Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

Лабораторная работа по теме “Структуры”

Выполнил: Студент группы РИС-23-3б

Асташин Дмитрий Алексеевич

Проверила: Доцент кафедры ИТАС

О. А. Полякова

2024

1. **Постановка задачи**

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

Структура: “Государство”:

* Название;
* Государственный язык;
* Денежная единица;
* Курс валюты относительно $;

Удалить элемент с указанным названием, добавить 2 элемента структуры в конец файла.

1. **Анализ задачи**

* Для записи структуры в файл используется открытие файла в бинарном режиме;
* Функция добавления элементов в структуру: addPersonToFile(). В функции открывается файл, создается экземпляр структуры и методом .getline() заполняются все элементы структуры. После чего структура записывается в файл методом .write();
* Функция для чтения данных из файла: readPersonsFromFile().

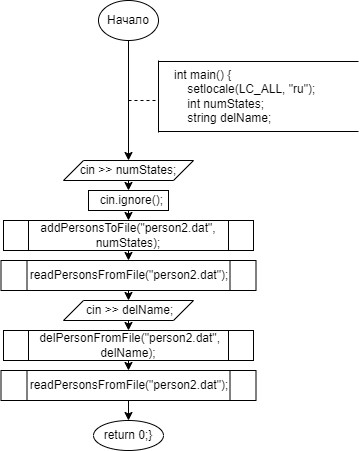
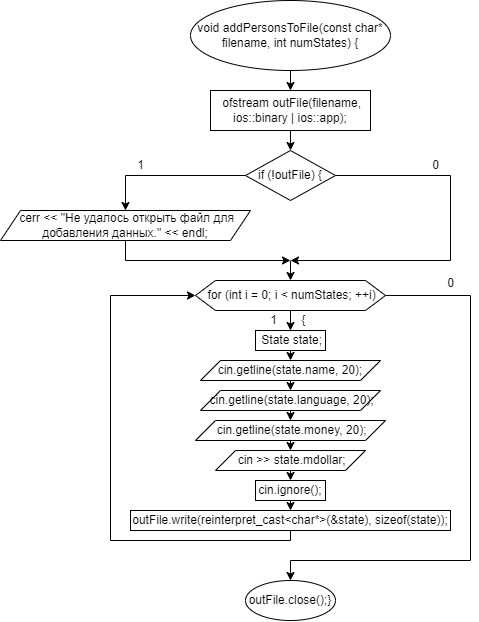
Создается экземпляр структуры.

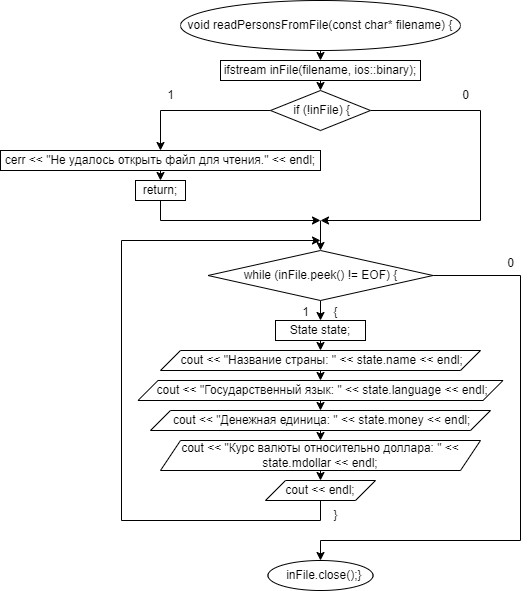
Данные из файла считываются, пока не достигнут его конец, и запсиываются в структуру с помощью метода .read(). После чего прочитанные данные выводятся на экран;

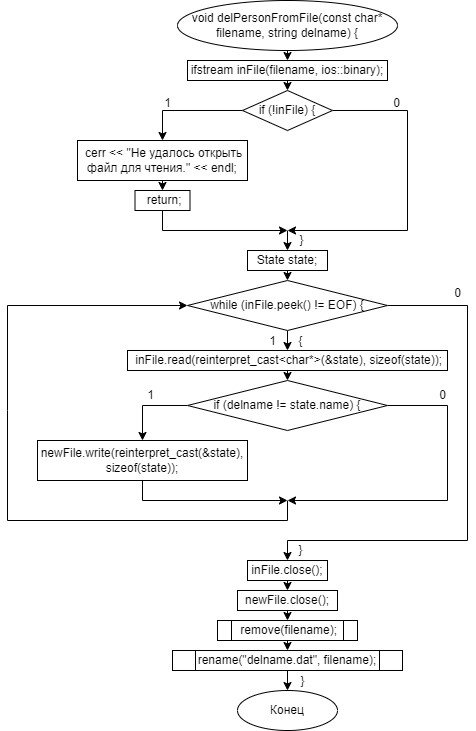
* Функция удаления экземпляров структуры из файла: delPersonsFromFile().

Удаление файла производится с помощью второго файла. Имя структуры, которая считывается из файла проверяется с введеным ключом для удаления. Если они не совпадают, то структура записывается во второй файл, иначе ничего не происходит. После проверки всех структур основной файл удаляется, а дополнительный переименовывается с помощью функции rename().

1. Блок-схема







1. Код программы

#include <iostream>

#include <fstream>

#include <clocale>

using namespace std;

// Пример структуры

struct State {

char name[20];

char language[20];

char money[20];

double mdollar;

};

// Функция для добавления элементов структуры в файл

void addPersonsToFile(const char\* filename, int numStates) {

// Открываем файл для добавления данных в бинарном режиме (режим добавления)

ofstream outFile(filename, ios::binary | ios::app);

if (!outFile) {

cerr << "Не удалось открыть файл для добавления данных." << endl;

return;

}

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

// Создаем экземпляр структуры для ввода данных

State state;

// Запрашиваем у пользователя ввод данных для каждого элемента

cout << "Введите название страны: ";

cin.getline(state.name, 20);

cout << "Введите государственный язык: ";

cin.getline(state.language, 20);

cout << "Введите денежную еденицу: ";

cin.getline(state.money, 20);

cout << "введите курс валюты относительно $: ";

cin >> state.mdollar;

// Очищаем буфер ввода перед следующей итерацией

cin.ignore();

// Записываем структуру в файл

outFile.write(reinterpret\_cast<char\*>(&state), sizeof(state));

}

// Закрываем файл

outFile.close();

cout << "Данные успешно добавлены в файл." << endl;

}

// Функция для чтения данных из файла и вывода их на экран

void readPersonsFromFile(const char\* filename) {

// Открываем файл для чтения в бинарном режиме

ifstream inFile(filename, ios::binary);

if (!inFile) {

cerr << "Не удалось открыть файл для чтения." << endl;

return;

}

// Считываем данные из файла до тех пор, пока не достигнем его конца

while (inFile.peek() != EOF) {

// Создаем экземпляр структуры для хранения данных из файла

State state;

// Читаем данные из файла в структуру

inFile.read(reinterpret\_cast<char\*>(&state), sizeof(state));

// Выводим прочитанные данные на экран

cout << "Название страны: " << state.name << endl;

cout << "Государственный язык: " << state.language << endl;

cout << "Денежная единица: " << state.money << endl;

cout << "Курс валюты относительно доллара: " << state.mdollar << endl;

cout << endl;

}

// Закрываем файл

inFile.close();

}

void delPersonFromFile(const char\* filename, string delname) {

ifstream inFile(filename, ios::binary);

if (!inFile) {

cerr << "Не удалось открыть файл для чтения." << endl;

return;

}

ofstream newFile("delname.dat", ios::binary);

if (!newFile) {

cerr << "Не удалось открыть файл для чтения." << endl;

return;

}

// Создаем экземпляр структуры для хранения данных из файла

State state;

while (inFile.peek() != EOF) {

// Читаем данные из файла в структуру

inFile.read(reinterpret\_cast<char\*>(&state), sizeof(state));

if (delname != state.name) {

newFile.write(reinterpret\_cast<char\*>(&state), sizeof(state));

}

}

inFile.close();

newFile.close();

remove(filename);

rename("delname.dat", filename);

}

int main() {

setlocale(LC\_ALL, "ru");

int numStates;

string delName;

cout << "Сколько элементов структуры вы хотите добавить в файл? ";

cin >> numStates;

// Очищаем буфер ввода перед вызовом функции

cin.ignore();

addPersonsToFile("person2.dat", numStates);

cout << endl << "Данные, сохраненные в файле:" << endl;

readPersonsFromFile("person2.dat");

cout << "Кого удаляем? ";

cin >> delName;

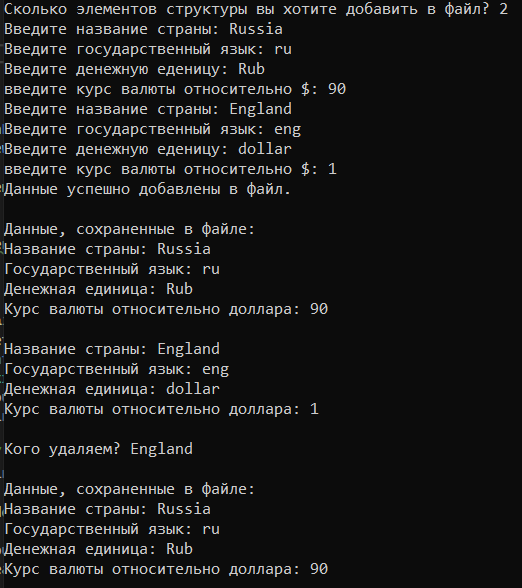
delPersonFromFile("person2.dat", delName);

cout << endl << "Данные, сохраненные в файле:" << endl;

readPersonsFromFile("person2.dat");

return 0;}

1. Результаты



1. Выводы:

Программа работает корректно: пользователем вводится нужное ему количество структур, одна из структур удаляется, также данные записываются в файл.

Таким образом, были получены навыки работы с типом Struct, а также освоена запись и чтение из файла структур.

https://github.com/DimetriusAsti/LABA\_8