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 КН-24

Безкровна В. В.

ПЕРЕВІРИЛА

викладачка кафедри кібербезпеки   
та програмного забезпечення

Анастасія КОВАЛЕНКО

Кропивницький – 2025

**ТЕМА: Програмна реалізація абстрактних типів даних**

**МЕТА: набуття 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 – 4**

**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 12.1.
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 проєкта 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.

**Хід роботи**

**Концептуалізація:** предметною областю є геометрична фігура – конус.

**Об’єктний аналіз:** об’єкт, це святковий ковпак (конус); він містить такі атрибути, як радіус основи (radius), висота (height) та колір (color); всі атрибути є приватними.

**Визначення інтерфейсів сутності:** для доступу до даних-членів класі здійснюється за допомогою відкритих функцій-членів; під час створення об’єкта класу всі його атрибути ініціалізуються конструктором; функції GetRadius, GetHeight та GetColor надають значення атрибутів; функції SetRadius, SetHeigh, SetColor заміняють значення заданих атрибутів; функція CountVolume обчислює значення об’єма конуса в залежності від його атрибутів.

**Аналіз задачі 12.1:**

Дано: сутність предметної області — геометричний об’єкт "конус".

Реалізувати: клас ClassLab12\_Bezkrovna; конструктор класу, методи для надання значень атрибутів, обчислення об’єму, заміни значень атрибутів.

**Постановка задачі 12.1:**

Вхід: Cone(radius, height, color).

Вихід: radius, height, color, volume.

**Функціональні вимоги задачі 12.1:**

Програмний модуль повинен реалізовувати клас, який моделює конус. Клас має забезпечувати:

* Ініціалізацію всіх параметрів об'єкта при створенні через конструктор.
* Отримання значень параметрів.
* Обчислення та повернення об’єму за заданими параметрами.
* Зміну значень параметрів з перевіркою коректності введених даних.

**Нефункціональні вимоги задачі 12.1:**

* Всі члени класу є закритими (private).
* Доступ до них відбувається через відкриті функції-члени (public).

**Проектування архітектури програмного модуля задачі 12.1:**

Клас ClassLab12\_Bezkrovna містить:

* Приватні атрибути, що містять радіус основи конуса, висоту та колір.
* Публічні функції-члени для надання значень атрибутів, обчислення об’єму, заміни значень атрибутів.

**Детальне проєктування програмного модуля задачі 12.1:**

Клас ClassLab12\_Bezkrovna складається з:

Рівня private:

* double radius – радіус основи.
* double height – висота.
* string color – колір.

Рівня public:

* ClassLab12\_Bezkrovna (double height, double radius, string color) – конструктор класу.
* double GetHeight() – надає поточне значення radius.
* double GetRadius() – надає поточне значення height.
* string GetColor() – надає поточне значення color.
* void SetHeight(float input\_height) – встановлює нове значення radius.
* void SetRadius(float input\_radius) – встановлює нове значення height.
* void SetColor(string input\_color) – встановлює нове значення color.
* float CountVolume() – обчислює та надає значення об’єму.

**Аналіз задачі Teacher:**

Дано: текстовий файл з 10 рядками, які містять висоту, радіус, очікуваний об’єм та колір.

Виконати: тестування правильності розрахунків, які виконує клас ClassLab12\_Bezkrovna, за допомогою обчислення об’му та порівняння з очікуваним об’ємом.

**Постановка задачі Teacher:**

Вхід: radius, height, volume, color

Вихід: result\_volume

**Формат тест-кейсів:**

Кожен рядок текстового файлу містить вхідні значення height, radius, volume та color для кожного тест кейсу, без розділових знаків.

**Протокол читання тест-кейсів:**

По одному зчитуються рядки з текстового файлу. Для поточного рядка створюється потік. Послідовно зчитуються три числа в змінні height, radius і volume. Решта рядка читається до кінця в змінну color.

**Вимоги задачі Teacher:**

* Програма має читати вхідні дані з файлу TestCase.txt.
* Кожен рядок файлу містить: висоту (height), радіус (radius), об'єм (volume) та колір (color) конуса.
* Записати результат тесту в файл TestResults.txt у вигляді:
  + Номер тесту;
  + Обчислений об’єм;
  + Статус ("passed" або "failed").
* Програма перевіряє чи знаходився файл проєкта main.срр під час його компіляції знаходився не в \Lab12\prj.

**Проектування архітектури задачі Teacher:**

* Функція is\_in\_lab12 перевіряє чи знаходився файл проєкта main.срр під час його компіляції знаходився не в \Lab12\prj.
* Функція check\_main\_location виводить рядок "Встановлені вимоги порядку виконання лабораторної роботи порушено!" в текстовий файл TestResults.tx та видає 100 звукових сигналів, якщо файл проєкта main.срр під час його компіляції знаходився не в \Lab12\prj.
* Гловна функція main() керує процесом тестування.

**Детальне проєктування задачі Teacher:**

* Функція is\_in\_lab12 приймає шлях до файлу, перевіряє наявність підрядка "\\Lab12\\prj" у шляху, повертає true або false.
* Функція check\_main\_location() використовує \_\_FILE\_\_, якщо файл не у потрібному каталозі, виконує звуковий сигнал 100 разів та запис у файл TestResults.txt з повідомленням про помилку, повертає true якщо розташування правильне, інакше — false.
* Головна функція main() викликає check\_main\_location(), відкриває файли TestCase.txt для читання та TestResults.txt для запису, для кожного рядка у вхідному файлі зчитує height, radius, volume і color, створює об’єкт ClassLab12\_Bezkrovna, обчислює об’єм, записує номер тесту, обчислений об’єм і статус тесту до файлу.

**Висновок:** в цій лабораторный роботі ми набули 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.

Ми досягли мети роботи:

1. Виконавши самостійну підготовку до виконання лабораторної роботи №12 для успішного виконання завдань.
2. Охайно оформивши роботу.
3. Аналізувавши мету роботи для правильного виконання лабораторної роботи.
4. Завантаживши копію власного Git-репозиторію.
5. Повторивши застосування команд git fetch та git merge на практиці.
6. Пригадавши використання команд git add, git commit, git merge для Git.
7. Створивши теки необхідні для роботи.
8. Пригадавши теоретичний метод пізнання – абстрагування
9. Пригадавши процес об’єктної декомпозиції задачі.
10. Пригадавши як здійснюється концептуалізація предметної області.
11. Пригадавши як здійснюється об’єктний аналіз предметної області.
12. Пригадавши як здійснюється визначення інтерфейсів сутностей предметної області.
13. Здійснивши аналіз задачі 12.1.
14. Здійснивши постановку задачі 12.1.
15. Програмно реалізувавши абстракцію сутності предметної області.
16. Пригадавши вимоги стандарту ISO/IEC 12207 в частині реалізації програмного елемента.
17. Здійснивши аналіз вимог до програмного модуля задачі 12.1.
18. Спроектувавши архітектуру програмного модуля.
19. Здійснивши детальне проєктування програмного модуля.
20. Розробивши тест-сьюти для проведення модульного тестування об’єкта класу.
21. Виконавши конструювання класу, об’єкт якого за наданим інтерфейсом реалізовує розв’язування задачі 12.1.
22. Пригадавши мету перевантаження функцій.
23. Пригадавши синтаксис запису перевантаження функцій.
24. Пригадали використання препроцесорних макросів.
25. Пригадали синтаксис запису функції з параметрами за замовчуванням.
26. Пригадавши призначення препроцесорних директив.
27. Пригадавши концепцію абстрактного типу даних.
28. Пригадавши синтаксис запису класу.
29. Пригадавши призначення оператора :: .
30. Пригадавши синтаксис запису оператора :: .
31. Пригадавши запис даних у глобальний об’єкт.
32. Пригадавши оголошення даних-членів класу.
33. Пригадавши синтаксис і основні правила запису функцій-членів класу.
34. Пригадавши основні правила запису функцій-членів класу.
35. Пригадавши рівень доступу private.
36. Пригадавши рівень доступу public.
37. Пригадавши синтаксис оголошення об’єкта класу.
38. Пригадавши призначення конструкторів класу.
39. Пригадавши призначення деконструкторів класу.
40. Пригадавши синтаксис запису конструкторів класу.
41. Пригадавши синтаксис запису деконструкторів класу.
42. Пригадавши процес обробки помилок.
43. Пригадавши оператори порівняння.
44. Пригадавши математичні оператори.
45. Виконати аналіз задачі Teacher.
46. Виконати постановку задачі Teacher.
47. Запропонувавши текстовий формат тест-кейса задля забезпечення коректного читання додатком Teacher текстових файлів.
48. Написавши протокол читання додатком Teacher текстових файлів.
49. Створивши текстові файли із тест-сьютами.
50. Виконавши аналіз вимог до програмного засобу Teacher.
51. Виконавши проєктування архітектури програмного засобу Teacher.
52. Виконавши детальне проєктування програмного засобу Teacher.
53. Виконавши конструювання програмного засобу Teacher.
54. Скомпілювавши проект.
55. Виконавши Unit-тестування об’єкта класу ClassLab12\_Bezkrovna
56. Пригадавши як відкривати, читати з файлу ifstream і писати у файл ofstream.
57. Використавши користувацький клас.
58. Навчившись контролювати, чи файл запущено у правильній директорії.
59. Дізналися, як отримати шлях до поточного файлу програми.
60. Навчилися перевіряти рівність дійсних чисел з допуском.
61. Практикували зчитування рядків з файлу.
62. Навчилися логічно оформлювати результати виводу у файл.
63. Згадали як викликати звуковий сигнал.
64. Згадали як підключати власний заголовковий файл.
65. Вивчили, як працювати з потоками рядків (stringstream) для парсингу складних рядків.
66. Практикували перевірку входження підрядка у рядок за допомогою find().
67. Пригадали як створювати власні функції, які повертають булеве значення (bool).
68. Згадали, як працювати з умовними конструкціями (if, else).
69. Пригадали використовувати цикли (while, for) для повторюваних дій.
70. Ознайомилися з передачею параметрів у конструктор класу.
71. Зрозуміли, як виводити помилки у консоль за допомогою cerr.
72. Побачили приклад використання математичних функцій, таких як abs().
73. Навчилися працювати з логічними значеннями true та false для контролю виконання.
74. Вивчили базові принципи тестування результатів через порівняння з еталонними значеннями.
75. Розглянули, як відокремити логіку перевірки шляху та логіку основної роботи у різні функції.
76. Ознайомилися з використанням простору імен std.
77. Навчилися структурувати код для зручності розуміння, налагодження та масштабування.
78. Зрозуміли принципи роботи з потоками введення/виведення у C++.
79. Навчилися організовувати основну логіку програми у функції main().
80. Ознайомилися з роботою з класами, створеними у зовнішніх модулях.
81. Розглянули приклад обробки текстового файлу для автоматичного тестування.
82. Навчилися створювати і використовувати тимчасові змінні для збереження результатів обчислень.
83. Навчилися формувати структурований і зручний для читання звіт про результати тестування.
84. Зрозуміли, як тестувати геометричні обчислення (об’єм конуса).
85. Дізналися, що порівняння з плаваючою точкою потрібно робити з урахуванням похибки.
86. Навчилися працювати із класом, який описує геометричну фігуру.
87. Дізналися, як застосовувати умовні оператори для контролю потоку програми.
88. Навчилися поєднувати текстові та числові дані у одному потоці.
89. Пригадали як використовувати інкремент для нумерації тестів (test\_number++).
90. Дізналися як перевірити правильність виконання лабораторної роботи через структуру проекту.
91. Дізналися як працювати з абсолютними шляхами до файлів.
92. Навчилися використовувати методи класів для інкапсуляції логіки.
93. Дізналися як працювати зі рядковими літерами у C++.
94. Зрозуміли, як коректно закривати файли після запису.
95. Реалізувавши програмний засіб згідно з завданням.
96. Створювавши коміти на протязі всієї роботи.
97. Відправлявши коміти до репозиторію.
98. Гарно організувавши свою роботу для ефективного засвоєння інформації.
99. Аналізувавши результати своєї роботи.
100. Підсумувавши виконану роботу.

**Додаток А**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | ClassLab12\_Bezkrovna |
| Назва проекта / ПЗ  Name of Project / Software | ModulesBezkrovna.h |
| Рівень тестування  Level of Testing | Модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Вероніка Безкровна |
| Виконавець  Implementer | Вероніка Безкровна |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Action (test steps) | Expected Result | Test Result (passed/failed) |
| TC-1 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 3, 4, "purple" | 50.2655 | passed |
| TC-2 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 10, 5, "yellow" | 261.799 | passed |
| TC-3 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 7, 2, "blue" | 29.3215 | passed |
| TC-4 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 0.1, 2, "red" | 0.418879 | passed |
| TC-5 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 12, 3, "white" | 113.097 | passed |
| TC-6 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 6, 0.8, "black" | 4.02124 | passed |
| TC-7 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 16, 27, "cyan" | 12214.5 | passed |
| TC-8 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 20, 10, "magenta" | 2094.4 | passed |
| TC-9 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 11, 9, "orange" | 933.053 | passed |
| TC-10 | Створити об’єкт класу – ClassLab12\_Bezkrovna з аргументами 16, 8, "green" | 1072.33 | passed |

**Додаток Б**

Лістинг ModulesBezkrovna.h:

#ifndef S\_CALCULATION

#define S\_CALCULATION

using namespace std;

#include <string>

#include <vector>

#include <iostream>

#include <cmath>

#include <stdexcept>

#define \_USE\_MATH\_DEFINES

float s\_calculation (float x, float y);

string ECTS\_rating (int rating);

string format\_float(float value);

string week\_temperature(float mon, float tue, float wed, float thu, float fri, float sat, float sun, float &week\_celsius, float &week\_fahrenheit);

int bits\_analysis(int N);

int input\_analysis(string file\_input, string file\_output);

int add\_poem\_with\_time(string file\_input);

int add\_calculation\_and\_bits(float x, float y, int b, string file\_output);

string read(string path);

vector<string> test\_results(string exp\_result\_1, string exp\_result\_2, string exp\_result\_3);

vector<string> test\_results\_10\_1(string exp\_result\_1, string exp\_result\_2, string exp\_result\_3, string exp\_result\_4, string exp\_result\_5);

vector<string> input\_10\_1(string exp\_result\_1, string exp\_result\_2, string exp\_result\_3, string exp\_result\_4);

vector<string> test\_results\_10\_2(string exp\_result\_1, string exp\_result\_2, string exp\_result\_3);

class ClassLab12\_Bezkrovna

{

private:

double height;

double radius;

string color;

public:

ClassLab12\_Bezkrovna();

ClassLab12\_Bezkrovna (double height, double radius, string color);

double GetHeight() const;

double GetRadius() const;

string GetColor() const;

void SetHeight(double input\_height);

void SetRadius(double input\_radius);

void SetColor(string input\_color);

double CountVolume();

};

ClassLab12\_Bezkrovna::ClassLab12\_Bezkrovna() : height(1.0), radius(1.0), color("yellow") {}

ClassLab12\_Bezkrovna::ClassLab12\_Bezkrovna (double input\_height, double input\_radius, string input\_color)

{

if (input\_height <= 0)

{

throw invalid\_argument("Помилка: висота повинна бути додатньою.");

}

if (input\_radius <= 0)

{

throw invalid\_argument("Помилка: радіус повинен бути додатнім.");

}

height = input\_height;

radius = input\_radius;

color = input\_color;

}

double ClassLab12\_Bezkrovna::GetHeight() const {return height;}

double ClassLab12\_Bezkrovna::GetRadius() const {return radius;}

string ClassLab12\_Bezkrovna::GetColor() const {return color;}

void ClassLab12\_Bezkrovna::SetHeight(double input\_height)

{

if (input\_height <= 0)

{

cerr << "Помилка!" << endl;

return;

}

height = input\_height;

}

void ClassLab12\_Bezkrovna::SetRadius(double input\_radius)

{

if (input\_radius <= 0)

{

cerr << "Помилка!" << endl;

return;

}

radius = input\_radius;

}

void ClassLab12\_Bezkrovna::SetColor(string input\_color)

{

color = input\_color;

}

double ClassLab12\_Bezkrovna::CountVolume()

{

return (1/3.0)\*(M\_PI\*pow(radius, 2)\*height);

}

#endif

Лістинг Teacher:

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

#include "ModulesBezkrovna.h"

using namespace std;

bool is\_in\_lab12(const string& filePath)

{

return filePath.find("\\lab12\\prj") != string::npos;

}

bool check\_main\_location()

{

string filePath = \_\_FILE\_\_;

if (!is\_in\_lab12(filePath))

{

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

{

cout << '\a';

}

ofstream output("TestResults.txt");

if (output)

{

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

output.close();

return false;

}

else

{

cerr << "Помилка, файл не може бути відкритий для письма!" << endl;

}

}

return true;

}

int main()

{

if (!check\_main\_location())

{

return 1;

}

ifstream input("C:\\Users\\Ніка\\Documents\\бмтп 2 сем\\Лабораторні роботи\\Bezkrovna-Veronika-KN24\\lab12\\TestSuite\\TestCase.txt");

ofstream output("TestResults.txt");

if (!input || !output)

{

cerr << "Помилка, файл не може бути відкритий!" << endl;

return 1;

}

string line;

const double EPSILON = 1e-1;

int test\_number = 1;

while (getline(input, line))

{

stringstream ss(line);

double height, radius, volume, result\_volume;

string color;

ss >> height >> radius >> volume >> ws;

getline(ss, color);

ClassLab12\_Bezkrovna Cone(height, radius, color);

result\_volume = Cone.CountVolume();

output << "Test " << test\_number++ << endl;

output << "Об'єм: " << result\_volume << endl;

if(abs(result\_volume - volume) < EPSILON)

{

output << "Status: passed" << endl;

}

else

{

output << "Status: failed" << endl;

}

}

return 0;

}

**Додаток В**

Вміст файлу TestResults.txt:

Test 1

Об'єм: 50.2655

Status: passed

Test 2

Об'єм: 261.799

Status: passed

Test 3

Об'єм: 29.3215

Status: passed

Test 4

Об'єм: 0.418879

Status: passed

Test 5

Об'єм: 113.097

Status: passed

Test 6

Об'єм: 4.02124

Status: passed

Test 7

Об'єм: 12214.5

Status: passed

Test 8

Об'єм: 2094.4

Status: passed

Test 9

Об'єм: 933.053

Status: passed

Test 10

Об'єм: 1072.33

Status: passed