МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 11

з навчальної дисципліни

“Базові методології та технології програмування”

Реалізація програмних засобів оброблення динамічних структур даних та бінарних файлів

ВИКОНАВ

Студент

академічної групи КІ-23

Замай Володимир

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

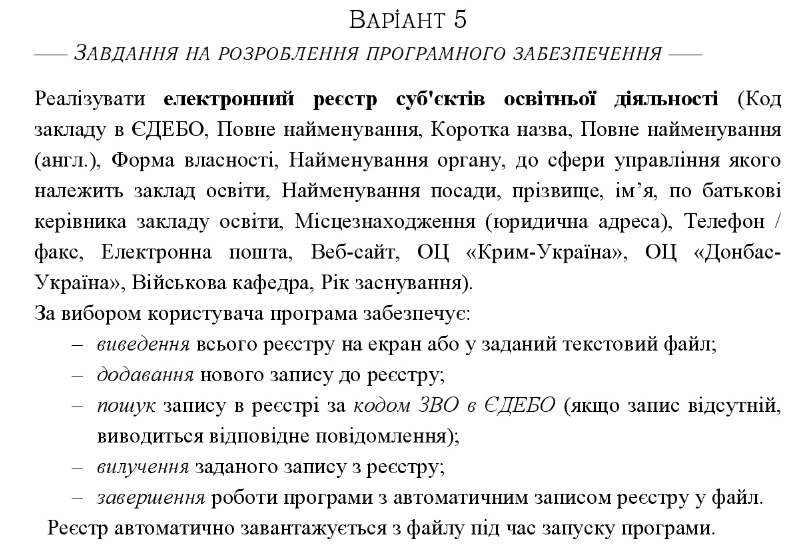
Ганна ДРЄЄВА

Кропивницький – 2024

**Тема:** Реалізація програмних засобів оброблення динамічних структур даних та бінарних файлів

**Мета:** Набуття ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам’яттю та бінарними файловими потоками.

**Завдання:**

****

**Порядок виконання лабораторної роботи та методичні рекомендації:**

1. Завантажити власний Git-репозиторій https://github.com/odorenskyi/ student-name (в \Lab11\tasks міститься завдання на реалізацію ПЗ).

2. До звіту з лабораторної роботи (далі ― звіт) записати мету роботи, номер варіанту, завдання, склад команди ІТ-проекта.

3. Зібратись командою на мітинг (від англ. meeting, зібрання) та колективно виконати наступне:

3.1) аналіз задач ІТ-проекта та вимог до програмного забезпечення;

3.2) обговорити специфікації ПЗ, концептуальні проектні рішення, сформулювати й узгодити архітектуру програмного засобу, загальні алгоритми функціонування та інтерфейси модулів тощо;

3.3) визначити й обґрунтувати обраний вид динамічної структури даних (список, стек, черга, дерево або ін.) для реалізації бази даних ПЗ, складових її елементів (поля структури і їх типи);

3.4) у \Lab11\prj створити заголовковий файл struct\_type\_project\_N.h (N ― номер варіанта завдання) та мовою програмування С++ описати в ньому елемент динамічної структури даних ;

3.5) розподілити між учасниками команди підзадачі з реалізації операцій над динамічною структурою даних відповідно до розробленої архітектури програмного засобу: кожен учасник реалізовує дві функції (модуля) оброблення даних;

3.6) скласти план робіт з виконання ІТ-проекта, виходячи із розподілених між учасниками команди субзадач та вимог міжнародного стандарту ISO/IEC 12207 щодо виконання процесів реалізації програмних засобів;

3.7) розроблений план виконання ІТ-проекта затвердити у викладача.

4. Затверджений план реалізації проекта включити до звіту .

5. Самостійно (індивідуально кожним учасником команди) виконати передбачений планом перелік робіт ; функції оброблення динамічної структури даних, опис типу елемента якої міститься у struct\_type\_project\_N.h, реалізувати як складові нової статичної бібліотеки і/або заголовкового файлу С/С++ із назвою, що містить прізвище розробника; отримані артефакти процесу реалізації програмного модуля (проектування архітектури, детального проектування, конструювання, тестування) включити до звіту.

6. Самостійно (індивідуально кожним учасником команди):

6.1) з Git-репозиторіїв учасників команди завантажити модулі (статичні бібліотеки .a і/або заголовкові файли .h) у \Lab11\prj.

6.2) в Code::Blocks IDE створити проект консольного додатка prj\_N\_Прізвище ( N ― номер варіанта завдання);

6.3) відповідно до стандарту ISO/IEC 12207 реалізувати програмний засіб згідно з завданням, використовуючи функції (модулі), розроблені учасниками команди ІТ-проекта ;

6.4) вихідний код (текст) проекта prj\_N\_Прізвище включити до звіту;

6.5) проаналізувати хід виконання лабораторних завдань і самостійно одержані результати, на основі чого сформулювати обґрунтовані висновки з виконаної лабораторної роботи, викласти їх обсягом не менше двох сторінок машинного (комп’ютерного) тексту та включити до звіту; 6.6) підготувати й зберегти у \Lab11\Report звіт про виконання лабораторної роботи, оформлений згідно з ДСТУ 3008:2015 “Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання”, зі змістом, визначеним цим порядком виконання лабораторної роботи ;

6.7) представити до захисту звіт з виконаної лабораторної роботи і проект у Git-репозиторії https://github.com/odorenskyi/student\_name.

**Склад команди IT-проекта:** Царенко С., Самброс С., Замай В.

**Результати обговорення**

1. Аналіз задач лабораторної роботи №11 та вимог до програмного забезпечення:

Програма повинна реалізовувати електронний реєстр суб'єктів оціночної діяльності з полями: Код, Повна назва, Коротка назва, Повна назва(англ.),Форма власності, Найменування органу, Найменування посади, Прізвище, ім’я, по батькові, Місцезнаходження, Телефон, Електронна пошта, Веб-сайт, ОЦ “Крим-Україна”, ОЦ “Донбас-Україна”, Військова кафедра, Рік заснування .

Програма має забезпечувати функції: виведення реєстру, додавання запису, пошук запису за кодом, вилучення запису, автоматичне збереження і завантаження реєстру з файлу.

1. Обговорення специфікацій ПЗ, архітектури, алгоритмів та інтерфейсів:

Для реалізації реєстру доцільно використати векторну динамічну структуру даних. Ця структура визначена в стандартній бібліотеці C++ (<vector>).

Основні модулі: структура даних для запису реєстру, функції для операцій над реєстром (додавання, пошук, вилучення, виведення тощо), інтерфейс користувача з меню вибору дій.

Дані будуть зберігатися у текстовому файлі.

1. Визначення динамічної структури даних для реалізації бази даних ПЗ:

Кожен елемент списку буде представляти запис реєстру з полями:

Код(int), Повна назва(string), Коротка назва(string), Повна назва(англ.) (string),Форма власності(string), Найменування органу(string), Найменування посади(string), Прізвище,ім’я по батькові(string), Місцезнаходження(string), Телефон(string), Електронна пошта(string), Веб-сайт(string), ОЦ “Крим-Україна”(bool) , ОЦ “Донбас-Україна”(bool), Військова кафедра(bool), Рік заснування(int).

1. Створення заголовкового файлу з описом елемента структури даних (struct\_type\_project\_5.h):

#ifndef STRUCT\_TYPE\_PROJECT\_5\_H\_INCLUDED

#define STRUCT\_TYPE\_PROJECT\_5\_H\_INCLUDED

#include <string>

using namespace std;

struct EducationalInstitution {

int code;

string fullName;

string shortName;

string englishName;

string ownershipForm;

string managementBody;

string directorPosition;

string directorName;

string address;

string phone;

string email;

string website;

bool isCrimeaUkraine;

bool isDonbasUkraine;

bool hasMilitaryDepartment;

int foundationYear;

};

#endif // STRUCT\_TYPE\_PROJECT\_5\_H\_INCLUDED

5. Архітектура ПЗ

Modules:

-ModulesTsarenko.h (Головний файл  Царенко)

-ModulesSambros.h(Головний файл Самброс)

-ModulesZamai.h(Головний файл Замай)

-struct\_type\_project\_5.h(Заголовковий файл який містить структуру проекту)

ModulesTsarenko (Модулі розроблені Царенко)

ModulesSambros (Модулі розроблені Самброс)

ModulesZamai (Модулі розроблені Замай)

prj\_5\_Tsarenko (програма в якій відбувається виклик розроблених модулів)

6.    Розподіл підзадач з реалізації операцій над структурою даних. Кожен учасник команди реалізує дві функції для роботи зі списком :

Пошук запису за кодом(Самброс)

Вилучення запису зі списку(Самброс)

Додавання нового запису в список (Замай)

Виведення списку на екран (Замай)

Збереження у файл (Царенко)

Завантаження з файла (Царенко)

7.    Складання плану робіт з виконання лабораторної :

- Розробка структури даних, функцій для роботи з нею

- Реалізація інтерфейсу користувача(меню с викликом функцій)

- Оформлення документації

**Лістинг ModulesZamai.h:**

#ifndef MODULESZAMAI\_H\_INCLUDED

#define MODULESZAMAI\_H\_INCLUDED

#include <string>

#include <vector>

#include "struct\_type\_project\_5.h"

void printRegistry();

void addInstitution();

#endif // MODULESZAMAI\_H\_INCLUDED

**Лістинг ModulesZamai:**

void printRegistry() {

for (const auto& institution : registry) {

cout << "Код закладу в ЄДЕБО: " << institution.code << endl;

cout << "Повне найменування: " << institution.fullName << endl;

cout << "Коротка назва: " << institution.shortName << endl;

cout << "Повне найменування(англ.): " << institution.englishName << endl;

cout << "Форма власності: " << institution.ownershipForm << endl;

cout << "Найменування органу, до сфери управління якого належить заклад освіти: " << institution.managementBody << endl;

cout << "Найменування посади керівника закладу освіти: " << institution.directorPosition << endl;

cout << "Прізвище, ім'я, по батькові керівника закладу освіти: " << institution.directorName << endl;

cout << "Місцезнаходження(юридична адреса): " << institution.address << endl;

cout << "Телефон: " << institution.phone << endl;

cout << "Електронна пошта: " << institution.email << endl;

cout << "Веб-сайт: " << institution.website << endl;

cout << "ОЦ <<Крим-Україна>>: " << institution.isCrimeaUkraine << endl;

cout << "ОЦ <<Донбас-Україна>>: " << institution.isDonbasUkraine << endl;

cout << "Військова кафедра: " << institution.hasMilitaryDepartment << endl;

cout << "Рік заснування: " << institution.foundationYear << endl;

}

}

void addInstitution() {

EducationalInstitution institution;

cout << "Введіть код закладу в ЄДЕБО: ";

cin >> institution.code;

cin.ignore();

cout << "Введіть повне найменування: ";

getline(cin, institution.fullName);

cout << "Введіть коротку назву: ";

getline(cin, institution.shortName);

cout << "Введіть повне найменування(англ.): ";

getline(cin, institution.englishName);

cout << "Введіть форму власності: ";

getline(cin, institution.ownershipForm);

cout << "Введіть найменування органу: ";

getline(cin, institution.managementBody);

cout << "Введіть найменування посади: ";

getline(cin, institution.directorPosition);

cout << "Введіть прізвище, ім'я, по батькові керівника закладу освіти: ";

getline(cin, institution.directorName);

cout << "Введіть місцезнаходження(юридична адреса): ";

getline(cin, institution.address);

cout << "Введіть телефон: ";

getline(cin, institution.phone);

cout << "Введіть електронна пошту: ";

getline(cin, institution.email);

cout << "Введіть веб-сайт: ";

getline(cin, institution.website);

cout << "ОЦ <<Крим-Україна>> (1 - так, 0 - ні): ";

string crimeaUkraineStr;

getline(cin, crimeaUkraineStr);

institution.isCrimeaUkraine = (crimeaUkraineStr == "1");

cout << "ОЦ <<Донбас-Україна>> (1 - так, 0 - ні): ";

string donbasUkraineStr;

getline(cin, donbasUkraineStr);

institution.isDonbasUkraine = (donbasUkraineStr == "1");

cout << "Військова кафедра (1 - так, 0 - ні): ";

string militaryDepartmentStr;

getline(cin, militaryDepartmentStr);

institution.hasMilitaryDepartment = (militaryDepartmentStr == "1");

cout << "Введіть рік заснування: ";

cin >> institution.foundationYear;

registry.push\_back(institution);

}

**Висновок:**  
У ході виконання лабораторної роботи №11 "Реалізація програмних засобів оброблення динамічних структур даних та бінарних файлів" були здобуті ґрунтовні вміння і практичні навички командної реалізації програмного забезпечення. Ми розробили функції для оброблення динамічних структур даних, зокрема для додавання та виведення інформації про освітні заклади. Використання стандартних засобів C++ для керування динамічною пам’яттю та бінарними файловими потоками дозволило створити ефективне та надійне програмне забезпечення.

Завдяки реалізації функцій printRegistry та addInstitution ми навчилися створювати та обробляти динамічні масиви даних, що містять інформацію про освітні заклади, використовувати динамічну пам'ять у програмі на C++, додаючи нові записи до реєстру освітніх закладів, застосовувати ввід та вивід даних, включаючи обробку строк та числових значень, а також реалізовувати логічні умови для обробки спеціальних полів (таких як статуси "Крим-Україна", "Донбас-Україна", наявність військової кафедри).