**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу**

**Кафедра системного проектування**

**Звіт**

**про виконання лабораторної роботи №14**

**з дисципліни «Алгоритмізація та програмування»**

Виконав:  
студент I курсу, групи ДА-02

Рудік Андрій Миколайович  
Прийняв:

к.т.н., доцент Безносик О. Ю.

Київ – 2021

Завдання

Створити масив структур. Кожна структура складається з таких елементів: інститут, факультет. Для інституту задати назву та план прийому на перший курс. Для факультету задати список спеціальностей та прохідний бал. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) скласти список спеціальностей по факультетах із вказаним прохідним балом; 2) впорядкувати масив за прохідним балом і планом прийому; 3) визначити інститут із вказаним середнім прохідним балом.

Код програми

main.cpp

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <clocale>

#include <fstream>

#include <windows.h>

#include "LinkedList.h"

#include "Faculty.h"

#include "Speciality.h"

#include "Institute.h"

#include "InstitutesList.h"

using namespace std;

int main()

{

srand(time(NULL));

setlocale(LC\_ALL, "rus");

InstitutesList institutesList("data.dat");

while (true) {

system("cls");

cin.clear();

fflush(stdin);

cout << "Главное меню:" << endl

<< "1) Вывести список университетов" << endl

<< "2) Отсортировать по проходному баллу" << endl

<< "3) Найти специальности по проходному баллу" << endl

<< "4) Найти институт по проходному баллу" << endl

<< "5) Добавить случайный институт" << endl

<< "6) Удалить интститут" << endl

<< "7) Выход" << endl << endl;

int key;

cin >> key;

system("cls");

cin.clear();

fflush(stdin);

switch (key) {

case 1: {

cout << institutesList << endl;

break;

}

case 2: {

institutesList.sortByPassingScore();

cout << "Готово" << endl;

break;

}

case 3: {

cout << "Введите промежуток на котором стоит искать специальности в формате \"от до\"" << endl;

float from, to;

cin >> from >> to;

cout << endl;

cout << institutesList.getSpecialitiesInRange(from, to);

break;

}

case 4: {

cout << "Введите средний проходной балл по институту чтобы найти подходящий" << endl;

float passingScore;

cin >> passingScore;

cout << endl;

try {

cout << institutesList.findByPassingScore(passingScore);

}

catch (const char\* e) {

cout << e;

}

break;

}

case 5: {

institutesList.append(Institute());

cout << "Готово" << endl;

break;

}

case 6: {

cout << "Введите номер института который желаете удалить" << endl;

int idx;

cin >> idx;

try {

institutesList.pop(idx - 1);

cout << "Готово" << endl;

}

catch (const char\* e) {

cout << e << endl;

}

break;

}

case 7: {

return 0;

break;

}

default: {

cout << endl << "Попробуйте еще раз" << endl;

break;

}

}

system("pause");

}

return 0;

}

stringtoBinary.h

#pragma once

#include <cstring>

#include <fstream>

using namespace std;

void swrite(ofstream& fout, const string& str);

void sread(ifstream& fin, string& str);

stringtoBinary.cpp

#include "stringToBinary.h"

void swrite(ofstream& fout, const string& str)

{

size\_t length = str.length() + 1;

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

fout.write((char\*)str.c\_str(), length);

}

void sread(ifstream& fin, string& str)

{

size\_t length;

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

char\* buffer = new char[length];

fin.read(buffer, length);

str = buffer;

delete[] buffer;

}

Speciality.h

#pragma once

#include <string>

#include "LinkedList.h"

struct Speciality

{

uint16\_t code;

std::string name;

Speciality();

Speciality(const uint16\_t&, const std::string&);

void write(std::ofstream& fout) const;

void read(std::ifstream& fin);

friend std::ostream& operator<< (std::ostream&, const Speciality&);

friend std::ostream& operator<< (std::ostream&, const LinkedList<std::pair<Speciality, float>>&);

};

Speciality.cpp

#include "Speciality.h"

#include <iostream>

#include <iomanip>

#include <fstream>

#include "stringToBinary.h"

#define NUMBER\_OF\_SPECIALITIES 12

using namespace std;

const pair<uint16\_t, string> SPECIALITIES[NUMBER\_OF\_SPECIALITIES] = {

{11, "Науки про освiту"},

{12, "Дошкiльна освiта"},

{13, "Початкова освiта"},

{14, "Середня освiта"},

{15, "Професiйна освiта"},

{16, "Спецiальна освiта"},

{17, "Фiзична культура i спорт"},

{121, "Iнженерiя програмного забезпечення"},

{122, "Комп’ютернi науки та iнформацiйнi технологiї"},

{123, "Комп’ютерна iнженерiя"},

{124, "Системний аналiз"},

{125, "Кiбербезпека"},

};

Speciality::Speciality() {

const pair<uint16\_t, string> p = SPECIALITIES[rand() % NUMBER\_OF\_SPECIALITIES];

code = p.first;

name = p.second;

}

Speciality::Speciality(const uint16\_t& acode, const string& aname) {

code = acode;

name = aname;

}

void Speciality::write(ofstream& fout) const

{

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

swrite(fout, name);

}

void Speciality::read(std::ifstream& fin)

{

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

sread(fin, name);

}

ostream& operator<<(ostream &out, const Speciality &spec) {

out << setw(3) << setfill('0') << spec.code << " " << spec.name;

return out;

}

ostream& operator<<(ostream& out, const LinkedList<pair<Speciality, float>>& specs)

{

int maxNameWidth = 0;

for (const pair<Speciality, float>& spec : specs) {

if (spec.first.name.size() > maxNameWidth) maxNameWidth = spec.first.name.size();

}

for (const pair<Speciality, float>& spec : specs) {

out << " " << setw(3) << setfill('0') << spec.first.code << " ";

out << setw(maxNameWidth) << setfill(' ') << left << spec.first.name

<< " : " << spec.second << endl;

out << right;

}

return out;

}

Результат роботи програми наведено на рисунку 1

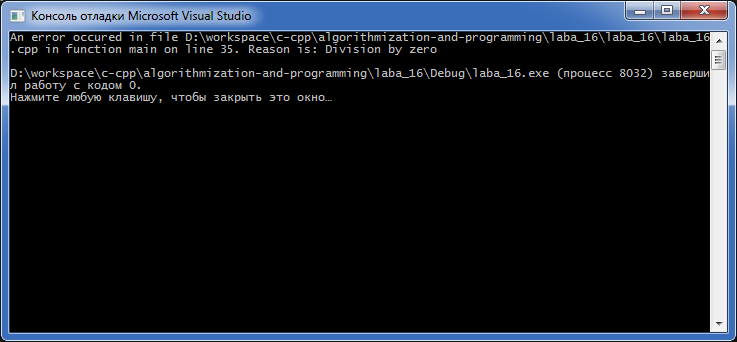


Рисунок 1 – Результат виконання програми

Висновки

В ході виконання лабораторної роботи №16 я вивчив особливості використання директив препроцесора користуючись мовою програмування С++.