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/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КI-22-1

Дяченко Р. П.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Собінов О. Г.

Кропивницький – 2023

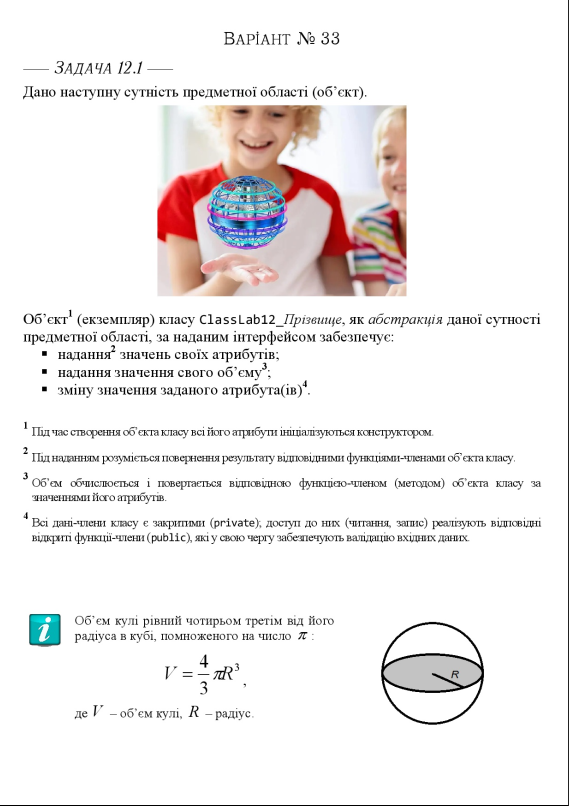
**Лабораторна робота №12**

**Тема:** Програмна реалізація абстрактних типів даних

**Мета:** полягає у набутті ґрунтовних вмінь і практичних  
навичок об’єктного аналізу й проектування, створення класів С++  
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  
   проекта main.срр 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.



**Аналіз задачі 12.1:**

Об’єкт, який я повинен реалізувати за допомогою абстрактного типу даних це куля. В завданні уже наведена сам куля, та її формула. Як можна побачити головними даними цієї кулі є R(радіус) та V(об’єм). Я буду використовувати тип даних double для їх позначення. Поля, або змінні, що будуть в цьому об’єкті це відповідно радіус і об’єм(приватні).

Відкритими будуть інтерфейси, що оперують цими даними, а саме: змінюють радіус/об’єм, отримуємо радіус/об’єм, і рахуємо об’єм(відповідно заданої формули).

**Аналіз задачі 12.2:**

Потрібно 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  
проекта main.срр під час його компіляції знаходився не в \Lab12\prj,  
інакше –– створює об’єкт класу ClassLab12\_Diachenko із заголовкового файлу ModulesDiachenko.h та виконує його unit-тестування  
за тест-сьютом(ами) із \Lab12\TestSuite\, протоколюючи результати  
тестування в текстовий файл \Lab12\TestSuite\TestResults.txt.

**Висновок:** на цій лабораторній роботі я набув ґрунтовних вмінь і практичних  
навичок об’єктного аналізу й проектування, створення класів С++  
та тестування їх екземплярів, використання препроцесорних  
директив, макросів і макрооператорів під час реалізації програмних  
засобів у кросплатформовому середовищі Code::Blocks. Створив власний абстрактний тип з використання класів мови програмування C++. Також створив додаток, що протестував об’єкти цього класу, зроблені згідно тест-сьютів.

Лабораторна робота, особисто для мене, була цікава та повчальна. Особливо було цікаво створювати свій клас, та його екземпляри відповідно. Ця тема мені дуже сподобалася. Щодо проблем з лабораторною роботою, їх не було.

**Додаток А(лістинги)**

**main.cpp:**

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

#include "../ModulesDiachenko.h"

int main() {

std::string projectPath = "Lab12/prj/main.cpp";

if (projectPath != "Lab12/prj/main.cpp") {

std::ofstream outputFile("TestResults.txt");

outputFile << "Встановлені вимоги порядку виконання лабораторної роботи порушено!" << std::endl;

outputFile.close();

for (int i = 0; i < 100; i++) {

std::cout<<'\a'<<std::endl;

}

}

else {

ClassLab12\_Diachenko testObject;

double testSuite[2][5] = {{5.5, 1.5, 1, 22.22, 5.555}, {696.91, 14.1372, 4, 45953.7, 718.027}};

std::ofstream outputFile("TestResults.txt"); // Відкриття файлу для запису

if (outputFile.is\_open()) {

for(int i = 0; i < 5; i++){

testObject.changeRadius(testSuite[0][i]);

testObject.calcVolume();

if ((int)testObject.getVolume() == (int)testSuite[1][i]) outputFile<< "ID-" << i+1 << ": PASSED"<<std::endl;

else outputFile<< "ID-" << i+1 << ": FAILED"<<std::endl;

}

} else {

std::cout << "Не вдалося відкрити файл для запису." << std::endl;

}

}

std::cout<< "Програму завершено. Файл знаходиться в prj/Teacher!";

return 0;

}

**ModulesDiachenko:**

#ifndef MODULESDIACHENKO\_H\_INCLUDED

#define MODULESDIACHENKO\_H\_INCLUDED

#include <cmath>

class ClassLab12\_Diachenko{

public:

ClassLab12\_Diachenko(){}

ClassLab12\_Diachenko(double radius);

void changeRadius(double radius);

void changeVolume(double volume);

double getRadius();

double getVolume();

double calcVolume();

private:

double radius = 0.0;

double volume = 0.0;

};

ClassLab12\_Diachenko::ClassLab12\_Diachenko(double radius){

this->radius = radius;

}

void ClassLab12\_Diachenko::changeRadius(double radius){

this->radius = radius;

}

void ClassLab12\_Diachenko::changeVolume(double volume){

this->volume = volume;

}

double ClassLab12\_Diachenko::getRadius(){

return radius;

}

double ClassLab12\_Diachenko::getVolume(){

return volume;

}

double ClassLab12\_Diachenko::calcVolume(){

this->volume = (4 \* M\_PI \* pow(this->radius, 3)) / 3;

}

#endif // MODULESDIACHENKO\_H\_INCLUDED

**TestResults.txt:**

ID-1: PASSED

ID-2: PASSED

ID-3: PASSED

ID-4: PASSED

ID-5: PASSED

**Додаток Б(тест-сьюти)**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TestSuite №12 |
| Назва проекта / ПЗ  Name of Project / Software | Teacher.exe |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Дяченко Руслан |
| Виконавець  Implementer | Дяченко Руслан |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| ID-1 | Ввести 5.5 | 696.6 | Passed |
| ID-2 | Ввести 1.5 | 14.1 | Passed |
| ID-3 | Ввести 1 | 14,2 | Passed |
| ID-4 | Ввести 22,22 | 45930.4 | Passed |
| ID-5 | Ввести 5.555 | 717.7 | Passed |