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”

На тему:

“Програмна реалізація абстрактних типів даних”

ВИКОНАВ

студент академічної групи КБ-21

\_\_\_\_\_\_\_\_\_ К.О. Задорожний

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_ П.С. Усік

м. Кропивницький 2022

**Мета роботи :**

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.

**Варіант:** №19

**Завдання**



**Аналіз виконання Задачі 12.1:**

Для опису м’яча нам потрібно радіус, щоб порахувати його площу. Радіус буде числом із плаваючим знаком

В екземплярі класу створюємо приватну змінну (радіус) для того, щоб не можна було отримати доступ, і реалізуємо функції для запису і отримання значення радіусу.

Далі, створюємо функцію для обчислення площі. Площа кулі обраховується за формулою 4/3 \* π \* R. Також створимо конструктор і перевантажимо, один буде порожній, а інший потрібно буде передавати радіус.

**Аналіз виконання Задачі 12.2:**

Для створення додатку Teacher нам потрібно test suite в вигляді текстового файлу, в якому ми запишемо taste case. Також потрібно зробити перевірку для компіляції. Якщо файл main.cpp знаходиться не в директорії..\ Lab12\prj то видати 100 звукових сигналів, та записати в TestResult.txt що вимог не дотримання. Інакше цей файл потрібно створити/переписати в директорії яка знаходиться за таким запитом ..\Lab12\TestSuite.

Далі потрібно прочитати файл і отримати з нього пункти Action та Expected Result. І в циклі га кількість тест кейсів перевірити, чи все правильно виконано, якщо так, то вивести pessed інакше вивести failed та вивести результат порівнянь результат того що повернула функція

**Висновок**

Під час виконання лабораторної роботи № 12 на тему **«**Програмна реалізація абстрактних типів даних**»** з предмету «Базові Методології та Технології Програмування» 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.

Відпрацював на практиці створення абстракції сутності предметної області (об’єкта). Моє завдання, було створення об'єкт класу м'яч, та створити застосунок Teacher для тестування цього класу. В класі я використав інкапсуляцію змінної radius і створив спеціальні методи, для роботи із нею. Також було створено метод, який рахує площу футбольного м'яча.

Моїм завданням на цю лабораторну роботу було:

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.

Для цієї роботи, потрібно було таке обладнання :

– персональний комп’ютер з ОС Windows XP / Vista / 7 / 8.x / 10, Linux 32-bit / 64-bit або Mac OS X; – вільне кросплатформове середовище розроблення програмного забезпечення Code::Blocks (www.codeblocks.org); – текстовий редактор (OpenOffice Writer, Microsoft Word або ін.).

При отриманні лабораторної роботи я дотримувався такого плану:

1. Завантажив свій Git-репозиторій
2. Створив папки prj, Software, Report, task, TestSuite
3. Створити Документ Word, та в нього записав титульну сторінку, мету, номер варіанту, завдання.
4. В файл README, записати тему, мету, варіант та завдання.
5. Проаналізував задачу 12.1 та вимог до неї.
6. Створив заголовковий файл ModulesZadorozhnhi.h, в ньому створив абстракції сутності предметної області.
7. Створив застосунок Teacher, який буде перевіряти клас і методи в ньому
8. Створив тестові артефакти + створив текстовий файл, в який записав Action та Expected Result
9. Протестував консольний застосунок Teacher і після того як всі тести було пройдено на позначку passed то завантажив програму на github
10. Завантажив тестовий артефакт на github
11. Завантажив папку Software на github з Teacher.exe
12. Завантажив готовий звіт на github

Текстовий файл в якому було записано test case я оформив в форматі таблиці. Для того щоб було зручно записувати дані та було наглядно видно, де які колонки, і куди що треба записувати. Написав окрему функцію для коректного зчитування таблиці. Функція повертає vector структур, в структурі находиться два поля, це action та expected result. При перевірці на коректну працю функції, в текстовий файл виводилося, що тест пройшов успішно, а коли функція давала збій, виводило що повинно бути, і що функція повертає.

Лабораторну роботу було цікаво виконувати, я зрозумів що класи - це потужний інструмент, для реалізації задач!

**ДОДАТОК А**  
**Лістинг ModulesZadorozhnyi.h**

#ifndef MODULESZADOROZHNYI\_H\_INCLUDED

#define MODULESZADOROZHNYI\_H\_INCLUDED

#include <cmath>

class ClassLab12\_Zadorozhnyi{

public:

ClassLab12\_Zadorozhnyi(){}

ClassLab12\_Zadorozhnyi(float radius);

void setRadius(float radius);

float getRadius();

float getVolumeBall();

private:

float radius = 0.0;

};

ClassLab12\_Zadorozhnyi::ClassLab12\_Zadorozhnyi(float radius){

this->radius = radius;

}

void ClassLab12\_Zadorozhnyi::setRadius(float radius){

this->radius = radius;

}

float ClassLab12\_Zadorozhnyi::getRadius(){

return radius;

}

float ClassLab12\_Zadorozhnyi::getVolumeBall(){

return (4 \* M\_PI \* pow(radius, 3))/3;

}

#endif // MODULESZADOROZHNYI\_H\_INCLUDED

**Лістинг main.cpp**

#include <iostream>

#include <fstream>

#include <vector>

#include <sstream>

#include <cstring>

#include <cmath>

#include "../ModulesZadorozhnyi.h"

using namespace std;

struct TestSuite{

float action = 0;

float expResult = 0;

};

vector<TestSuite> readCVSFiles(string filePath){

ifstream testSuite (filePath);

vector<string> row;

vector<TestSuite> autotest;

TestSuite ts;

string line, value;

getline(testSuite, line);

while(getline(testSuite, line)){

row.clear();

stringstream s(line);

while(getline(s, value, ',')){

row.push\_back(value);

}

ts.action = stof(row[0]);

ts.expResult = stof(row[1]);

autotest.push\_back(ts);

}

testSuite.close();

return autotest;

}

int main()

{

ofstream testResult;

string pathFile = \_\_FILE\_\_;

size\_t found = pathFile.find("\\Lab12\\prj");

if(found == string::npos){

testResult.open("TestResult.txt");

if(!testResult.is\_open()) return -1;

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

cout << '\a';

}

testResult << "Встановлені вимоги порядку виконання лабораторної роботи порушено!";

}

else{

testResult.open("..\\..\\TestSuite\\TestResult.txt");

if(!testResult.is\_open()) return -1;

ClassLab12\_Zadorozhnyi ball(10);

vector<TestSuite> autotest = readCVSFiles("..\\..\\TestSuite\\TS.txt");

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

ball.setRadius(autotest[i].action);

if(round(autotest[i].expResult) == round(ball.getVolumeBall())){

testResult << "test №" << i+1 << " -> passed" << endl << endl;

}

else{

testResult << "test №" << i+1 << " -> failed" << endl

<< "answers don't match" << endl

<< "It should be -> " << autotest[i].expResult << endl

<< "Received response -> " << ball.getVolumeBall() << endl << endl;

}

}

}

testResult.close();

system("pause");

return 0;

}

**ДОДАТОК Б**

|  |  |
| --- | --- |
| 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 | Задорожний Костянтин |
| Виконавець  Implementer | Задорожний Костянтин |

|  |  |  |  |
| --- | --- | --- | --- |
| 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) |
| TS-01 | Ввести 5.25 | 606.131 | Passed |
| TS-02 | Ввести 15.5 | 71518.8 | Passed |
| TS-03 | Ввести 25.75 | 195432 | Passed |
| TS-04 | Ввести 36 | 195432 | Passed |
| TS-05 | Ввести 46.25 | 414404 | Passed |