

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
КБ-23

Литвин М.В.

ПЕРЕВІРИВ
викладач кафедри кібербезпеки
та програмного забезпечення

Дреєва Г.М.

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.

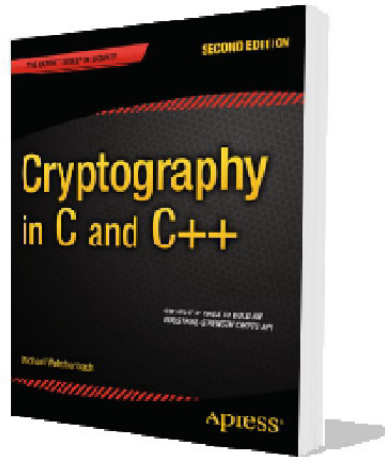
Завдання:

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 проекту 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.

ВАРІАНТ № 6

— ЗАДАЧА 12.1 —

Дано наступну сутність предметної області (об'єкт).



Об'єкт¹ (екземпляр) класу `ClassLab12_Прізвище`, як абстракція даної сутності предметної області, за наданим інтерфейсом забезпечує:

- надання² значень своїх атрибутів;
- надання значення площі³ своєї палітурки⁴;
- зміну значення заданого атрибута(ів)⁵.

¹ Під час створення об'єкта класу всі його атрибути ініціалізуються конструктором.

² Під наданням розуміється повернення результату відповідними функціями-членами об'єкта класу.

³ Об'єм обчислюється і повертається відповідною функцією-членом (методом) об'єкта класу за значеннями його атрибутів.

⁴ Палітурка або оправа — цупка, захисна обкладинка книжки.

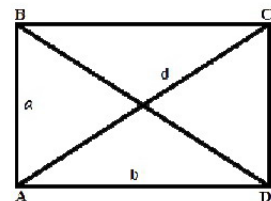
⁵ Всі дані-члени класу є закритими (`private`); доступ до них (читання, запис) реалізують відповідні відкриті функції-члени (`public`), які у свою чергу забезпечують валідацію вхідних даних.



Площа прямокутника дорівнює добутку довжин двох його суміжних сторін:

$$S = a \cdot b,$$

де S — площа, a і b — довжини сторін прямокутника.



Аналіз задачі 12.1

Зчитувати ми будемо всю інформацію з вхідного файлу, а записувати тільки до вихідного файлу. Ці файли повинні були формату .txt.

Постановка задачі 12.1

Для цієї задачі нам потрібно розробити в .h файлі динамічний об'єкт «Книга», який буде мати приватні параметри, такі як: довжина та ширина, також матиме публічні методи за допомогою яких можна задати дані параметрам, або обчислити площу.

Аналіз задачі 12.2

Протестувати динамічний об'єкт, який був створено у попередній задачі, й доробити декілька функцій, функції завдання будуть знаходитися у файлі зі класом.

Постановка задачі 12.2

Треба розробити тестовий драйвер, який буде автоматично зчитувати дані з .txt файлу, після чого заносити їх до динамічного списку, а після цього обчислювати їх за допомогою створених методів класу, й записувати результат у вихідний файл.

Додаток А

Лістинг задач 12.1:

```
#ifndef MODULESLYTVYN_H_INCLUDED
#define MODULESLYTVYN_H_INCLUDED
// Цей кодовий блок забезпечує багаторазове включення файлу заголовка,
використовуючи захист від повторного включення.

#include <fstream>
#include <string>
#include <iostream>
// Підключаємо бібліотеки для роботи з файлами, рядками і стандартного вводу-
виводу.

void HelloWorld() {
    std::cout << "Hello world";
}
// Функція HelloWorld виводить "Hello world" на консоль.

class ClassLab12_Lytvyn {
public:
    ClassLab12_Lytvyn(const float length = 1, const float width = 1)
        : Length(length), Width(width) {}
    // Конструктор класу з параметрами за замовчуванням для довжини і ширини.
    // Ініціалізує члени класу Length і Width.

    float getArea() const;
    // Метод для отримання площі об'єкта.

    float getLength() const;
    // Метод для отримання значення довжини об'єкта.

    float getWidth() const;
```

```

    // Метод для отримання значення ширини об'єкта.

void setLength(float length);
// Метод для встановлення значення довжини об'єкта.

void setWidth(float width);
// Метод для встановлення значення ширини об'єкта.

private:
    float Length;
    // Приватний член класу, що зберігає значення довжини.

    float Width;
    // Приватний член класу, що зберігає значення ширини.
};

float ClassLab12_Lytvyn::getArea() const {
    return Length * Width;
}
// Реалізація методу getArea, який обчислює і повертає площу об'єкта.

float ClassLab12_Lytvyn::getLength() const {
    return Length;
}
// Реалізація методу getLength, який повертає значення довжини об'єкта.

float ClassLab12_Lytvyn::getWidth() const {
    return Width;
}
// Реалізація методу getWidth, який повертає значення ширини об'єкта.

void ClassLab12_Lytvyn::setLength(float length) {
    Length = length;
}
// Реалізація методу setLength, який встановлює значення довжини об'єкта.

void ClassLab12_Lytvyn::setWidth(float width) {
    Width = width;
}
// Реалізація методу setWidth, який встановлює значення ширини об'єкта.

bool checkFileInFolder() {
    std::string file = __FILE__;
    size_t checking = file.find("\\prj");
    if (checking == std::string::npos) {
        std::ofstream fileResult("../TestSuite/TestResult.txt");
        fileResult << "Встановлені вимоги порядку виконання лабораторної роботи
порушено!\n";
        for (int i = 0; i < 100; i++) {
            std::cout << "a";
        }
        fileResult.close();
        return false;
    }
    return true;
}
// Функція checkFileInFolder перевіряє, чи міститься шлях до файлу у певній папці
(\prj).
// Якщо ні, створюється файл з повідомленням про порушення вимог, і програма
повертає false.

std::string writeResultToFile(const ClassLab12_Lytvyn& book, int counter) {
    return "Test Suite No" + std::to_string(counter)
        + "\nLength = " + std::to_string(book.getLength())
        + "\tWidth = " + std::to_string(book.getWidth())
        + "\nArea = " + std::to_string(book.getArea()) + "\n";
}

```

```

}
// Функція writeResultToFile формує рядок з результатами тестування об'єкта book,
// включаючи його довжину, ширину та площу, і нумерує тестовий набір за допомогою
counter.

#endif
// Кінець блоку захисту від повторного включення файлу заголовка.

```

Лістинг Teacher main.cpp:

```

#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include "ModulesLytvyn.h"
// Підключаємо необхідні бібліотеки для роботи з потоками введення/виведення,
рядками, векторами
// і заголовковий файл з оголошенням класу ClassLab12_Lytvyn.

using namespace std;

int main() {
    string line;
    if (!checkFileInFolder()) {
        return 0;
    }
    // Перевіряємо, чи файл знаходиться в потрібній папці за допомогою функції
checkFileInFolder.
    // Якщо файл не знайдено, програма завершується.

    ClassLab12_Lytvyn book;
    ifstream Tests_File("../TestSuite/Tests.txt");
    ofstream TestResult_File("../TestSuite/TestResult.txt");
    // Створюємо об'єкт book класу ClassLab12_Lytvyn.
    // Відкриваємо файл для зчитування тестових даних (Tests_File) і файл для
запису результатів (TestResult_File).

    if (Tests_File.is_open() && TestResult_File.is_open()) {
        int counter = 1;
        float number;
        while (getline(Tests_File, line)) {
            istringstream iss(line);
            vector<float> arrNumbers;
            while (iss >> number) {
                arrNumbers.push_back(number);
            }
            // Читаємо кожен рядок з файлу Tests.txt, розбиваємо його на окремі
числа і зберігаємо у векторі arrNumbers.

            book.setLength(arrNumbers[0]);
            book.setWidth(arrNumbers[1]);
            // Встановлюємо значення довжини і ширини об'єкта book з вектора
arrNumbers.

            TestResult_File << writeResultToFile(book, counter);
            counter++;
            // Записуємо результати тестування об'єкта book у файл TestResult.txt,
            // додаючи номер тесту, і збільшуємо лічильник counter.
        }
    }

    Tests_File.close();
    TestResult_File.close();
    // Закриваємо файли після завершення роботи.
    return 0;
    // Програма повертає 0, що вказує на успішне завершення.

```


Висновок: Під час виконання лабораторної роботи № 12 на тему «Програмна реалізація абстрактних типів даних» з предмету «Базові Методології та Технології Програмування» я відпрацював на практиці роботу з класами та засвоїв тестування динамічних типів даних за допомогою модульного тестування. Набув ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування C++ програмних модулів створення й оброблення даних типів масив, структура, об'єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек для оброблення символічної інформації.

В ході лабораторної роботи було на практиці продемонстровано роботу с динамічними об'єктами пам'яті та використано бібліотеки для оброблення текстової інформації. В цілому, робота з класами та тестування їх у C++ є важливими етапами розробки програмного забезпечення, що дозволяють забезпечити правильність та надійність коду.

Додаток Б

Назва тестового набору Test Suite Description	TestSuite
Назва проекту / ПЗ Name of Project / Software	Teacher.exe
Рівень тестування Level of Testing	модульне / Unit Testing
Автор тест-сьюта Test Suite Author	Литвин Максим Вікторович

Test Case ID	Action	Expected Result	Test Result
TC-01	Length = 1, width = 1	$S = 1$	passed
TC-02	Length = 2, width = 3	$S = 6$	passed
TC-03	Length = 0, width = 5	$S = 0$	passed
TC-04	Length = 3, width = 2.5	$S = 7.5$	passed
TC-05	Length = 10, width = 10	$S = 100$	passed
TC-06	Length = 0.5, width = 0.5	$S = 0.25$	passed
TC-07	Length = 7, width = 8	$S = 56$	passed
TC-08	Length = 1.1, width = 2.2	$S = 2.42$	passed
TC-09	Length = 4, width = 4	$S = 16$	passed
TC-10	Length = 6.5, width = 3.2	$S = 20.8$	passed

Додаток В

Зміст вхідного файлу:

1 1

2 3

0 5

3 2.5

10 10

0.5 0.5

7 8

1.1 2.2

4 4

6.5 3.2

Зміст вихідного файлу:

Test Suite No1

Length = 1 Width = 1

Area = 1

Test Suite No2

Length = 2 Width = 3

Area = 6

Test Suite No3

Length = 0 Width = 5

Area = 0

Test Suite No4

Length = 3 Width = 2.5

Area = 7.5

Test Suite No5

Length = 10 Width = 10

Area = 100

Test Suite No6

Length = 0.5 Width = 0.5

Area = 0.25

Test Suite No7

Length = 7 Width = 8

Area = 56

Test Suite No8

Length = 1.1 Width = 2.2

Area = 2.42

Test Suite No9

Length = 4 Width = 4

Area = 16

Test Suite No10

Length = 6.5 Width = 3.2

Area = 20.8