

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 12
з навчальної дисципліни
“Базові методології та технології програмування”
ПРОГРАМНА РЕАЛІЗАЦІЯ АБСТРАКТНИХ ТИПІВ ДАНИХ

ЗАВДАННЯ ВИДАВ
доцент кафедри кібербезпеки
та програмного забезпечення
Доренський О. П.
<https://github.com/odorenskyi/>

ВИКОНАВ
студент академічної групи КІ-22-2
Карпова Є. І.

ПЕРЕВІРИВ
викладач кафедри кібербезпеки
та програмного забезпечення
Собінов О. Г.

Тема: Програмна реалізація абстрактних типів даних.

Мета: Набуття 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 ModulesKarpova.h розробити клас ClassLab12_Karpova — 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 проекту main.cpp під час його компіляції знаходився не в \Lab12\prj, інакше — створює об'єкт класу ClassLab12_Karpova із заголовкового файлу ModulesKarpova.h та виконує його unit-тестування за тест-сьютом(ами) із \Lab12\TestSuite\, протоколюючи результати тестування в текстовий файл \Lab12\TestSuite\TestResults.txt.

Варіант 37

Лістинг коду із ModulesKarpova.cpp

```
#include <cmath>
#include "ModulesKarpova.h"

ClassLab12_Karpova::ClassLab12_Karpova(float heightToSet, float radiusToSet) {
    height = heightToSet;
    radius = radiusToSet;
}

float ClassLab12_Karpova::getHeight() {
    return height;
}

float ClassLab12_Karpova::getRadius() {
    return radius;
}

float ClassLab12_Karpova::getVolume() {
    return M_PI*pow(radius, 2)*height;
}

bool ClassLab12_Karpova::setHeight(float heightToSet) {
    if (heightToSet > 0) {
        height = heightToSet;
    }
}
```

```

        return true;
    } else {
        return false;
    }
}

bool ClassLab12_Karpova::setRadius(float radiusToSet) {
    if (radiusToSet > 0) {
        radius = radiusToSet;
        return true;
    } else {
        return false;
    }
}

```

Лістинг із ModulesKarpova.h

```

#ifndef MODULESKARPOVA_H_INCLUDED
#define MODULESKARPOVA_H_INCLUDED

class ClassLab12_Karpova {
private:
    float height;
    float radius;
public:
    ClassLab12_Karpova(float, float);
    float getHeight();
    float getRadius();
    float getVolume();
    bool setHeight(float);
    bool setRadius(float);
};

#endif // MODULESKARPOVA_H_INCLUDED

```

Лістинг коду із Teacher.cpp

```

#include <fstream>
#include <iostream>
#include <sstream>
#include <string>
#include <sys/stat.h>
#include <vector>
#include "ModulesKarpova.h"

using namespace std;

int main() {

```

```

bool inSoftware = false;
struct stat sb;
ofstream result;

if (stat("../TestSuite", &sb) == 0) // якщо у prj/Teacher
    result.open("../TestSuite/TestResults.txt");
else { // якщо у Software
    inSoftware = true;
    result.open("../TestSuite/TestResults.txt");
}

if (!result)
    return 1;

string path = __FILE__;

if (path.find("\\lab12\\prj") == string::npos) {
    for (int i = 0; i < 100; i++)
        cout << "\a";
    result << "Встановлені вимоги порядку виконання лабораторної роботи
порушено!" << endl;
    return 2;
}

ifstream testsuite;
if (!inSoftware) //
    testsuite.open("../TestSuite/unit.txt");
else {
    inSoftware = true;
    testsuite.open("../TestSuite/unit.txt");
}

vector<vector<string>> content;
vector<string> row;
string line, word;
unsigned int lines = 0;
bool firstLine = true;

if (testsuite.is_open()) {
    while(getline(testsuite, line)) {
        if (firstLine) {
            firstLine = false;
            continue;
        }

        row.clear();

        stringstream str(line);

```

```

        while(getline(str, word, '\u0009'))
            row.push_back(word);
        content.push_back(row);
        lines++;
    }
} else
    return 3;

float height[lines];
float radius[lines];
float expectedVolume[lines];

for (unsigned int i = 0; i < content.size(); i++) {
    height[i] = stof(content[i][0]);
    radius[i] = stof(content[i][1]);
    expectedVolume[i] = stof(content[i][2]);
}

ClassLab12_Karpova *classLab = new ClassLab12_Karpova(height[0], radius[0]);
for (unsigned int i = 0; i < lines; i++) {
    float volume = classLab->getVolume();
    result << "Test " << i + 1 << " " << (volume == expectedVolume[i] ? "passed" :
"failed") << endl;
    if (i + 1 != lines) {
        classLab->setHeight(height[i + 1]);
        classLab->setRadius(radius[i + 1]);
    }
}
}

```

TestSuite

TestResults.txt

Test 1 passed
 Test 2 passed
 Test 3 passed

unit.txt

Height	Radius	Volume
15	5	1178.0972450961724
10	10	3141.592653589793
5	3	141.3716694115407

Висновок:

У ході виконання лабораторної роботи "Програмна реалізація абстрактних типів даних" було досягнуто поставленої мети - набуто ґрунтовних вмінь і практичних навичок об'єктного аналізу й проектування, створення класів C++ та тестування їх екземплярів, використання препроцесорних директив, макросів і макрооператорів.

У першому завданні було розроблено клас `ClassLab12_Karova`, який виступає формальним представленням абстракції сутності предметної області (об'єкта) за варіантом. Цей клас реалізує поведінку об'єкта, необхідну для розв'язування задачі 7.1. Реалізація даного класу була виконана у заголовковому файлі `ModulesKarova.h`.

У другому завданні був створений додаток `Teacher`, який виконує дві основні функції. По-перше, він видає 100 звукових сигналів. По-друге, в залежності від розташування файлу проекту `main.cpp`, відбувається запис певного повідомлення у текстовий файл `TestResults.txt`. Якщо файл проекту не знаходився у відповідній директорії `\Lab12\prj`, то в файл записується рядок "Встановлені вимоги порядку виконання лабораторної роботи порушено!". У протилежному випадку, створюється об'єкт класу `ClassLab12_Karova` з файлу `ModulesKarova.h` і проводиться його unit-тестування за тест-сьютами з директорії `\Lab12\TestSuite\`. Результати тестування протоколюються в текстовий файл `\Lab12\TestSuite\TestResults.txt`.

У процесі виконання лабораторної роботи було успішно досягнуто поставлених цілей та завдань. Було показано розуміння принципів об'єктного аналізу й проектування, вміння створювати класи C++, а також проводити їх тестування. Крім того, використання препроцесорних директив, макросів і макрооператорів дозволило ефективно реалізовувати програмні засоби у кросплатформовому середовищі `Code::Blocks`.

Отже, лабораторна робота дала змогу розширити теоретичні знання з об'єктного аналізу й проектування, а також отримати практичні навички розробки та тестування програмних засобів.