***Министерство образования Республики Беларусь***

***Учреждение Образования***

***«Брестский Государственный Технический Университет»***

***Кафедра ИИТ***

**Лабораторная работа №13**

**По дисциплине ОАиП за II семестр**

**Тема: «Бинарные файлы. Файл из структур»**

**Выполнил:**

Студент 1-го курса

Группы ИИ-15(1)

Волк И. А.

**Проверила:**

Хацкевич М. В.

Брест 2018

Цель: научиться работать с бинарными файлами во взаимодействии со структурами.

**Задание 1**

Реализовать меню, разработанное с применением массива указателей на функции.

**Задание 2**

Для каждого из вариантов задания необходимо разработать следующие функции:

1. Создание файла.

2. Добавление элемента в конец файла.

3. Вывод содержимого файла на экран.

4. Удаление элемента из файла:

* элемента с заданным номером;
* элемента с заданным ключом.

5. Упорядочивание элементов в файле по выбранному признаку. Методы сортировки:

1. пузырьком;
2. перемешиванием;
3. вставками;
4. подсчетом;
5. слиянием;
6. выбором;
7. Шелла;

«Студент»:

фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год,

месяц число); номер телефона; домашний адрес (почтовый индекс, страна,область, район, город, улица, дом, квартира); ВУЗ; курс; группа; средний бал;специальность.

Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <conio.h>

#include <cmath>

#include <string>

#include <fstream>

using namespace std;

#pragma pack (push, 1)

struct fullNameStruct

{

char name[1024];

char lastName[1024];

char patronymic[1024];

};

struct dateOfBirthStruct

{

int year, month, date;

};

struct homeAddressStruct

{

char postcode[1024];

char country[1024];

char region[1024];

char district[1024];

char town[1024];

char street[1024];

char house[1024];

char apartmentNumber[1024];

};

struct highSchoolStruct

{

char name[1024];

char course[1024];

char group[1024];

int averageGrade;

char specialty[1024];

};

struct student

{

int numOfStud;

fullNameStruct fullName;

enum

{

male = 1,

female

}gender;

char nationality[1024];

double hight, wight;

dateOfBirthStruct dateOfBirth;

char phoneNumber[1024];

homeAddressStruct address;

highSchoolStruct highSchool;

};

#pragma pack (pop)

void swapStud(student&, student&);

int getMenuItem(int, int);

void writeToFile(const char\*, const char\*, int);

void error(const char\*, int);

template<typename T>

T\* increase(T\*, int&);

char \* newFile(int&);

void outputStruct(student);

void addToEnd(char \*, int &);

void fileOut(char \*, int &);

void deleteElement(char \*, int &);

void sortFile(char \*, int &);

int menu();

void(\*functions[])(char\*, int&) = {addToEnd, fileOut, deleteElement, sortFile};

int main()

{

int numOfStud = 0;

char \* nameOfFileTemp = newFile(numOfStud);

char nameOfFile[1024];

strcpy\_s(nameOfFile, sizeof(nameOfFile), nameOfFileTemp);

cin.clear();

while (1)

{

int input = menu() - 1;

if (!input)

{

nameOfFileTemp = newFile(numOfStud);

strcpy\_s(nameOfFile, sizeof(nameOfFile), nameOfFileTemp);

}

else (\*functions[input - 1])(nameOfFile, numOfStud);

}

system("pause");

return 0;

}

int menu()

{

cout << "1. New file" << endl;

cout << "2. New element" << endl;

cout << "3. View" << endl;

cout << "4. Delete element" << endl;

cout << "5. Sort file" << endl;

int input = getMenuItem(1, 5);

system("cls");

return input;

}

char \* newFile(int &numOfStud)

{

char nameOfFile[1024];

cout << "Name of new file: ";

cin >> nameOfFile;

writeToFile(nameOfFile, "", ios::out);

system("cls");

numOfStud = 0;

return nameOfFile;

}

void addToEnd(char \* nameOfFile, int &numOfStud)

{

cin.clear();

student stud; numOfStud++;

cout << "Element #" << numOfStud << endl; stud.numOfStud = numOfStud;

cout << "Name: "; cin >> stud.fullName.name;

cout << "Last name: "; cin >> stud.fullName.lastName;

cout << "Patronymic: "; cin >> stud.fullName.patronymic;

int gender;

cout << "Gender(1 - male, 2 - female): "; cin >> gender;

(gender == 1) ?

(stud.gender = student::male)

: (stud.gender = student::female);

cout << "Nationality: "; cin >> stud.nationality;

cout << "Hight: "; cin >> stud.hight;

cout << "Wight: "; cin >> stud.wight;

cout << "Year of birth: "; cin >> stud.dateOfBirth.year;

cout << "Month of birth: "; cin >> stud.dateOfBirth.month;

cout << "Date of birth: "; cin >> stud.dateOfBirth.date;

cout << "Phone number: "; cin >> stud.phoneNumber;

cout << "Postcode: "; cin >> stud.address.postcode;

cout << "Country: "; cin >> stud.address.country;

cout << "Region: "; cin >> stud.address.region;

cout << "District: "; cin >> stud.address.district;

cout << "Town: "; cin >> stud.address.town;

cout << "Street: "; cin >> stud.address.street;

cout << "House: "; cin >> stud.address.house;

cout << "Apartment number: "; cin >> stud.address.apartmentNumber;

cout << "#High school#\nName: "; cin >> stud.highSchool.name;

cout << "Course: "; cin >> stud.highSchool.course;

cout << "Group: "; cin >> stud.highSchool.group;

cout << "Average grade: "; cin >> stud.highSchool.averageGrade;

cout << "Specialty: "; cin >> stud.highSchool.specialty;

system("cls");

ofstream fout;

fout.open(nameOfFile, ios::out | ios::binary | ios::app);

if (!fout)

{

error("Cannot open the file!", 500);

return;

}

fout.write((char\*)&stud, sizeof(stud));

}

void fileOut(char \* nameOfFile, int &numOfStud)

{

if (numOfStud == 0)

return;

ifstream fin;

fin.open(nameOfFile, ios::binary);

if (!fin)

{

error("Cannot open the file!", 500);

return;

}

student stud;

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

{

fin.read((char\*)&stud, sizeof(stud));

outputStruct(stud);

cout << endl << endl;

}

system("pause");

system("cls");

}

void deleteElement(char \* nameOfFile, int &numOfStud)

{

if (numOfStud == 0)

return;

int num;

cout << "Number of student: "; cin >> num;

system("cls");

ifstream fin;

fin.open(nameOfFile, ios::binary);

if (!fin)

{

error("Cannot open the file!", 500);

return;

}

ofstream fout;

fout.open("temp.txt", ios::out | ios::binary | ios::app);

if (!fout)

{

error("Cannot open the file!", 500);

return;

}

student stud;

bool deleted = false;

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

{

fin.read((char\*)&stud, sizeof(stud));

if (stud.numOfStud == num)

{

deleted = true;

i--; numOfStud--;

continue;

}

if (deleted)

stud.numOfStud--;

fout.write((char\*)&stud, sizeof(stud));

}

fout.close();

fin.close();

if (!remove(nameOfFile))

rename("temp.txt", nameOfFile);

else

error("Error!", 500);

}

void sortFile(char \* nameOfFile, int &numOfStud)

{

cout << "1. Last name" << endl;

cout << "2. Year of birth" << endl;

int input = getMenuItem(1, 2);

system("cls");

if (!input) return;

ifstream fin;

fin.open(nameOfFile, ios::binary);

if (!fin)

{

error("Cannot open the file!", 500);

return;

}

student \*stud = new student[numOfStud];

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

fin.read((char\*)&(stud[i]), sizeof(stud[i]));

int right = numOfStud - 1;

int left = 0;

while (left<right && numOfStud>1)

{

for (int i = left; i < right; i++)

{

bool forIf = (input-1) ? (stud[i].dateOfBirth.year < stud[i + 1].dateOfBirth.year)

:(stud[i].fullName.lastName[0] > stud[i + 1].fullName.lastName[0]);

if (forIf)

swapStud(stud[i], stud[i + 1]);

}

right--;

for (int i = right; i > left; i--)

if (stud[i].fullName.lastName[0] > stud[i + 1].fullName.lastName[0])

swapStud(stud[i], stud[i + 1]);

left++;

}

writeToFile(nameOfFile, "", ios::out);

ofstream fout;

fout.open(nameOfFile, ios::out | ios::binary | ios::app);

if (!fout)

{

error("Cannot open the file!", 500);

return;

}

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

{

fout.write((char\*)&(stud[i]), sizeof(stud[i]));

}

}

void swapStud(student& a, student& b)

{

student temp = a;

a = b;

b = temp;

int numTemp = a.numOfStud;

a.numOfStud = b.numOfStud;

b.numOfStud = numTemp;

}

void outputStruct(student stud)

{

cout << "Struct #" << stud.numOfStud << endl;

cout << "Name: " << stud.fullName.name << endl;

cout << "Last name: " << stud.fullName.lastName << endl;

cout << "Patronymic: " << stud.fullName.patronymic << endl;

cout << "Gender: " << stud.gender << endl;

cout << "Nationality: " << stud.nationality << endl;

cout << "Hight: " << stud.hight << endl;

cout << "Wight: " << stud.wight << endl;

cout << "Year of birth: " << stud.dateOfBirth.year << endl;

cout << "Month of birth: " << stud.dateOfBirth.month << endl;

cout << "Date of birth: " << stud.dateOfBirth.date << endl;

cout << "Phone number: " << stud.phoneNumber << endl;

cout << "Postcode: " << stud.address.postcode << endl;

cout << "Country: " << stud.address.country << endl;

cout << "Region: " << stud.address.region << endl;

cout << "District: " << stud.address.district << endl;

cout << "Town: " << stud.address.town << endl;

cout << "Street: " << stud.address.street << endl;

cout << "House: " << stud.address.house << endl;

cout << "Apartment number: " << stud.address.apartmentNumber << endl;

cout << "#High school#\nName: " << stud.highSchool.name << endl;

cout << "Course: " << stud.highSchool.course << endl;

cout << "Group: " << stud.highSchool.group << endl;

cout << "Average grade: " << stud.highSchool.averageGrade << endl;

cout << "Specialty: " << stud.highSchool.specialty;

}

void error(const char\* content, int timer)

{

system("cls");

cout << content << endl;

Sleep(timer);

system("cls");

}

void writeToFile(const char\* nameOfFile, const char\* content, int mode)

{

ofstream fout;

fout.open(nameOfFile, mode);

if (!fout)

{

error("Cannot open the file!", 500);

return;

}

fout << content;

fout.close();

}

template<typename T>

T\* increase(T\* arr, int &size)

{

size++;

int sizeVar = size;

T\* arrRes = new T[sizeVar];

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

arrRes[i] = arr[i];

return arrRes;

}

int getMenuItem(int a, int b)

{

int res;

while (1)

{

res = \_getch() - 48;

if (res >= a&&res <= b)

break;

if ((res + 48) == 27)

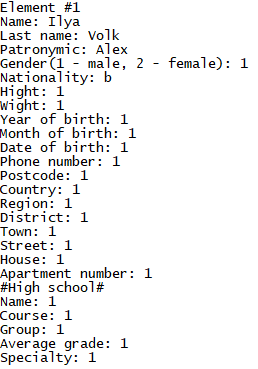
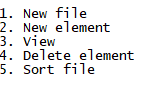
return 0;

}

return res;

}

Результат выполнения:



Удаление:



Сортировка:



**Задание 3**

Сформировать бинарный файл, содержимым которого являются вещественные матрицы (структуры матриц).

При этом количество матриц (компонент структуры) и их размерность вводится в процессе выполнения программы с клавиатуры. В соответствии с индивидуальным заданием обработать содержимое полученного файла. Для каждого из вариантов задания вывести содержимое исходного файла на экран до и после преобразования. При работе с матри­цами, где это возможно, использовать динамические мас­сивы.

Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <fstream>

#include <cmath>

#include <string>

using namespace std;

bool isSym(int \*\*, int);

int main()

{

int numOfMatrix;

cout << "Number of matrix: "; cin >> numOfMatrix;

system("cls");

int n;

cout << "(matrix NxN) \nN: "; cin >> n;

system("cls");

ofstream fout("initial.txt", ios::out);

fout << "";

fout.close();

fout.open("initial.txt", ios::out | ios::app | ios::binary);

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

{

int \*\* matrix = new int\*[n];

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

{

matrix[j] = new int[n];

for (int z = 0; z < n; z++)

{

matrix[j][z] = rand() % 10;

fout.write((char\*)&matrix[j][z], sizeof(int));

}

}

}

fout.close();

ofstream res1("res1.txt", ios::out);

res1 << "";

res1.close();

res1.open("res1.txt", ios::out | ios::app | ios::binary);

ofstream res2("res2.txt", ios::out);

res2 << "";

res2.close();

res2.open("res2.txt", ios::out | ios::app | ios::binary);

ifstream fin("initial.txt", ios::binary);

int num1 = 0, num2 = 0;

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

{

int \*\* matrix = new int\*[n];

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

{

matrix[j] = new int[n];

for (int z = 0; z < n; z++)

fin.read((char\*)&matrix[j][z], sizeof(int));

}

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

{

for (int z = 0; z < n; z++)

cout << matrix[j][z] << " ";

cout << endl;

}

cout << endl;

if (isSym(matrix, n))

{

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

for (int z = 0; z < n; z++)

res1.write((char\*)&matrix[j][z], sizeof(int));

num1++;

}

else

{

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

for (int z = 0; z < n; z++)

res2.write((char\*)&matrix[j][z], sizeof(int));

num2++;

}

}

res2.close();

res1.close();

ifstream res1in("res1.txt", ios::binary);

ifstream res2in("res2.txt", ios::binary);

cout << "Not sym: " << endl;

for (int i = 0; i < num2; i++)

{

int \*\* matrix = new int\*[n];

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

{

matrix[j] = new int[n];

for (int z = 0; z < n; z++)

res2in.read((char\*)&matrix[j][z], sizeof(int));

}

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

{

for (int z = 0; z < n; z++)

cout << matrix[j][z] << " ";

cout << endl;

}

cout << endl;

}

cout << "Sym: " << endl;

for (int i = 0; i < num1; i++)

{

int \*\* matrix = new int\*[n];

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

{

matrix[j] = new int[n];

for (int z = 0; z < n; z++)

res1in.read((char\*)&matrix[j][z], sizeof(int));

}

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

{

for (int z = 0; z < n; z++)

cout << matrix[j][z] << " ";

cout << endl;

}

cout << endl;

}

system("pause");

return 0;

}

bool isSym(int \*\* matrix, int size)

{

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

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

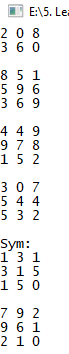
if (matrix[i][j] != matrix[j][i])

return false;

return true;

}

Результат выполнения:



Вывод: научился работать с бинарными файлами во взаимодействии со структурами.

isSym:

