else if (i == 2) {

cout << "введите номер поля ";

cin >> o;

polez(a, o - 1);//вывод нужного поля значений

}

else if (i == 3) {

vivodtb(a, n);//вывод всех значений

}

else if (i == 4) {

new\_DB();//создаётся новый файл

}

else if (i == 5) {

remove\_DB();;//удаление файла

}

else if (i == 6) {

clear(a, n);//очистка памяти

}

else if (i == 7) {

save\_DB(a, n);//запись в файл

}

else if (i == 8) {

a = load\_DB(a, n);//выгрузка данных из файла

}

cin >> i;

}

}

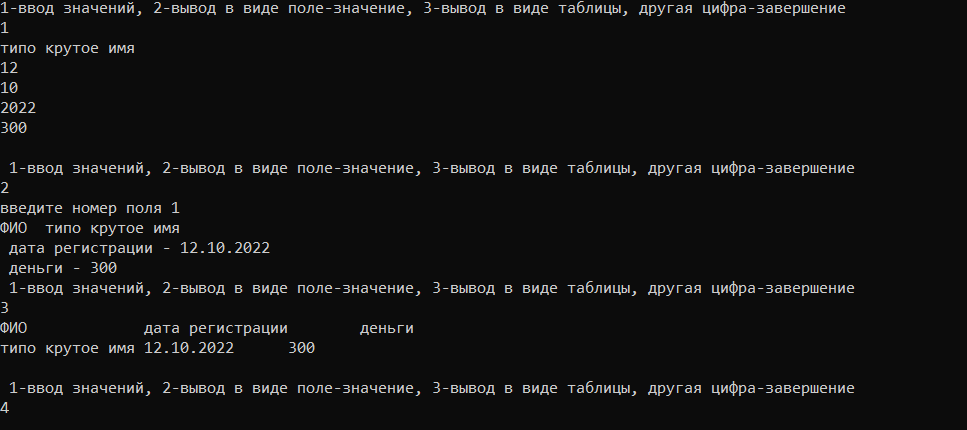
****

Рисунок 1 – Результат работы заданий

**Вывод:** в данной лабораторной работе были изучены принципы работы со файлами. Был создан файл и записаны в него данные. Память можно очищать в процессе.