МІНІСТЕРСВО ОСВІТИ І НАУКИ УКРАЇНИ

Технічний коледж

НУ «Львівська політехніка»

Відділення

Інформаційних технологій

та комп`ютерної техніки

Звіт про виконання завдань

З предмету навчальна практика з основ програмування

та алгоритмічних мов.

Підготував

Студент групи 21-ПЗ

Гула Володимир

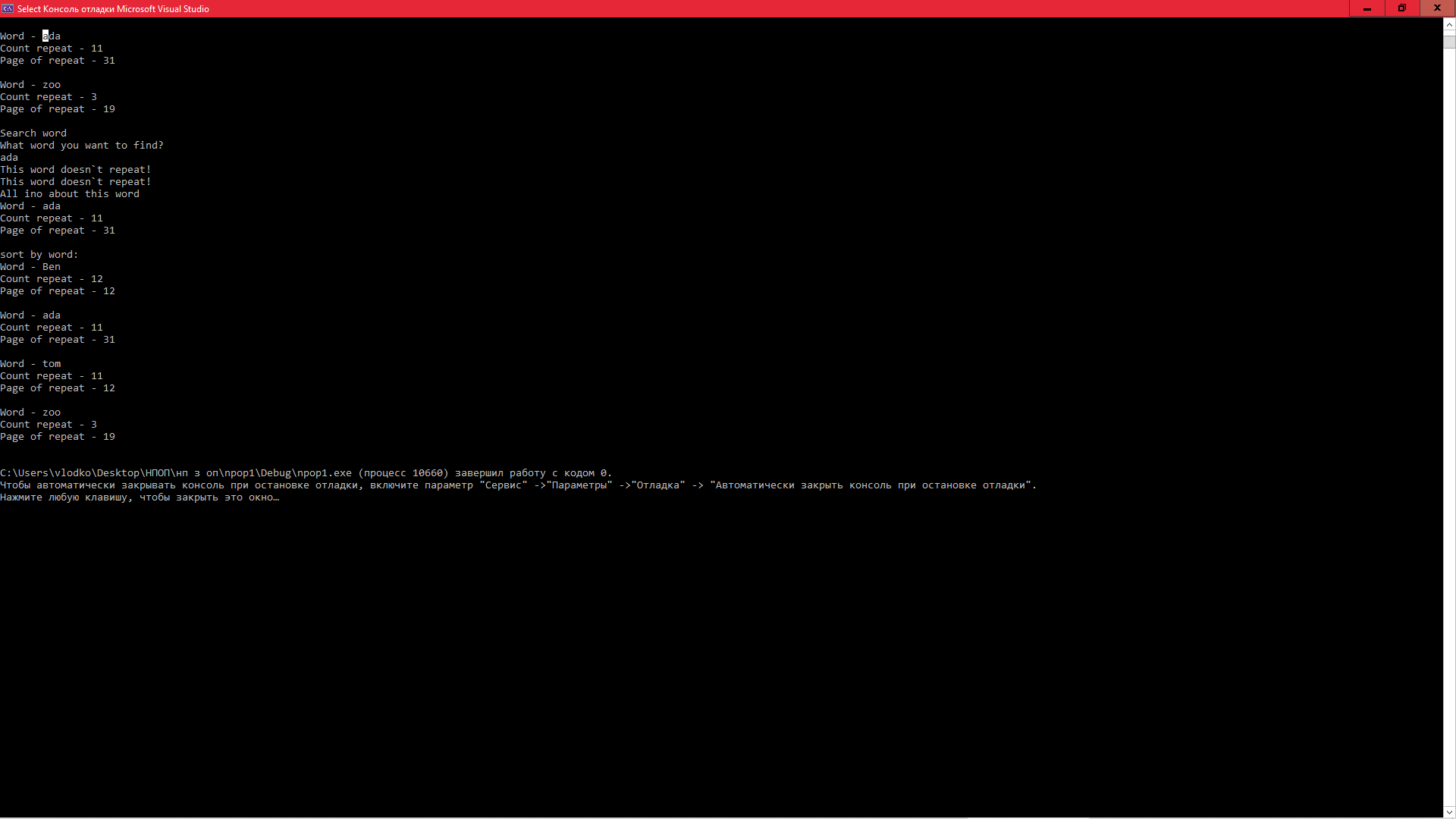
Львів 2021

Лабораторна робота № 1. Створення класу C++

**Мета роботи:** порівняння об’єктно-орієнтованого та функціонального підходів; початкове знайомство з класами, об’єктами та головними елементами об’єктного підходу.

Створити клас для обробки масиву даних у відповідності з наданим варіантом. Розмістити інтерфейс класу у заголовочному файлі, а визначення функцій та головну функцію програми – у двох окремих файлах. Передбачити можливість роботи з довільним числом записів, а також реалізувати окремими функціями класу: конструктори без параметрів та з параметрами ; додавання; знищення; виведення інформації на екран; пошук потрібної інформації за конкретною ознакою; редагування записів; сортування за різними полями

Варіант №6 - „предметний покажчик” -- Слово, к-сть сторінок де воно зустрічається, номера сторінок де воно зустрічається

   
Головна програма:  
#include <iostream>

#include <string>

#include "subIndex.h"

using namespace std;

int main() {

int size;

cout << "How much words you input?\n";

cin >> size;

subIndex\* note = new subIndex[50];

for (int i = 0; i < size; i++) { note[i].input(); }

cout << "Show info:\n";

for (int i = 0; i < size; i++) { note[i].show(); }

cout << "Add info: \n";

int count;

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

cout << "How much more repiting " << note[i].getWord() << endl;

cin >> count;

note[i] = note[i] + count;

}

cout << endl;

cout << "Show info:\n";

for (int i = 0; i < size; i++) { note[i].show(); }

cout << "Add new info:\nHow much word you wna to add?\n";

cin >> count;

subIndex\* tmp = new subIndex[count];

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

for (int j = size; j < size + count; j++) { note[j] = tmp[i].addInfo(); }

}

cout << endl;

size += count;

for (int i = 0; i < size; i++) { note[i].show(); }

cout << "Search word\n";

note->searchByWord(note, size);

cout << "sort by word:\n";

note = sortByWord(note, size);

for (int i = 0; i < size; i++) { note[i].show(); }

}

Оголошення класу:  
#include <iostream>

#include <string>

using namespace std;

#pragma once

class subIndex

{

private:

string word;

int repeatCount;

int pageNum;

private:

void inputInfo();

void showInfo();

public:

// Constructor

subIndex();

subIndex(string name, int count, int num);

~subIndex();

// Input outout methods

void input(bool ans = false);

void show(bool ans = false);

// Seter

void setAll(string name, int count, int num);

string getWord();

// Operator mathods

subIndex operator+ (int count); // Plusing count of repeating

subIndex addInfo();

// SORT

friend subIndex \*sortByWord(subIndex\* data, int size);

// search

void searchByWord(subIndex\* data, int size);

void searchByCount(subIndex\* data, int size);

};

Опис функцій класу:  
#include "subIndex.h"

#include <iostream>

#include <string>

#include <math.h>

#include <algorithm>

using namespace std;

void subIndex::inputInfo() {

cout << "Input word - ";

cin.ignore();

getline(cin, word);

cout << "Input how much '" << word << "' repeat - ";

cin >> repeatCount;

cout << "Input number page where this owrd repeat - ";

cin >> pageNum;

cout << endl;

}

void subIndex::showInfo() {

cout << "Word - " << word << endl;

cout << "Count repeat - " << repeatCount << endl;

cout << "Page of repeat - " << pageNum << endl;

cout << endl;

}

// Public mathods

subIndex::subIndex() {

word = "Undefined";

repeatCount = 0;

pageNum = 0;

}

subIndex::subIndex(string name, int count, int num) {

word = name;

repeatCount = count;

pageNum = num;

}

subIndex::~subIndex(){}

void subIndex::setAll(string name, int count, int num) {

word = name;

repeatCount = count;

pageNum = num;

}

string subIndex::getWord() { return word; }

void subIndex::input(bool ans) { inputInfo(); }

void subIndex::show(bool ans) { showInfo(); }

subIndex subIndex::operator+(int count) {

subIndex tmp;

if (count < 0) { tmp.setAll(word, repeatCount + abs(count), pageNum); }

else { tmp.setAll(word, repeatCount + count, pageNum); }

return tmp;

}

subIndex subIndex::addInfo() {

subIndex tmp;

tmp.input();

return tmp;

}

void subIndex::searchByWord(subIndex\* data, int size) {

string searchWord;

cout << "What word you want to find?\n";

cin >> searchWord;

int n = 0;

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

if (searchWord == data[i].word) {

cout << "All ino about this word\n";

data[i].show();

n++;

}

if (n == 0) { cout << "This word doesn`t repeat!\n"; }

}

}

void subIndex::searchByCount(subIndex\* data, int size) {

int countSearch;

cout << "What word you want to find?\n";

cin >> countSearch;

int n = 0;

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

if (countSearch == data[i].repeatCount) {

cout << "All ino about this word\n";

data[i].show();

n++;

}

if (n == 0) { cout << "This word doesn`t repeat!\n"; }

}

}

subIndex \*sortByWord(subIndex\* data, int size) {

string\* allWords = new string[size];

subIndex\* sortArr = new subIndex[size];

for (int i = 0; i < size; i++) { allWords[i] = data[i].word; }

sort(allWords, allWords + size);

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

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

if (allWords[i] == data[j].word) { sortArr[i] = data[j]; }

}

}

return sortArr;

}