

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 22-2

_____ Горбачов Є. І.

ПЕРЕВІРИВ
викладач кафедри кібербезпеки
та програмного забезпечення

_____ Олександр СОБІНОВ

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-сютом(ами) із \Lab12\TestSuite\, протоколюючи результати тестування в текстовий файл \Lab12\TestSuite\TestResults.txt.

Варіант 2

Задача 12.1:

ВАРІАНТ № 2

— 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:

- падашся² значення своїх атрибутів;
- падашся значення свого об'єму³;
- зміну значення заданого атрибута (ів)⁴.

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

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

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

⁴ 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.

Аналіз задачі 12.1

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

Постановка задачі 12.1

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

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 MODULESHORBACHOV_H_INCLUDED
#define MODULESHORBACHOV_H_INCLUDED
#include <fstream>
#include <string>

void HelloWorld()
{
    std::cout << "Hello wolrd";
}

class ClassLab12_Horbachov
{
public:
    ClassLab12_Horbachov(const float length = 1, const float width = 1, const
float height = 1) :
        Length(length),
        Width(width),
        Height(height)
    {

    }
    float getV();
    float getLength();
    float getWidth();
    float getHeight();
    float changeLength(float length);
    float changeWidth(float width);
    float changeHeight(float height);

private:
    float Length;
    float Width;
    float Height;
};

float ClassLab12_Horbachov::getV()
{
    return Length * Width * Height;
}

float ClassLab12_Horbachov::changeLength(float length)
{
    Length = length;
}
```

```

float ClassLab12_Horbachov::changeWidth(float width)
{
    Width = width;
}

float ClassLab12_Horbachov::changeHeight(float height)
{
    Height = height;
}

float ClassLab12_Horbachov::getLength()
{
    return Length;
}

float ClassLab12_Horbachov::getHeight()
{
    return Height;
}

float ClassLab12_Horbachov::getWidth()
{
    return Width;
}

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;
}

std::string writeResultToFile(ClassLab12_Horbachov akvarium, int counter)
{
    return "Test Suite №" + std::to_string(counter)
        + "\nLength = " + std::to_string(akvarium.getLength())
        + "\theight = " + std::to_string(akvarium.getHeight())
        + "\twidth = " + std::to_string(akvarium.getWidth())
        + "\nResult = " + std::to_string(akvarium.getV()) + "\n";
}

#endif // MODULESHORBACHOV_H_INCLUDED

```

Лістинг Teacher main.cpp:

```

#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include "ModulesHorbachov.h"

using namespace std;

```

```

int main()
{
    string line;
    if(checkFileInFolder() == false)
        return 0;

    ClassLab12_Horbatchov akvarium;
    ifstream Tests_File("../TestSuite/Tests.txt");
    ofstream TestResult_File("../TestSuite/TestResult.txt");

    if (Tests_File.is_open() || TestResult_File.is_open())
    {
        int counter = 1;
        float number;
        while (getline(Tests_File, line, '\n'))
        {
            istringstream iss(line);
            vector<float> arrNumbers;

            while (iss >> number)
                arrNumbers.push_back(number);

            akvarium.changeLength(arrNumbers[0]);
            akvarium.changeWidth(arrNumbers[1]);
            akvarium.changeHeight(arrNumbers[2]);
            TestResult_File << writeResultToFile(akvarium, counter);
            counter++;
        }
    }
    Tests_File.close();
    TestResult_File.close();
    system("pause");
    return 0;
}

```

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 символьної інформації. Під час лабораторної роботи було виконано такі пункти:

- Завантажено завдання №12 із репозиторію за допомогою команди `git pull`
- Після цього було створено папки `prj`, `Software`, `Test Suite`, та `Report` і заповнено `README.md` файл.
- Після цього було відправлено ці дані за допомогою команд :
`git add -A`
`git commit -m "Add folder prj, Software, Test Suite, Report and feeling README file"`
`git push`
- Почергово було виконано аналіз і постановку задач 12.1 і 12.2, аналіз вимог до ПЗ та вмісту вхідного текстового файлу, проектування архітектури, детальне проектування програмних модулів; одержані артефакти задокументувати й включити до звіту
- Далі було розроблено один тест-сюет задля проведення автоматизованого unit-тестування заголовкового файлу розв'язування задачі 12.1. Тестові артефакти було задокументовано та включено до звіту як ДОДАТОК Б
- За допомогою отриманих під час проектування програмних модулів артефактами виконати конструювання функцій: мовою програмування C++ було реалізовано функції, які реалізують розв'язування задач 12.1 і 12.2.
- Скомпільовано проект заголовкового фалу `Teacher` та завантажено на `git repository`
- Потім мовою програмування C++ було реалізовано консольний застосунок – тестовий драйвер для модульного тестування функцій розв'язування задач 12.1 за

допомогою розробленого тест-сьюту з \Lab12\TestSuite та вхідного і/або вихідного текстового файлу

- Створений застосунок Teacher.exe переміщено у \Lab10\Teacher
- За допомогою Teacher.exe було виконано автоматизоване тестування розроблених функцій розв'язування задач 12.1 та 12.2.
- Вихідний код (текст) проектів Teacher та ModulesHorbachov.h включено в звіт як ДОДАТОК А .

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	Горбачов Євгеній Ігорович

Test Case ID	Action	Expected Result	Test Result
TC-01	Length = 1, width = 10, height = 2	V = 20	passed
TC-02	Length = 2, width = 3, height = 5	V = 30	passed
TC-03	Length = 3, width = 6, height = 19.2	V = 345.600006	passed
TC-04	Length = 4, width = 76, height = 7	V = 2128	passed
TC-05	Length = 5, width = 5, height = 6	V = 150	passed
TC-06	Length = 6, width = 0, height = 1	V = 0	passed
TC-07	Length = 7, width = 100, height = 1	V = 700	passed
TC-08	Length = 8, width = 7, height = 8	V = 448	passed
TC-09	Length = 9, width = 90, height = 0.5	V = 405	passed
TC-10	Length = 10, width = 7, height = 4	V = 280	passed

Додаток В

Зміст вхідного файлу:

1 10 2
2 3 5
3 6 19.2
4 76 7
5 5 6
6 0 1
7 100 1
8 7 8

9 90 0.5

10 7 4

Зміст вихідного файлу:

Test Suite №1

Length = 1.000000 height = 2.000000 width = 10.000000

Result = 20.000000

Test Suite №2

Length = 2.000000 height = 5.000000 width = 3.000000

Result = 30.000000

Test Suite №3

Length = 3.000000 height = 19.200001 width = 6.000000

Result = 345.600006

Test Suite №4

Length = 4.000000 height = 7.000000 width = 76.000000

Result = 2128.000000

Test Suite №5

Length = 5.000000 height = 6.000000 width = 5.000000

Result = 150.000000

Test Suite №6

Length = 6.000000 height = 1.000000 width = 0.000000

Result = 0.000000

Test Suite №7

Length = 7.000000 height = 1.000000 width = 100.000000

Result = 700.000000

Test Suite №8

Length = 8.000000 height = 8.000000 width = 7.000000

Result = 448.000000

Test Suite №9

Length = 9.000000 height = 0.500000 width = 90.000000

Result = 405.000000

Test Suite №10

Length = 10.000000 height = 4.000000 width = 7.000000

Result = 280.000000