

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ  
КАФЕДРА КІБЕРБЕЗПЕКИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ЗВІТ  
з лабораторної роботи № 12  
з дисципліни  
«Базові методології та технології програмування»  
на тему:  
ПРОГРАМНА РЕАЛІЗАЦІЯ АБСТРАКТНИХ ТИПІВ ДАНИХ

Виконав:  
студент академічної групи КН-22  
ПІВНЕНКО О.М.

Перевірив:  
викладач  
СОБІНОВ О.Г.

## Лабораторна робота № 12

### Варіант № 17

**ТЕМА:** Програмна реалізація абстрактних типів даних.

**МЕТА:** полягає у набутті ґрунтовних вмінь і практичних навичок об'єктного аналізу й проектування, створення класів C++ та тестування їх екземплярів, використання препроцесорних директив, макросів і макрооператорів під час реалізації програмних засобів у кросплатформовому середовищі Code::Blocks.

### ЗАВДАННЯ:

1. Як складову заголовкового файлу ModulesПрізвище.h розробити клас ClassLab12\_Прізвище — формальне представлення абстракції сутності предметної області (об'єкта) за варіантом, — поведінка об'єкта якого реалізовує розв'язування задачі 7.1.
2. Реалізувати додаток Teacher, який видає 100 звукових сигналів і в текстовий файл TestResults.txt записує рядок “Встановлені вимоги порядку виконання лабораторної роботи порушено!”, якщо файл проектуа main.cpp під час його компіляції знаходився не в \Lab12\prj, інакше — створює об'єкт класу ClassLab12\_Прізвище із заголовкового файлу ModulesПрізвище.h та виконує його unit-тестування за тест-сьютом(ами) із \Lab12\TestSuite\, протоколюючи результати тестування в текстовий файл \Lab12\TestSuite\TestResults.txt.

### ХІД ЛАБОРАТОРНОЇ РОБОТИ

У ході виконання лабораторної роботи було створено абстрактний тип даних (клас), який відобразив об'єкт предметної області. Був створений конструктор, функції-гетери, які повертали значення, а також обробка помилок (у разі, якщо передається від'ємне число у якості довжини грані куба). Вихідний код, що реалізує клас, зазначений у лістингу 12.1. Був реалізований проект Teacher, який, у разі якщо компіляція відбувалась за правильним шляхом,

проводил Unit-тестування, або, інакше, виводив повідомлення про те, що умови перевірки порушені, після чого примусово закривав додаток. Вихідний код проектів Teacher, ModulesPivnenko зазначений у додатках 1, 2 відповідно. Тест-сьют зазначений у додатку 3. Результати тестування (текст із файлу, куди записані результати), зазначений у додатку 4.

### Лістинг 12.1

```
#include <string>
#include <exception>

class ClassLab12_Pivnenko{
public:
    float calc_volume(){
        return length_edge * length_edge * length_edge;
    }

    float rtn_edge(){
        return length_edge;
    }

    ClassLab12_Pivnenko(float edge);

private:
    float length_edge = 0;
};

ClassLab12_Pivnenko::ClassLab12_Pivnenko(float edge){
    if(edge < static_cast<float>(0))
        throw std::invalid_argument("Length will be > 0");
    length_edge = edge;
}
```

## ВИСНОВОК

У ході виконання лабораторної роботи я закріпив свої знання із мови програмування C++, її синтаксису, будови. Попрацював із динамічним масивом, файловим вводом/виводом, створив тест-сьют, реалізував unit-тестування. Варто також зауважити, що важливим надбанням є той факт, що я закріпив знання із структурної, процедурної, модульної, а також основ об'єктно-орієнтованого програмування. Створив клас, який представляв предметну область (а саме куб), реалізував поля, що зберігали значення об'єкта, провів аналіз, створив конструктор, обробку можливих помилок (передача від'ємного аргумента у якості довизни грані куба), створив функції-гетери, що могли б дати користувачу інформацію про об'єм, а також довжину грані куба. Крім цього, закріпив свої знання щодо різних видів помилок (invalid\_argument, exception).

```

#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include "ModulesPivnenko.hpp"

using namespace std;

string getPath(){
    return __FILE__;
}

bool isFind(string subStr, string mainStr){
    if(mainStr.find(subStr) != string::npos)
        return true;

    return false;
}

bool isCondMet(){
    return isFind(R"(\Lab12\prj)", getPath());
}

bool testCase(float a, float excp, bool calc_or_edge){
    try{
        ClassLab12_Pivnenko lab = ClassLab12_Pivnenko(a);

        if(calc_or_edge == true){
            if(lab.calc_volume() == excp)
                return true;
            else
                return false;
        }
        else{
            if(lab.rtn_edge() == excp)
                return true;
            else
                return false;
        }
    }
    catch(const std::invalid_argument& e){
        return true;
    }
    catch(...){
        return false;
    }
    return false;
}

vector<float> readCases(string path){
    string line_content = "";
    vector<float> output_floats;
    std::ifstream reader;
    cout << "The path is " << path << endl;
    reader.open(path);

    if(reader.is_open() == false){
        cout << "Не можна зчитати файл із тест-кейсами." << endl;
        cout << "Натисніть ENTER, щоб вийти із програми." << endl;
        getchar();
        exit(0);
    }
}

```

```

while(!reader.eof()){
    getline(reader, line_content);
    cout << "\n" << line_content << "\n";
    output_floats.push_back(stof(line_content));
}
reader.close();

return output_floats;
}

void testSuite(string path_read, string path_write){
    vector<float> excp_values = readCases(path_read);
    const float test_values[] = {0, 5, 0, -1.5, 1.5, 15, -15, 20.0, 57.0};
    const bool calc_or_edge[] = {false, true, true, false, true, false, false, true, true};
    ofstream writer (path_write);

    for(int i = 0; i < 9; i += 1){
        if(testCase(test_values[i], excp_values[i], calc_or_edge[i])){
            writer << "-----\n" << "Test Case number " << i + 1 << "
PASSED" << "\n-----\n";
        }
        else
            writer << "-----\n" << "Test Case number " << i + 1 << "
FAILED" << "\n-----\n";
    }

    writer.close();
}

int main()
{
    setlocale(LC_ALL, "ukr");

    if(isCondMet() == false){
        cout << "ВСТАНОВЛЕНІ ВИМОГИ ПОРЯДКУ ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ ПОРУШЕНО!";
        cout << "\a";
        getchar();
        exit(0);
    }

    string path_read = "";
    string path_write = "";

    cout << "Уведіть шлях до файлу, звідки хочете читати: ";
    cin >> path_read;
    cout << endl;

    cout << "Уведіть шлях до файлу, куди хочете записати: ";
    cin >> path_write;
    cout << endl;

    testSuite(path_read, path_write);

    cout << "Натисніть ENTER щоб вийти.";
    getchar();
    getchar();

    return 0;
}

```

```
#include <string>
#include <exception>

class ClassLab12_Pivnenko{
public:
    float calc_volume(){
        return length_edge * length_edge * length_edge;
    }

    float rtn_edge(){
        return length_edge;
    }

    ClassLab12_Pivnenko(float edge);

private:
    float length_edge = 0;
};

ClassLab12_Pivnenko::ClassLab12_Pivnenko(float edge){
    if(edge < static_cast<float>(0))
        throw std::invalid_argument("Length will be > 0");
    length_edge = edge;
}
```

## Додаток 3.

Artifact: Test Suite

Date: 5/5/2023

Назва тестового набору Test Suite Description	TS_lab12
Назва проекту / ПЗ Name of Project / Software	ClassLab12_Pivnenko.h
Рівень тестування Level of Testing	модульний / Module Testing
Автор тест-сюита Test Suite Author	Олександр Півненко
Виконавець Implementer	Олександр Півненко

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	Створити об'єкт класу ClassLab12_Pivnenko з конструктором (0)	Об'єкт буде створено. Помилки не виникає.	PASSED
TC_02	Створити об'єкт класу ClassLab12_Pivnenko з конструктором (5). Викликати calc_volume()	Об'єкт створено, об'єм буде дорівнювати 125.	PASSED
TC_03	Створити об'єкт класу ClassLab12_Pivnenko з конструктором (0). Викликати calc_volume()	Об'єкт створено, об'єм буде дорівнювати 0.	PASSED
TC_04	Створити об'єкт класу ClassLab12_Pivnenko з конструктором (-1.5). Викликати calc_volume()	Об'єкт НЕ буде створено. Викликається помилка.	PASSED
TC_05	Створити об'єкт класу ClassLab12_Pivnenko з конструктором (1.5). Викликати calc_volume()	Об'єкт створено, об'єм буде дорівнювати 3.375.	PASSED
TC_06	Створити об'єкт класу ClassLab12_Pivnenko з конструктором (15). Викликати rtn_edge ()	Об'єкт створено, повернено rtn_edge() 15.	PASSED
TC_07	Створити об'єкт класу ClassLab12_Pivnenko з конструктором (-15). Викликати rtn_edge ()	Об'єкт НЕ буде створено. Викликається помилка.	PASSED
TC_08	Створити об'єкт класу ClassLab12_Pivnenko з конструктором (20.0). Викликати calc_volume ()	Об'єкт створено, об'єм буде дорівнювати 8000.	PASSED
TC_09	Створити об'єкт класу ClassLab12_Pivnenko з конструктором (57.0). Викликати calc_volume ()	Об'єкт створено, об'єм буде дорівнювати 185193.	PASSED

-----  
Test Case number 1 PASSED  
-----

-----  
Test Case number 2 PASSED  
-----

-----  
Test Case number 3 PASSED  
-----

-----  
Test Case number 4 PASSED  
-----

-----  
Test Case number 5 PASSED  
-----

-----  
Test Case number 6 PASSED  
-----

-----  
Test Case number 7 PASSED  
-----

-----  
Test Case number 8 PASSED  
-----

-----  
Test Case number 9 PASSED  
-----