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

ВИКОНАВ

студент академічної групи КІ-21-2

\_\_\_\_\_\_\_\_\_ Мельник А.М

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_ Павло УСІК

Кропивницький – 2022

**ЛАБОРАТОРНА РОБОТА №12**

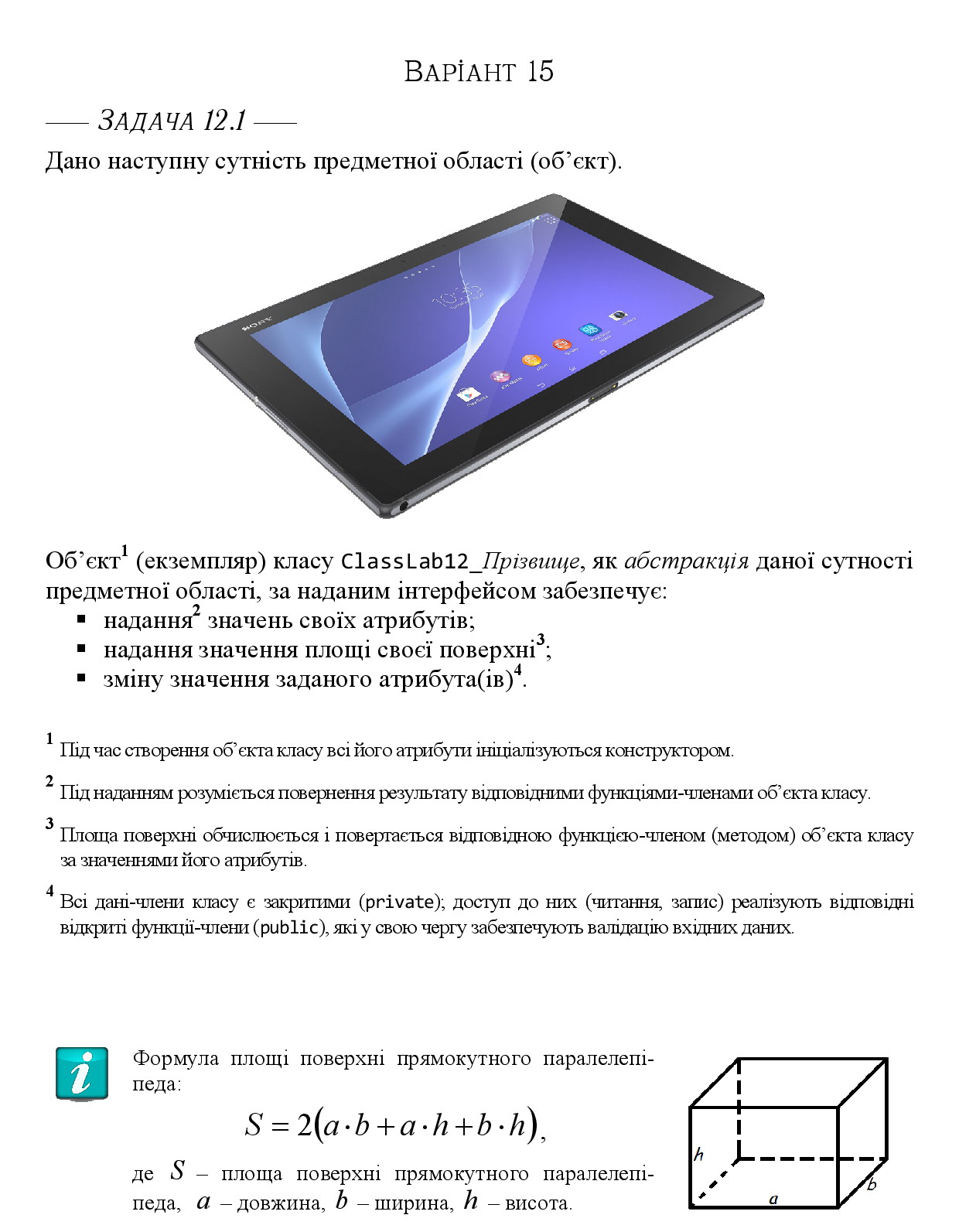
**ТЕМА:** програмна реалізація абстрактних типів даних

**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.

**ВАРІАНТ 6**

**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



**Концептуалізація предметної області:**

За умовою завдання, дано візуалізацію об’єкта, структуру якого можна інтерпретувати як “планшет”.Однак для значень довжини, ширини та висоти потрібно обмежитися прямокутним паралелепіпедом.

Визначення всіх сутностей об'єкта описують довжина, ширина та висота. Обмеження та допущення вводяться випадково.

**Вимоги до програмного модуля:**

Властивості об’єкта (радіус – в якості поля) оголошуються на рівні private. Методи отримання значень з рівня private (функції мають префікс get в ідентифікаторі) та методи ініціалізації поля класу (функція, що мають префікс set в ідентифікаторі) оголошуються на рівні public.

Поля класу за замовчуванням ініціалізуються 25.0 (в метричній системі), що надалі використовуються як обмеження або виявлення некоректної ініціалізації.

**Тестові артефакти:**

Завдяки текстовому файлу TS.txt вводимо всі значення ширини, висоти та довжини, а за допомогою Teaсher виводимо результати в TestResults.txt.

Зчитування тестового файлу здійснюється двома методами: у випадку, якщо зчитування неможливе, до текстового файлу TestResults.txt записується “Встановлені вимоги порядку виконання лабораторної роботи порушено!” та завершується робота застосунку; якщо зчитування вдале - відбувається Unit-тестування.

Вміст файлу TS.txt:

tabletHeight: 2.0

tabletLength: 6.0

tabletWidth: 4.0

tabletSquare: 30.0

tabletHeight: 5.8

tabletLength: 4.4

tabletWidth: 9.05

tabletSquare: 40.0

tabletHeight: 6.0

tabletLength: 2.7

tabletWidth: 6.87

tabletSquare: 94.518

tabletHeight: 12.32

tabletLength: 43.29

tabletWidth: 38.8

tabletSquare: 900.4

tabletHeight: 54.7

tabletLength: 6.3

tabletWidth: 12.8

tabletSquare: 230.90

tabletHeight: 2.6

tabletLength: 6.2

tabletWidth: 4.9

tabletSquare: 36.0

tabletHeight: 2.45

tabletLength: 9.36

tabletWidth: 7.25

tabletSquare: 89.62

**Аналіз та архітектура завдання 12.2:**

На початку роботи, програма перевіряє, чи знаходиться файл у потрібній папці. Далі створюється об’єкт класу та відкривається файл TS.txt для зчитування та TestResults.txt для дозапису, передньо очищуючи всю попередню інформацію. Далі відбувається чи відкриті ці текстові файли.

Цикл у Techer/main.cpp створюються для запису у текстовий файл TestResults.txt, інший створений для видалення 14 перших символів з TS.txt.

Відбувається зчитування з TS.txt в TestResults.txt файл.

**ВИСНОВКИ**

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.

За умовою завдання, дано візуалізацію об’єкта, структуру якого можна інтерпретувати як “планшет”. Однак для значень довжини, ширини та висоти потрібно обмежитися прямокутним паралелепіпедом.

Визначення всіх сутностей об'єкта описують довжина, ширина та висота. Обмеження та допущення вводяться випадково.

Властивості об’єкта (радіус – в якості поля) оголошуються на рівні private. Методи отримання значень з рівня private (функції мають префікс get в ідентифікаторі) та методи ініціалізації поля класу (функція, що мають префікс set в ідентифікаторі) оголошуються на рівні public.

Поля класу за замовчуванням ініціалізуються 25.0 (в метричній системі), що надалі використовуються як обмеження або виявлення некоректної ініціалізації.

Завдяки текстовому файлу TS.txt вводимо всі значення ширини, висоти та довжини, а за допомогою Teaсher виводимо результати в TestResults.txt.

Зчитування тестового файлу здійснюється двома методами: у випадку, якщо зчитування неможливе, до текстового файлу TestResults.txt записується “Встановлені вимоги порядку виконання лабораторної роботи порушено!” та завершується робота застосунку; якщо зчитування вдале - відбувається Unit-тестування.

На початку роботи, програма перевіряє, чи знаходиться файл у потрібній папці. Далі створюється об’єкт класу та відкривається файл TS.txt для зчитування та TestResults.txt для дозапису, передньо очищуючи всю попередню інформацію. Далі відбувається чи відкриті ці текстові файли.

Цикл у Techer/main.cpp створюються для запису у текстовий файл TestResults.txt, інший створений для видалення 14 перших символів з TS.txt.

Відбувається зчитування з TS.txt в TestResults.txt файл.

Під час виконання роботи було створенно декілька додаткових функцій:

* fileInDirectory():робить перевірку наявності файлу в теці та містить реалізацію вимог другого завдання.
* declareTestSuiteResults(): запис результатів тестування у текстовий файл TestResults.txt.

Під час виконання 12 лабораторної роботи було прийнято рішення, що після виконання всієї роботи я вже готові папки завантажувала через сайт. Також виникла проблема при читанні коду: замість кирилиці написано символи, які не стосуються роботи, це через те, що робота для зручності використовує кодування Windows1251, але для читання роботи на git hubкраще використовувати UTF-8.

Для цього спочатку потрібно головний файл проекту відрити у блокноті та зберегти з кодуванням UTF-8 (раніше було ANSI).Також в коді потрібно підключити “system (chcp 866 & cls)”, і в налаштуваннях editor налаштувати на кодування UTF-8.

За допомогою цієї лабораторної роботи я навчилася працювати з класами та тестуванні їх екземплярів.

**ДОДАТОК А**

(набір тестових результатів TestResults.txt)

----------------------------------------------------

| TEST CASE 1 |

| (input) Tablet length: 6 a |

| (input) Tablet width: 4 b |

| (input) Tablet height: 2 h |

| (output) Tablet square: 30.0 S |

| (expected) Tablet square: 68 S |

| TEST CASE RESULT: false |

----------------------------------------------------

----------------------------------------------------

| TEST CASE 2 |

| (input) Tablet length: 4.4 a |

| (input) Tablet width: 9.05 b |

| (input) Tablet height: 5.8 h |

| (output) Tablet square: 40.0 S |

| (expected) Tablet square: 157.65 S |

| TEST CASE RESULT: false |

----------------------------------------------------

----------------------------------------------------

| TEST CASE 3 |

| (input) Tablet length: 2.7 a |

| (input) Tablet width: 6.87 b |

| (input) Tablet height: 6 h |

| (output) Tablet square: 94.518 S |

| (expected) Tablet square: 94.518 S |

| TEST CASE RESULT: true |

----------------------------------------------------

----------------------------------------------------

| TEST CASE 4 |

| (input) Tablet length: 43.29 a |

| (input) Tablet width: 38.8 b |

| (input) Tablet height: 12.32 h |

| (output) Tablet square: 900.4 S |

| (expected) Tablet square: 4370.65 S |

| TEST CASE RESULT: false |

----------------------------------------------------

----------------------------------------------------

| TEST CASE 5 |

| (input) Tablet length: 6.3 a |

| (input) Tablet width: 12.8 b |

| (input) Tablet height: 54.7 h |

| (output) Tablet square: 230.90 S |

| (expected) Tablet square: 1206.05 S |

| TEST CASE RESULT: false |

----------------------------------------------------

----------------------------------------------------

| TEST CASE 6 |

| (input) Tablet length: 6.2 a |

| (input) Tablet width: 4.9 b |

| (input) Tablet height: 2.6 h |

| (output) Tablet square: 89.62 S |

| (expected) Tablet square: 89.62 S |

| TEST CASE RESULT: true |

----------------------------------------------------

**ДОДАТОК Б**

(Вихідний код ModulesMelnik.h)

#ifndef MODULESMELNYK\_H\_INCLUDED

#define MODULESMELNYK\_H\_INCLUDED

class ClassLab12\_Melnyk{

public:

ClassLab12\_Melnyk(){}

ClassLab12\_Melnyk(float height, float wight, float lenght);

void setTabletHeight(float height);

float getTabletHeight();

void setTabletWidth(float width);

float getTabletWidth();

void setTabletLength(float length);

float getTabletLength();

float getTabletSquare();

private:

float height = 1.0;

float width = 10.0;

float length = 20.0;

};

ClassLab12\_Melnyk::ClassLab12\_Melnyk(float height, float wight, float lenght){

this->height = height;

this->length = length;

this->width = width;

}

//set

void ClassLab12\_Melnyk::setTabletHeight(float height){

this->height = height;

}

void ClassLab12\_Melnyk::setTabletLength(float length){

this->length = length;

}

void ClassLab12\_Melnyk::setTabletWidth(float width){

this->width = width;

}

//get

float ClassLab12\_Melnyk::getTabletHeight(){

return height;

}

float ClassLab12\_Melnyk::getTabletLength(){

return length;

}

float ClassLab12\_Melnyk::getTabletWidth(){

return width;

}

float ClassLab12\_Melnyk::getTabletSquare(){

return (2 \* (length \* width) + length \* height + width \* height);

}

#endif // MODULESMELNYK\_H\_INCLUDED

**ДОДАТОК В**

(Вихідний код HelperFunctions.h)

#ifndef HELPERFUCTIONS\_H\_INCLUDED

#define HELPERFUCTIONS\_H\_INCLUDED

#include "ModulesMelnyk.h"

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

void declareTestSuiteResults(ofstream &resFile, ClassLab12\_Melnyk Tablet, string outArea, string expArea, int index)

{

resFile << " ---------------------------------------------------- " << endl

<< "| TEST CASE " << setw(2) << index << setw(38) << "|" << endl

<< "| (input) Tablet length: " << setw(20) << Tablet.getTabletLength() << " a |" << endl

<< "| (input) Tablet width: " << setw(20) << Tablet.getTabletWidth() << " b |" << endl

<< "| (input) Tablet height: " << setw(20) << Tablet.getTabletHeight() << " h |" << endl

<< "| (output) Tablet square: " << setw(20) << outArea << " S |" << endl

<< "| (expected) Tablet square: " << setw(20) << Tablet.getTabletSquare() << " S |" << endl;

resFile << "| TEST CASE RESULT: " << setw(20) << boolalpha << (expArea.compare(outArea) == 0) << " |" << endl

<< " ---------------------------------------------------- " << endl;

}

#endif // HELPERFUCTIONS\_H\_INCLUDED

**… /Teacher / main.cpp**

#include "HelperFuctions.h"

using namespace std;

bool fileInDirectory()

{

string cpp = \_\_FILE\_\_;

size\_t found = cpp.find("\\lab12\\prj");

cout << cpp << endl;

if (found == string::npos) {

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

cout << "\a";

}

ofstream resFile("../../TestSuite/TestResults.txt");

resFile << "Встановлені вимоги порядку виконання лабораторної роботи порушено!" << endl;

resFile.close();

return false;

}

return true;

}

int main()

{

if (!fileInDirectory()) {

cout << "Встановлені вимоги порядку виконання лабораторної роботи порушено!" << endl;

return -1;

}

ClassLab12\_Melnyk Tablet;

ifstream caseFile("../../TestSuite/TS.txt");

if(!caseFile.is\_open()){ cout << "TS.txt is not opened" << endl; return -2; };

ofstream resultFile("../../TestSuite/TestResults.txt");

if(!resultFile.is\_open()){ cout << "TestResults.txt is not opened" << endl; return -3; };

for (int i = 1; i < 7; i++) {

string tabletHeight;

string tabletLength;

string tabletWidth;

string tabletSquare;

string bufLine;

float bufFloat;

getline(caseFile, bufLine);

tabletHeight = bufLine.erase(0, 14);

getline(caseFile, bufLine);

tabletLength = bufLine.erase(0, 14);

getline(caseFile, bufLine);

tabletWidth = bufLine.erase(0, 14);

getline(caseFile, bufLine);

tabletSquare = bufLine.erase(0, 14);

Tablet.setTabletHeight(stof(tabletHeight));

Tablet.setTabletLength(stof(tabletLength));

Tablet.setTabletWidth(stof(tabletWidth));

declareTestSuiteResults(resultFile, Tablet, bufLine, tabletSquare, i);

}

caseFile.close();

resultFile.close();

system("pause");

return 0;

}