Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторних робіт №3

**“ Інтегровані структури даних, запису.”**

з дисципліни "Алгоритми та структури даних ”

В-15

Виконала студентка групи СА-32

Катеринчук Тетяна Андріївна

Прийняв доцент

Щербак С.С.

*2017*

**Мета роботи:** придбання і закріплення навиків в роботі із записами, в інтеграції даних, в модульному програмуванні

**Завдання:**

Для заданої прикладної області розробити опис об'єктів цієї області. Розробити процедури, що реалізують базові операції над цими об'єктами, зокрема:

* текстове введення-виведення (консольний і файловий);
* присвоювання;
* видалення;
* редагування
* задання константних значень;
* порівняння (не менше 2-х типів);
* пошук за шаблоном.

Підготувати файл початкових даних, що містять не менше 10 значень конкретних об'єктів. В якості формату даних/файлів використовуйте CSV або XML.

Використовуючи процедури і описи модуля типу даних, розробити програму, що забезпечує введення та зчитування початкових даних з першого файлу даних в пам'ять і зберігання їх в масиві, сортування масиву по алфавітному і по числовому параметру.

**Індивідуальне завдання:**

Для кожної області перераховані параметри об'єкту. Серед параметрів обов'язково є ключове алфавітне поле (наприклад, прізвище), яке ідентифікує об'єкт, у кожного об'єкту є також одне або декілька числових полів, по яким вірогідні звернення до об'єкту. Набір характеристик може бути розширений і ускладнений по розсуду виконавця.

|  |  |  |
| --- | --- | --- |
| 15 | Відомості про сім'ю | прізвище студента, ім'я, по батькові, факультет, спеціальність батька, спеціальність матері, кількість братів і сестер |

**Текст програми**

Source.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "UserInterface.h"

#include "ActionData.h"

#include <iostream>

#include <stdio.h>

#include <string>

#include <string.h>

#include <vector>

using namespace std;

//відкриття файла

bool OpenFile(vector<vector<char>>& line, int& n) {

char fileName[61] = "2.csv";

//cout << "Enter fole name: ";

//cin.getline(fName, 61);

char element;

FILE \* fRead;

if ((fRead = fopen(fileName, "r")) == NULL) {

cerr << "Помилка!!!" << endl;

return false;

}

element = getc(fRead);

while (element != EOF) {

if (element == '\n') {

n++;

line.push\_back(vector<char>());

}

else

line[n].push\_back(element);

element = getc(fRead);

}

fclose(fRead);

return true;

}

//запис даних в файл

void WriteDataFile(vector<vector<char>> line, int n) {

FILE \* fWrite;

char fileName[61] = "2.csv";

//cout << "Enter fole name: ";

//cin.getline(fName, 61);

if ((fWrite = fopen(fileName, "w")) == NULL) {

cerr << "Помилка";

exit(1);

}

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

for (int j = 0; j < line[i].size(); j++) {

putc(line[i][j], fWrite);

}

putc('\n', fWrite);

}

fclose(fWrite);

}

int main() {

setlocale(LC\_CTYPE, "ukr");

vector<vector <char>> line;

line.push\_back(vector<char>());

int n = 0;

int menu = -1;

if (OpenFile(line, n)) {//якщо файл існує

OutputLine(line, n);//вивід даних з файла

while (menu != 0) {

cout << "1-Видалення даних про задану персону\n2-Додання даних про нову персону\n3-Сортування даних в алфавітному порядку(прізвище)\n4-Сортування даних за кількістю братів і сестер\n5-Редагування даних\n0-вихід\n";

cin >> menu;

switch (menu){

case 1:

DeletePerson(line, n);

break;

case 2:

AddPerson(line, n);

break;

case 3:

SortAlphabet(line, n);

break;

case 4:

SortNumber(line, n);

break;

case 5:

ChangeData(line, n);

break;

case 0:

exit(0);

default:

cout << "Помилка"<<endl;

break;

}

OutputLine(line, n);

WriteDataFile(line, n);

}

}

system("pause");

return 0;

}

UserInterface.h

#pragma once

#include <iostream>

#include <stdio.h>

#include "ActionData.h"

#include <string>

#include <vector>

using namespace std;

void ThrowData(vector<vector <char>>& line, int& n, string data);

void OutputLine(vector<vector <char>> line, int n);

void AddPerson(vector<vector <char>>& line, int& n);

void InputChange(vector<vector <char>>& line, int n, int& numberChange, string& newData1);

void OutputLine(vector<vector <char>> line, int n) {

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

cout << "Прiзвище |Iм'я |По батьковi |Факультет |Сп. батька |матерi |К-ть братiв i сестер" << endl;

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

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

for (int j = 0; j < line[i].size(); j++) {

if (line[i][j] == ';') {

cout << " |";

}

else

cout << line[i][j];

}

cout << " |" << endl;

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

}

}

void AddPerson(vector<vector <char>>& line, int& n) {

line.push\_back(vector<char>());

string data;

cout << "Введiть прiзвищe: ";

cin >> data;

ThrowData(line, n, data);

line[n].push\_back(';');

cout << "Введiть iм'я: ";

cin >> data;

ThrowData(line, n, data);

line[n].push\_back(';');

cout << "Введiть по батьковi: ";

cin >> data;

ThrowData(line, n, data);

line[n].push\_back(';');

cout << "Факультет: ";

cin >> data;

ThrowData(line, n, data);

line[n].push\_back(';');

cout << "Введiть спецiальнiсть батька: ";

cin >> data;

ThrowData(line, n, data);

line[n].push\_back(';');

cout << "Введiть спецiальнiсть матерi: ";

cin >> data;

ThrowData(line, n, data);

line[n].push\_back(';');

cout << "Введiть кiлькiсть сестер i братiв: ";

cin >> data;

ThrowData(line, n, data);

n++;

}

void InputChange(vector<vector <char>>& line, int n, int& numberChange, string& newData1) {

cout << "Якi данi ви хочете змiнити:\n1-Прiзвище\n2-Iм'я\n3-По батьковi\n4-Факультет\n5-Спецiальнiсть батька\n6-Спецiальнiсть матерi\n7-К-ть братiв i сестер\n";

cin >> numberChange;

numberChange = numberChange - 1;

cout << "Введiть новi данi: ";

cin >> newData1;

}

ActionData.h

#pragma once

#include "UserInterface.h"

#include <iostream>

#include <stdio.h>

#include <string>

#include <string.h>

#include <vector>

using namespace std;

void DeletePerson(vector<vector <char>>& line, int& n);

bool ChekingName(vector<vector <char>> line, int k);

void ExchangeLine(vector<vector <char>>& line, int k);

int SearchName(vector<vector <char>> line, int n);

void AddPerson(vector<vector <char>>& line, int& n);

void InputChange(vector<vector <char>>& line, int n, int& numberChange, string& newData1);

int SearchName(vector<vector <char>> line, int n) {

cout << "Введiть прiзвище: ";

string namePerson1;

cin >> namePerson1;

char \*namePerson = new char[namePerson1.size()];

for (int i = 0; i < namePerson1.size(); i++) {

namePerson[i] = namePerson1[i];

}

int iName = 0;

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

for (int j = 0; j < line[i].size(); j++) {

if (line[i][j] == ';') {

int nName = 0,k=0;

while (k<j || k<namePerson1.size()) {

if (namePerson[k] == line[i][k]) {

nName++;

}

k++;

}

if (nName == k) {

iName = i;

}

break;

}

}

}

free(namePerson);

return iName;

}

void DeletePerson(vector<vector <char>>& line, int& n) {

int iName = SearchName(line, n);

line[iName].clear();

for (int k = iName; k < n; k++) {

line[iName] = line[iName + 1];

}

line[n - 1].clear();

n--;

}

void SortAlphabet(vector<vector <char>>& line, int& n) {

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

for (int k = 0; k < n - i - 1; k++) {

if (ChekingName(line, k)) {

ExchangeLine(line, k);

}

}

}

}

void ExchangeLine(vector<vector <char>>& line, int k) {

char auxiliary1[100];

char auxiliary2[100];

int n1 = line[k].size(), n2 = line[k + 1].size();

for (int j = 0; j < n1; j++) {

auxiliary1[j] = line[k][j];

}

for (int j = 0; j < n2; j++) {

auxiliary2[j] = line[k + 1][j];

}

line[k].clear();

line[k + 1].clear();

for (int j = 0; j < n1; j++) {

line[k + 1].push\_back(auxiliary1[j]);

}

for (int j = 0; j < n2; j++) {

line[k].push\_back(auxiliary2[j]);

}

}

bool ChekingName(vector<vector <char>> line, int k) {

int iSymbol = 0;

while (line[k][iSymbol] != ';' || line[k + 1][iSymbol] != ';') {

if (line[k][iSymbol] > line[k + 1][iSymbol]) {

return true;

}

else if (line[k][iSymbol] == line[k + 1][iSymbol]) {

if (line[k][iSymbol + 1] > line[k + 1][iSymbol + 1]) {

return true;

}

if (line[k][iSymbol + 1] < line[k + 1][iSymbol + 1]) {

return false;

}

}

else {

return false;

}

iSymbol++;

}

return false;

}

void SortNumber(vector<vector <char>>& line, int& n) {

int \*number = new int[n];

int amount = 0;

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

for (int j = 0; j < line[i].size(); j++) {

if (line[i][j] == ';') {

amount++;

}

else if (amount == 6) {

number[i] = line[i][j] - 48;

amount = 0;

}

}

}

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

for (int j = 0; j < n - i - 1; j++) {

if (number[j] > number[j + 1]) {

ExchangeLine(line, j);

int element = number[j];

number[j] = number[j + 1];

number[j + 1] = element;

}

}

}

free(number);

}

void ThrowData(vector<vector <char>>& line, int& n, string data) {

for (int i = 0; i < data.size(); i++) {

line[n].push\_back(data[i]);

}

}

void ChangeData(vector<vector <char>>& line, int n) {

int personChenge = SearchName(line, n), numberChange, nChenge = 0;

string newData1;

InputChange(line, n, numberChange, newData1);

char \*newData = new char[newData1.size()];

for (int i = 0; i < newData1.size(); i++) {

newData[i] = newData1[i];

}

for (int j = 0; j < line[personChenge].size(); j++) {

if (line[personChenge][j] == ';') {

nChenge++;

}

else if (nChenge == numberChange) {

int n1 = line[personChenge].size();

char \* auxiliary = new char[n1];

for (int i = 0; i < n1; i++) {

auxiliary[i] = line[personChenge][i];

}

line[personChenge].clear();

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

line[personChenge].push\_back(auxiliary[i]);

}

for (int i = 0; i < newData1.size(); i++) {

line[personChenge].push\_back(newData[i]);

}

for (int i = j + 1; i < n1; i++) {

if (auxiliary[i] == ';') {

j = i;

break;

}

}

for (int i = j; i < n1; i++) {

line[personChenge].push\_back(auxiliary[i]);

}

break;

}

}

}

**Результат програми**

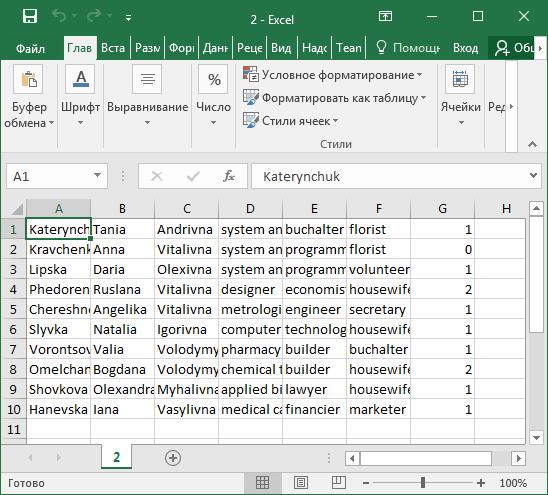


Рис.1 Початкові значення

Сортування в алфавітному порядку (за прізвищем):

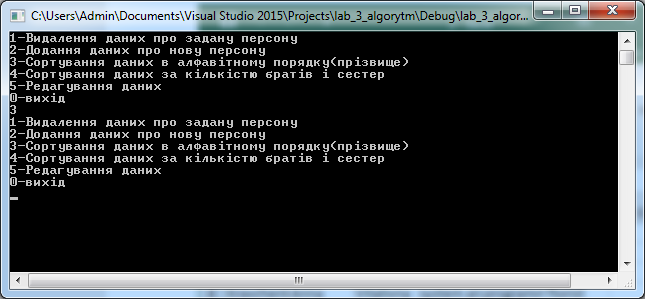
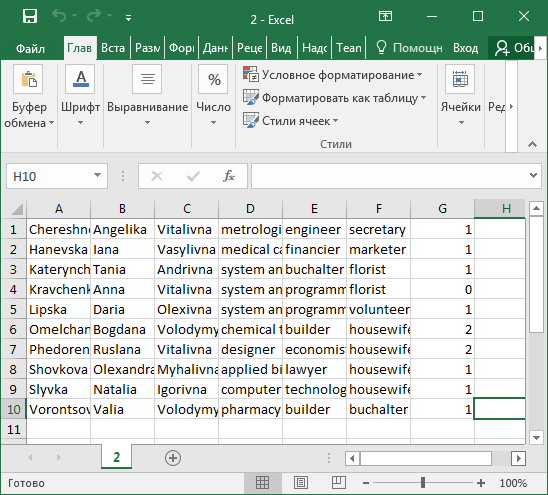
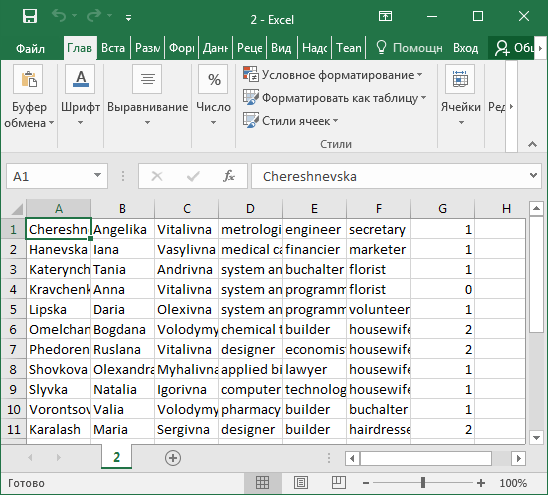


Рис. 2-3 Сортування в алфавітному порядку

Додання даних про нову персону:



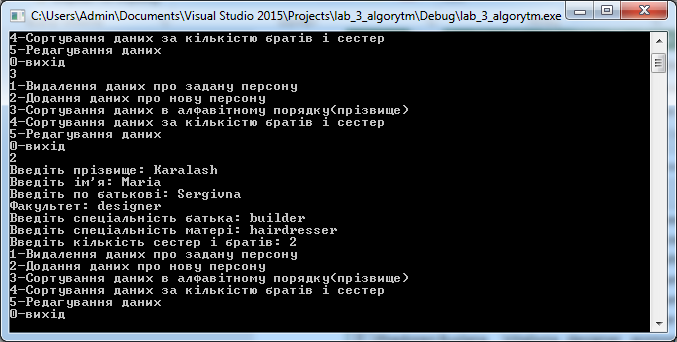


Рис. 4-5 Додання даних про нову персону

Редагування даних:

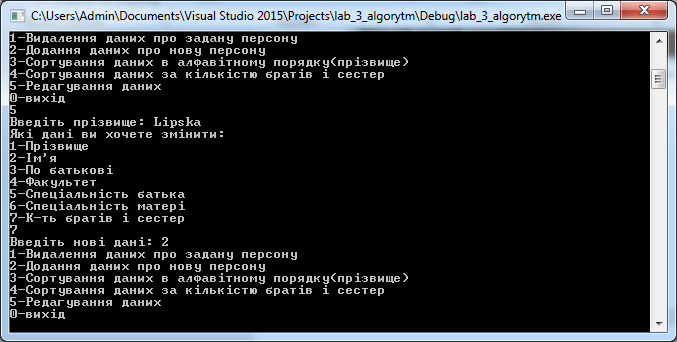
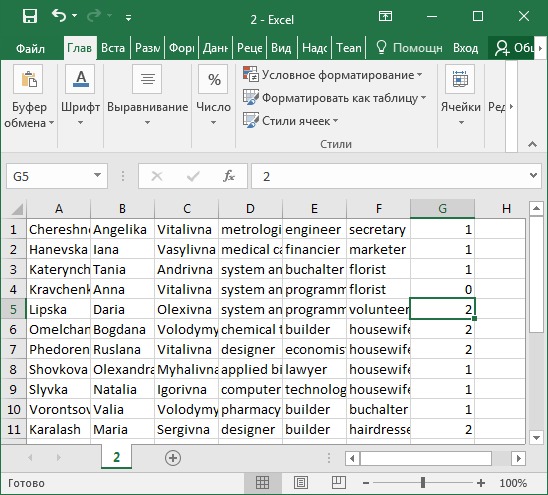


Рис.6-7 Редагування даних

**Висновок:** При виконання даної лабараторної роботи я набула навичок в роботі із записами, в інтеграції даних, в модульному програмуванні. Написала програму у якій розробила відомісті про сім’ю студентів. Розробила процедури, що реалізують базові операції над цими об'єктами, зокрема: текстове введення-виведення (консольний і файловий); присвоювання; видалення відомостів про певного студента; редагування вибраної відомості про; сортування в алфавітному (за прізвищем) та числовому (за кількістю братів і сестер) параметру.