Фаховий коледж ракетно-космічного машинобудування

Дніпровського національного університету імені Олеся Гончара

ЗВІТ

з лабораторних робіт

з дисципліни «Технології (Програмування)»

Спеціальність 123 Комп’ютерна інженерія

Група

Виконав І.Д. Колосов

Перевірив М.М. Гапоненко

2022-2023

ЗМІСТ

Лабораторна робота № 16

ЛАБОРАТОРНА РОБОТА № 16

Тема: Складання та налагодження лінійних програм.

Мета: Придбати практичні навички при написанні та налагодженні програм лінійних алгоритмів на мові програмування Сі\С++.

Хід роботи

1 Постановка задачі.

Загальна постановка завдання:

Створити та реалізувати обробку файлу структур даних згідно Вашого варіанта. Програма повинна задовольняти наступним вимогам:

− повинні бути реалізовані наступні функції обробки файлу структур:

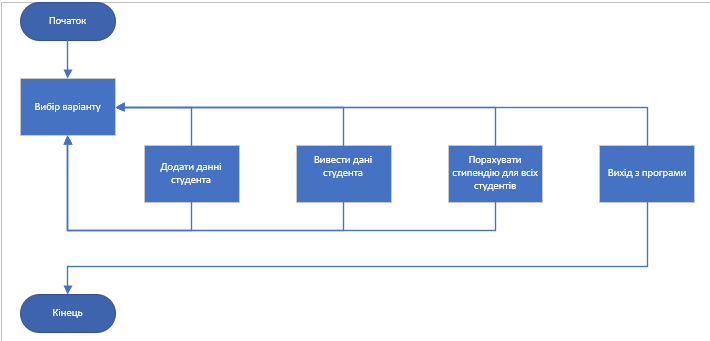
1. перегляд вмісту файлу;
2. додавання нового запису в файл;
3. реалізація запитів до файлу згідно з індивідуальними завданнями Вашого варіанту.

* організувати меню для вибору користувачем варіанту виконання вказаних вище функцій. Забезпечити повернення в меню програми після виконання чергового пункту меню. Вихід із програми здійснювати по відповідному пункту меню: «Вихід з програми»;
* при обробці файлу враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;

− введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

|  |  |  |
| --- | --- | --- |
| 7 | Скласти програму, що містить відомості про здачу студентами 3 курсу відділення 1 сесії.  Структура повинна складатися з полів:   * Група; * ПІБ студента; * Відмітки по 2 іспитам; * Відмітки по 3 залікам; * Ознака участі в суспільній праці (активна \ неактивна); * База навчання (бюджет/ контракт) | Видати відомість нарахування студентам- бюджетникам групи Х стипендії. Студент, що отримав всі відмітки «5» і активно брав участь в суспільній праці, зараховується на підвищену стипендію - доплата 50%, неактивно брав участь – доплата 25%. Студенти, що отримали «4» і «5», зараховуються на звичайну стипендію. Студент, що отримав одну оцінку «3», але активно займався суспільною працею, також зараховується на звичайну стипендію, інакше зачислення не буде.  Х та розмір звичайної стипендії – вводити з клавіатури. |

1. Блок-схема алгоритму рішення задачі.



1. Текст програми мовою програмування С++.

#include <iostream>

#include <string.h>  *// substr, getline*

#include <windows.h> *// encoding*

#include <vector>   *// vectors, push\_back*

#include <ctype.h> *// strcpy(), c\_str*

#include <fstream> *// files, is\_opem, .close(), .open(), ios::in, ios::out*

#include <filesystem> *// current\_path() directory\_iterator()*

using *namespace* std;

*class* Read {

*protected:*

*// returns vector of all strigs in file*

*// vector <string> fileReadAll(string filename)*

        vector <string> fileReadAll(string *filename*) {

            fstream file\_read;

            string line;

            vector <string> fileStrings;

            file\_read.open(*filename*, ios::in);

            if (file\_read.is\_open()) {

                while (getline(file\_read, line)) {

                    fileStrings.push\_back(line);

                }

                file\_read.close();

            }

            return fileStrings;

        }

*// reads specific line in vector, if line not found returns "not found"*

*// string vectorReadLine(vector <string> fileStrings, int lineNo)*

        string vectorReadLine(vector <string> *fileStrings*, *int* *lineNo*) {

*int* lineCounter = 0;

            string outLine = "not found";

            for (*auto* i: *fileStrings*) {

                if (lineCounter == *lineNo*) {outLine = i; break;}

                lineCounter++;

            }

            return outLine;

        }

};

*class* StudentClass: *protected* Read {

*protected:*

*// string name(int studentNo)*

        string name(*int* *studentNo*) {

            vector <string> vecData;

            vecData = fileReadAll("Student" + to\_string(*studentNo*) + ".txt");

            return vectorReadLine(vecData, 0).substr(5);

        }

*// string group(int studentNo)*

        string group(*int* *studentNo*) {

            vector <string> vecData;

            vecData = fileReadAll("Student" + to\_string(*studentNo*) + ".txt");

            return vectorReadLine(vecData, 1).substr(6);

        }

*// int \* gradesExam(int studentNo)*

*int* *\** gradesExam(*int* *studentNo*) {

            vector <string> vecData;

*static* *int* grades[2];

            string line;

            vecData = fileReadAll("Student" + to\_string(*studentNo*) + ".txt");

            line = vectorReadLine(vecData, 2).substr(11);

            grades[0] = stoi(line.substr(0, 1));

            grades[1] = stoi(line.substr(1, 2));

            return grades;

        }

*// int \* gradesTest(int studentNo)*

*int* *\** gradesTest(*int* *studentNo*) {

            vector <string> vecData;

*static* *int* grades[3];

            string line;

            vecData = fileReadAll("Student" + to\_string(*studentNo*) + ".txt");

            line = vectorReadLine(vecData, 3).substr(11);

            grades[0] = stoi(line.substr(0, 1));

            grades[1] = stoi(line.substr(1, 1));

            grades[2] = stoi(line.substr(2, 3));

            return grades;

        }

*// int socWork(int studentNo)*

*int* socWork(*int* *studentNo*) {

            vector <string> vecData;

            vecData = fileReadAll("Student" + to\_string(*studentNo*) + ".txt");

            return stoi(vectorReadLine(vecData, 4).substr(8));

        }

*// int studBase(int studentNo)*

*int* studBase(*int* *studentNo*) {

            vector <string> vecData;

            vecData = fileReadAll("Student" + to\_string(*studentNo*) + ".txt");

            return stoi(vectorReadLine(vecData, 5).substr(9));

        }

*// vector <int> listOfStudents()*

*// returns vector with students numbers*

        vector <*int*> listOfStudents() {

            vector <*int*> listOfStudents;

*char* check[1];

*// takes current path (path to main.cpp)*

            string path = filesystem::current\_path().u8string();

*// iterates trought all files in directory*

            for (*const* *auto* & entry : filesystem::directory\_iterator(path)) {

                string strPath = entry.path().u8string();

*int* i = strPath.find("Student" );

                if (i > 0) {

*// if finds student in path, writes it number to vector*

                    strPath = strPath.substr(i+7);

*// string to char convert*

                    strcpy(check, strPath.substr(1,1).c\_str());

*// if second symbol (after Student0) is digit than student number is 00 else 0*

                    if (isdigit(check[0]))

                        listOfStudents.push\_back(stoi(strPath.substr(0,2)));

                    else

                        listOfStudents.push\_back(stoi(strPath.substr(0,1)));

                }

            }

            return listOfStudents;

        }

*public:*

*// Виводить данні студента*

*void* printData(*int* *studentNo*) {

*int* flag = 0;

            for (*auto* i: listOfStudents())

                if (i == *studentNo*) {

                    flag = 1;

                    break;

            }

            if (flag) {

                cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl

                << "|Ім'я: " << name(*studentNo*) << endl

                << "|Группа: " << group(*studentNo*)<< endl

                << "|Оцінки за іспити: "<< gradesExam(*studentNo*)[0] << " " << gradesExam(*studentNo*)[1] << endl

                << "|Оцінки за залік: "<< gradesTest(*studentNo*)[0] << " " << gradesTest(*studentNo*)[1] << " " << gradesTest(*studentNo*)[2] << endl;

                if (socWork(*studentNo*) == 0)

                    cout << "|Соціальна робота: Активна" << endl;

                else

                    cout << "|Соціальна робота: Неактивна" << endl;

                if (studBase(*studentNo*) == 0)

                    cout << "|База навчання: Бюджет" << endl;

                else

                    cout << "|База навчання: Контракт" << endl;

                cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

            }

            else cout << endl << "Студента немає" << endl;

        }

*// Рахує бали студента та показує чи зарахований він та з якою стипендією*

*void* scholarship() {

*int* scholarship;

            if (!listOfStudents().empty()) {

                cout << "Введіть розмір базової, базованої, бази, степендії: ";

                cin >> scholarship;

                for (*auto* i: listOfStudents()) {

*int* flag3grade, flag4grade, score;

                    score = 0;

                    flag3grade = 0;

                    flag4grade = 0;

                    for (*int* c = 0; c < 2; c++) {

                        if (gradesExam(i)[c] <= 3)

                            flag3grade = 1;

                        else if (gradesExam(i)[c] == 4)

                            flag4grade = 1;

                    }

                    for (*int* v = 0; v < 3; v++) {

                        if (gradesTest(i)[v] <= 3)

                            flag3grade++;

                        else if (gradesTest(i)[v] == 4)

                            flag4grade = 1;

                    }

                    if (flag4grade == 0 && flag3grade == 0) score+= 2;

                    else if (flag4grade == 1 && flag3grade == 0) score++;

                    else if (flag3grade == 1) score--;

                    if (socWork(i) == 0) score++;

                    if (socWork(i) == 0 && flag3grade == 1) score = 1;

                    if (score == 3)

                        cout << endl << "  Студент "

                        <<  name(i) << " Зарахований та отримує степендію у розмірі "

                        << scholarship \* 1.5;

                    else if (score == 2 )

                        cout << endl << "  Студент "

                        <<  name(i) << " Зарахований та отримує степендію у розмірі "

                        << scholarship \* 1.25;

                    else if (score == 1)

                        cout << endl << "  Студент "

                        <<  name(i) << " Зарахований";

                    else if (score <= 0)

                        cout << endl << "  Студент "

                        <<  name(i) << " Не зарахований";

                }

            }

            else cout << endl << "Студентів немає, спробуйте додати" << endl;

        }

*// void enterData(int studentNo)*

*void* enterData(*int* *studentNo*) {

            fstream file\_write;

            string name, group;

*int* gradesExamArr[2], gradesTestArr[3];

*int* socWork, studBase;

*// Запис данних студента у змінні*

            cout << "Данні (ім'я, група) вводити латинськмим літерами" << endl;

            cin.sync();

            cout << "Введіть ім'я студента: ";

            getline(cin, name);

            cin.sync();

            cout << "Введіть групу студента: ";

            getline(cin, group);

            cin.sync();

            cout << "Введіть оцінки студента за іспити" << endl;

            for (*int* i = 0; i < 2; i++) {

                cout << i+1 << " оцінка: ";

                cin >> gradesExamArr[i];

            }

            cout << "Введіть оцінки студента за заліки" << endl;

            for (*int* i = 0; i < 3; i++) {

                cout << i+1 << " оцінка: ";

                cin >> gradesTestArr[i];

            }

            cout << "Введіть участь студента у суспільній роботі де" << endl

            << "0 Активна участь, 1 Неактивка участь" << endl

            << "Введіть: ";

            cin >> socWork;

            cout << "Введіть базу навчання студента, де" << endl

            << "0 Бюджет, 1 Контракт" << endl

            << "Введіть: ";

            cin >> studBase;

*// Запис данних студента зі змінних у файл*

            file\_write.open("Student" + to\_string(*studentNo*) + ".txt", ios::out);

            if (file\_write.is\_open()) {

                file\_write << "name=" + name << endl

                << "group=" + group << endl

                << "gradesExam=" + to\_string(gradesExamArr[0]) + to\_string(gradesExamArr[1]) << endl

                << "gradesTest=" + to\_string(gradesTestArr[0]) + to\_string(gradesTestArr[1]) + to\_string(gradesTestArr[2]) << endl

                << "socWork=" + to\_string(socWork) << endl

                << "studBase=" + to\_string(studBase);

                file\_write.close();

                cout << "- - Студент записаний - -" << endl;

            }

            else

                cout << endl << "Не вдалось відкрити файл" << endl;

        }

};

*void* menu() {

    StudentClass student;

*int* option = 0;

*int* studentNo;

    while (option != 4) {

        cout << endl

        << "    Menu:    " << endl

        << "   1. Подивитись данні студента" << endl

        << "   2. Записати данні студента" << endl

        << "   3. Вивести список степендії" << endl

        << "   4. Вихід з програми" << endl

        << "  Виберіть варіант з запропонованих" << endl

        << "  Варіант: ";

        cin.sync();

        cin >> option;

        switch(option) {

            case 1:

                cout << "  Введіть номер студента котрий вам потрібен: ";

                cin >> studentNo;

                student.printData(studentNo);

                break;

            case 2:

                cout << "  Введіть номер студента котрий вам потрібен: ";

                cin >> studentNo;

                student.enterData(studentNo);

                break;

            case 3:

                student.scholarship();

                break;

            case 4:

                break;

            default:

                cout << endl << "Введіть варіант котрий є в меню" << endl;

        }

        cout << endl << endl << " Натисніть enter щоб продовжити" << endl;

        cin.sync();

        cin.get();

    }

}

*int* main()

{

*// encoding*

    SetConsoleCP(65001);

    SetConsoleOutputCP(65001);

    menu();

    return 0;

}

*/\**

*class Read*

*public:*

*vector <string> fileReadAll(string filename)*

*- Reads Whole file and returns vector of strings that file contains*

*string vectorReadLine(vector <string> fileStrings, int lineNo)*

*class StudentClass: protected ReadNWrite*

*protected:*

*string name(int studentNo)*

*- returns name from file*

*string group(int studentNo)*

*- returns group from file*

*int \* gradesExam(int studentNo)*

*- returns array[2] with 2 grades from file*

*int \* gradesTest(int studentNo)*

*- returns array[3] with 3 grades from file*

*int socWork(int studentNo)*

*- returns socWork from file*

*int studBase(int studentNo)*

*- returns studBase from file*

*vector <int> listOfStudents()*

*- searches StudentXX.txt files in directory, and returns vector <int> with student numbers*

*public:*

*void printData(int studentNo)*

*- prints all student data (watch protected StudentStruct)*

*- if student file not exists outputs "Студента немає"*

*void scholarship()*

*- takes vector from listOfStudents() and prints if each student "отримує стипендію чи ні, чи незарахований"*

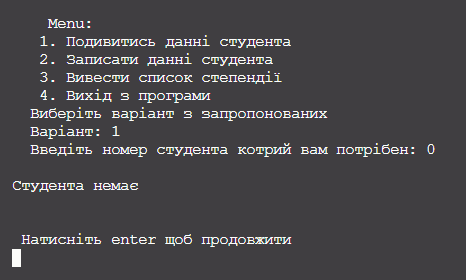
*- if vector is empty outputs "Студентів немає, спробуйте додати"*

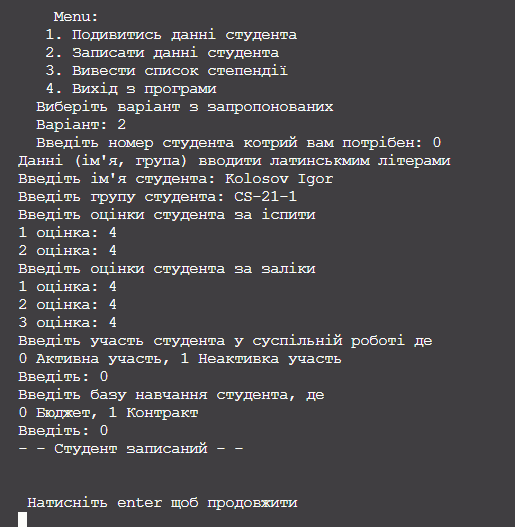
*void enterStudentData(int studentNo)*

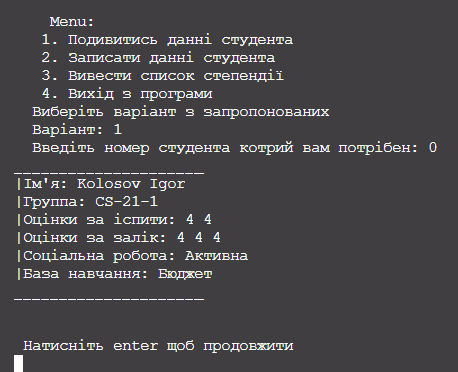
*- takes all student data from input kbd and writes it to file*

*\*/*

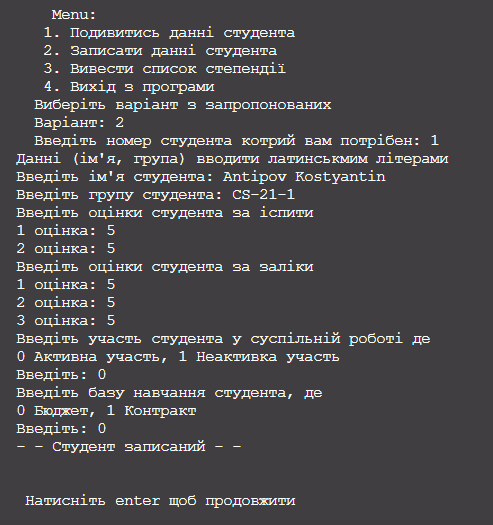
4 Копія вікна виконання програми

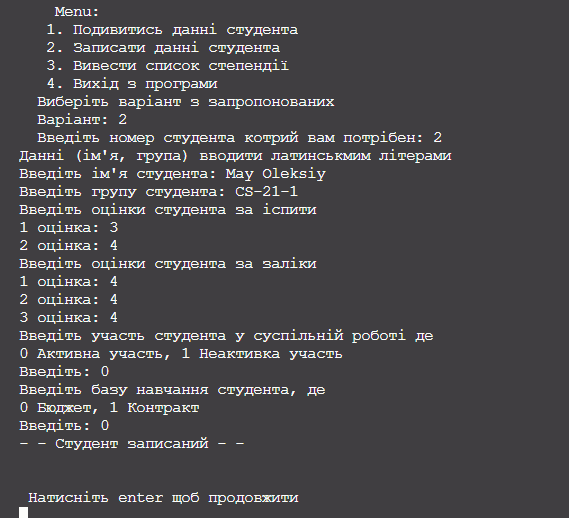


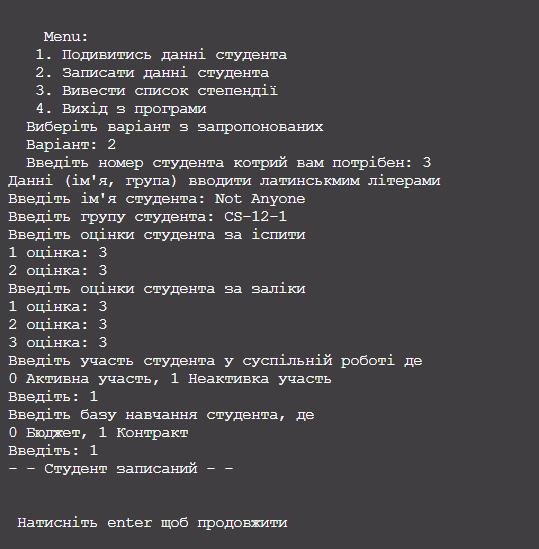


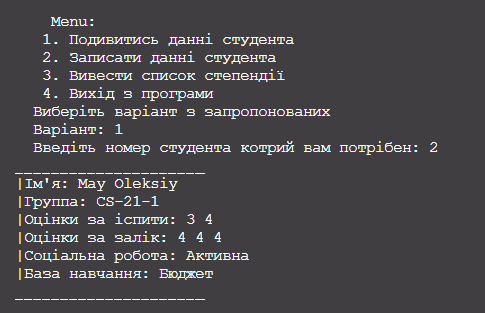


(записуємо ще декілька студентів згідно ТЗ)

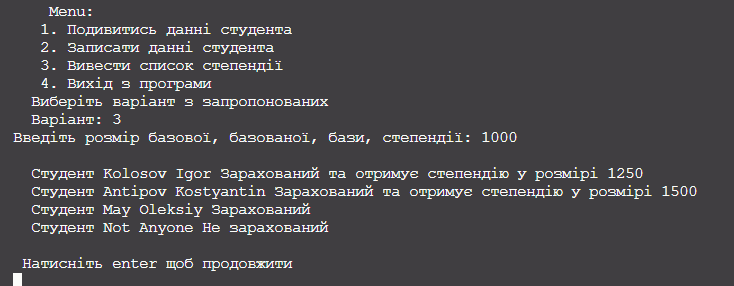


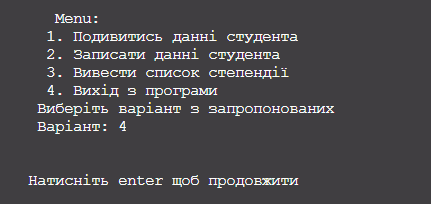












5 Висновок

В результаті виконання лабораторної роботи опрацьовано теоретичний матеріал за темами:

* Класи