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 \_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_ Вероніка СИТЕНКОВА

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ганна ДРЄЄВА

Кропивницький – 2024

**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.

**Завдання до лабораторної роботи:**

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 35**

****

**Аналіз та постановка задачі 12.1**

Концептуалізація:

Об’єктом є оглядове колесо. Фактично, це є коло, з даними якого будемо працювати далі.

Об’єктний аналіз:

Атрибути:

* Радіус колеса.
* Кількість кабін.

Визначення інтерфейсів сутності:

Можливість отримувати та задавати радіус оглядового колеса, кількість кабін та час оберту.

Аналіз вимог до програмного модуля ModulesSytenkova:

Назва класу - ClassLab12\_Sytenkova. Атрибути ініціалізуються конструктором. Радіус колеса, як і інші атрибути, мають бути приватними. Доступ до них здійснюється завдяки функціям-членам. Площа колеса обчислюється та повертається також функцією-членом.

Атрибутів будуть створені наступним чином:

- Радіус: (radius) (float)

- Кількість кабін: (cabin) (int)

Функції-члени класу:

- float getRadius(), void setRadius (float radius)

- int getCabin(), void setCabin (int cabin)

**Аналіз та постановка задачі 12.2**

Концептуалізація:

Потрібно реалізувати додаток Teacher, , який виконує перевірку розташування файлу проекту та проводить unit-тестування, протоколюючи результати.

Аналіз завдання:

1. Додаток повинен видати 100 звукових сигналів. Це можна реалізувати за допомогою відповідних функцій або бібліотек, які генерують звукові сигнали.
2. Необхідно перевірити, чи знаходиться файл проекту main.cpp у правильному каталозі (\Lab12\prj). Якщо файл знаходиться в іншому місці, додаток повинен записати рядок "Встановлені вимоги порядку виконання лабораторної роботи порушено!" у текстовий файл TestResults.txt.
3. Якщо файл проекту розташований правильно, додаток повинен створити об'єкт класу ClassLab12\_ Sytenkova, використовуючи заголовковий файл Modules Sytenkova.h.
4. Додаток повинен виконати unit-тестування створеного об'єкта за тест-сьютом(ами), розташованим(и) у каталозі \Lab12\TestSuite. Unit-тестування передбачає перевірку окремих частин коду на коректність роботи.
5. Результати тестування необхідно записати у текстовий файл \Lab12\TestSuite\TestResults.txt. Цей файл буде містити інформацію про те, які тести пройшли успішно, а які - ні, і допоможе оцінити коректність реалізації класу ClassLab12\_ Sytenkova.

Постановка задачі:

1. Створити функцію або модуль, відповідальний за генерацію 100 звукових сигналів.
2. Написати код для перевірки розташування файлу проекту main.cpp. Якщо файл знаходиться не в каталозі \Lab12\prj, записати відповідне повідомлення у текстовий файл TestResults.txt.
3. Реалізувати створення об'єкта класу ClassLab12\_Sytenkova на основі заголовкового файлу ModulesSytenkova.h.
4. Виконати unit-тестування створеного об'єкта, використовуючи тест-сьют(и) з каталогу \Lab12\TestSuite.
5. Протоколювати результати тестування у текстовий файл \Lab12\TestSuite\TestResults.txt, вказуючи, які тести пройшли успішно, а які - ні.

Вхідні дані:

Шлях до папки з файлом проекту main.cpp.

Вихідні дані:

Звукові сигнали (у разі невірного розташування файлу проекту).

Текстовий файл TestResults.txt з результатами перевірки та тестування.

**Лістинг ModulesSytenkova**

#ifndef MODULESSYTENKOVA\_H\_INCLUDED

#define MODULESSYTENKOVA\_H\_INCLUDED

#include <math.h>

using namespace std;

class ClassLab12\_Sytenkova{

private:

float radius;

int cabin;

public:

ClassLab12\_Sytenkova(float radius = 1.0, int cabin = 1) : radius(radius), cabin(cabin) {}

void set\_radius(float r);

void set\_cabin(int c);

float get\_radius() const { return radius; }

float get\_cabin() const { return cabin; }

float get\_area() const { return M\_PI \* pow(radius, 2); }

};

void ClassLab12\_Sytenkova::set\_radius(float r)

{

if (r >= 1)

{

radius = r;

}

else

{

cout << "Число занадто мале!" << endl;

}

}

void ClassLab12\_Sytenkova::set\_cabin(int c)

{

if (c >= 1)

{

cabin = c;

}

else

{

cout << "Число занадто мале!" << endl;

}

}

#endif // MODULESSYTENKOVA\_H\_INCLUDED

**Лістинг Teacher**

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <sstream>

#include <filesystem>

#include "ModulesSytenkova.h"

using namespace std;

int main() {

system("chcp 1251 & cls");

string currentFilePath = \_\_FILE\_\_;

int checkResult = currentFilePath.find("\\lab12\\prj\\");

if (checkResult == -1)

{

ofstream outputFile("TestResults.txt");

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

outputFile.close();

for (int i = 0; i < 100; ++i) cout << "\a";

return 0;

}

else

{

ifstream inputFile("C:\\Users\\user\\university\\Sytenkova-Veronika-KN23\\lab12\\TestSuite\\TestSuite.txt");

ofstream outputFile("C:\\Users\\user\\university\\Sytenkova-Veronika-KN23\\lab12\\TestSuite\\TestResults.txt");

if (inputFile.is\_open() && outputFile.is\_open()){

string line;

vector<float> numbers;

float num;

if (getline(inputFile, line))

{

stringstream ss1(line);

for (int i = 0; i < 2; ++i)

{

ss1 >> num;

numbers.push\_back(num);

}

}

if (getline(inputFile, line))

{

stringstream ss2(line);

for (int i = 0; i < 4; ++i)

{

ss2 >> num;

numbers.push\_back(num);

}

}

if (getline(inputFile, line))

{

stringstream ss3(line);

for (int i = 0; i < 4; ++i)

{

ss3 >> num;

numbers.push\_back(num);

}

}

inputFile.close();

ClassLab12\_Sytenkova j;

// Тест-кейс 1

outputFile << "Тест-кейс №1" << endl;

float radius = j.get\_radius();

outputFile << "Радіус колеса: "<< radius << endl;

float area = j.get\_area();

outputFile << "Площа колеса: "<< area << endl;

int cabin = j.get\_cabin();

outputFile << "Кількість кабін в оглядовому колесі: "<< cabin << endl;

if (abs(area - numbers[0]) < 0.001 && cabin == numbers[1])

{

outputFile << "Статус тест-кейса: passed\n" << endl;

}

else

{

outputFile << "Статус тест-кейса: failed\n" << endl;

}

// Тест-кейс 2

outputFile << "Тест-кейс №2" << endl;

j.set\_radius(numbers[2]);

j.set\_cabin(numbers[3]);

radius = j.get\_radius();

outputFile << "Радіус колеса: "<< radius << endl;

area = j.get\_area();

outputFile << "Площа колеса: "<< area << endl;

cabin = j.get\_cabin();

outputFile << "Кількість кабін в оглядовому колесі: "<< cabin << endl;

if (abs(area - numbers[4]) < 0.001 && cabin == numbers[5])

{

outputFile << "Статус тест-кейса: passed\n" << endl;

}

else

{

outputFile << "Статус тест-кейса: failed\n" << endl;

}

// Тест-кейс 3

outputFile << "Тест-кейс №3" << endl;

j.set\_radius(numbers[6]);

j.set\_cabin(numbers[7]);

radius = j.get\_radius();

outputFile << "Радіус колеса: "<< radius << endl;

area = j.get\_area();

outputFile << "Площа колеса: "<< area << endl;

cabin = j.get\_cabin();

outputFile << "Кількість кабін в оглядовому колесі: "<< cabin << endl;

if (abs(area - numbers[8]) < 0.001 && cabin == numbers[9])

{

outputFile << "Статус тест-кейса: passed\n" << endl;

}

else

{

outputFile << "Статус тест-кейса: failed\n" << endl;

}

outputFile.close();

}

}

return 0;

}

**Вміст файлу TestResults.txt**

Тест-кейс №1

Радіус колеса: 1

Площа колеса: 3.14159

Кількість кабін в оглядовому колесі: 1

Статус тест-кейса: passed

Тест-кейс №2

Радіус колеса: 5

Площа колеса: 78.5398

Кількість кабін в оглядовому колесі: 3

Статус тест-кейса: passed

Тест-кейс №3

Радіус колеса: 10

Площа колеса: 314.159

Кількість кабін в оглядовому колесі: 8

Статус тест-кейса: passed

Висновок:

Лабораторна робота на тему "Програмна реалізація абстрактних типів даних" спрямована на розвиток фундаментальних навичок об'єктно-орієнтованого аналізу і проектування з використанням мови програмування С++. В процесі виконання завдань я набувають досвіду створення класів, тестування їх екземплярів, а також застосування препроцесорних директив, макросів і макрооператорів у кросплатформовому середовищі Code::Blocks.

В завданні 12.1 потрібно створити клас. Завдяки поставленому завданню, на практиці можна освоїти такі речі як: рівень доступу в класі та методи. Оскільки атрибути повинні мати рівень доступу приватний, з ними потрібно взаємодіяти за рахунок функцій-членів. Завдяки інформації даній на лекціях та методичним вказівками, де детально написані шаги виконання лабораторної роботи, труднощів не виникло. Особисто мені був корисний наступний пункт:

* реалізацію функцій-членів класу, тіло яких складається з понад одного рядка, необхідно здійснювати поза тілом класу; для ідентифікації функції як члена класу потрібно її ім’я кваліфікувати ідентифікатором класу за допомогою оператора розширення області видимості; наприклад, bool MyClass::out(char \*name)

Спочатку не було розуміння того, як краще реалізувати методи, але цей уточнення було корисним.

Під час виконання завдання 12.2 було більше труднощів. Однією з складних частин є тестування. Зупинила вибір на звичайному текстовому файлі, в якому вписані потрібні числа, а саме тестування відбувається автоматично. Я не зовсім задоволена результатом, оскільки можна зробити набагато більш оптимізовано та, можливо, легше, але це був найоптимальніший варіант.

Також були труднощі з передаванням файлу. Нажаль, єдиний варіант, який в мене працює - це передавання повного шляху.