Лабораторна робота №3

на тему: „Поліморфізм C++“

**Виконав:**

студент II курсу, групи ІП-71

Мокроусов В. Р.

**Перевірив:**

асистент Пономаренко М. Н.

**ЗМІСТ**

1. Мета роботи…………………………………………………………………..3
2. Постановка задачі…………………………………………………………….4
3. Алгоритм вирішення задачі………………………………………………….5
4. UML діаграма………………………………………………………………....6
5. Код програми………………………………………………………………….7
6. Приклад роботи програми…………………………………………………..11
7. Висновок……………………………………………………………………..12

**МЕТА РОБОТИ**

Мета роботи ― вивчити основні концепції об'єктно-орієнтованого програмування. Вивчити особливості віртуальних функцій, абстрактних класів і поліморфізму.

**ПОСТАНОВКА ЗАДАЧІ**

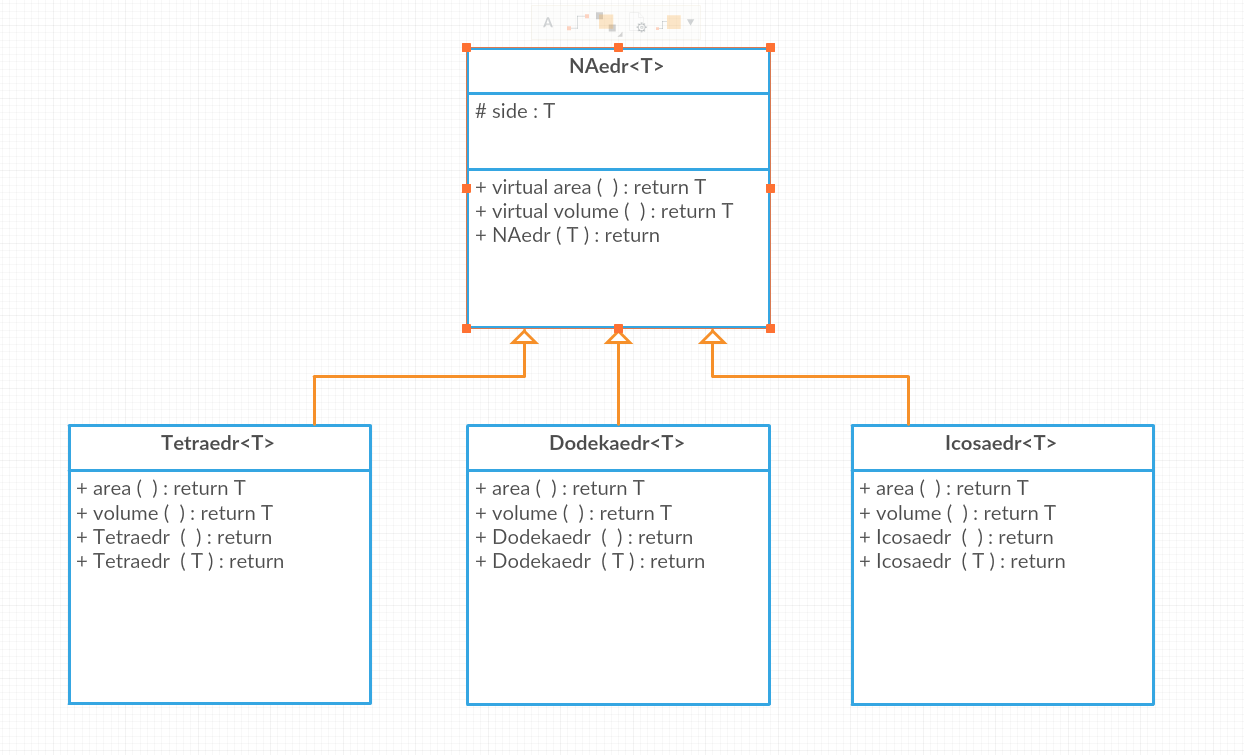
Задачею лабораторної роботи є реалізація класів *багатогранник* та його спадкоємців: *тетраедрб додекаедр, ікосаедр.* У базовому класі необхідно визначити методи обчислення повної площі поверхні та об'єму фігури*.* Ці методи мають бути перевизначені у спадкоємців. У базовому класі необхідно оголосити довжину сторони *side*. Продемонструвати роботу методів.

**АЛГОРИТМ ВИРІШЕННЯ ЗАДАЧІ**

Для вирішення задачі було імплементовано класи *Naedr* (багатогранник) та спадкоємців *Tetraedr, Dodekaedr, Icosaedr*. Базовий клас містить елемент дані : *side* та методи *area(), volume().* Спадкоємці наслідують від базового класу та перевизначають дані методи. Для кожного з них також визначається конструктор, що викликає конструктор абстрактоного класу *NAedr*.

**UML ДІАГРАМА**

Нижче наведено UML діаграму класів *NAedr* та *Tetraedr, Dodekaedr, Icosaedr*.



**КОД ПРОГРАМИ**

// main.cpp

#include "NAedrTemplate.hpp"

#include <iostream>

using namespace std;

int main(int argc, char\* argv[]) {

int side;

cout << "Insert side : "; cin >> side;

NAedr<double>\* tetraedr = new Tetraedr<double>(side);

cout << "Tetrader area : " << tetraedr->area() << endl;

cout << "Tetrader volume : " << tetraedr->volume() << endl;

NAedr<double>\* dodekaedr = new Dodekaedr<double>(side);

cout << "Dodekaedr area : " << dodekaedr->area() << endl;

cout << "Dodekaedr volume: " << dodekaedr->volume() << endl;

NAedr<double>\* icosaedr = new Icosaedr<double>(side);

cout << "Icosaedr area : " << icosaedr->area() << endl;

cout << "Icosaedr volume : " << icosaedr->volume() << endl;

delete tetraedr;

delete dodekaedr;

delete icosaedr;

}

//NaedrTemplate.hpp

#ifndef NAEDR\_HPP

#define NAEDR\_HPP

#include <stdlib.h>

#include <cmath>

#include <utility>

#define DEFAULT 10

// NAedr accepts only numeric types [int, float, short, double]

template<class T>

class NAedr {

protected:

T side;

public:

virtual T area() = 0;

virtual T volume() = 0;

NAedr<T>(T key) : side(std::abs(key)) { }

//NAedr<T>(const T& key) : side(std::abs(key)) { }

//NAedr<T>(T&& key) : side(std::move(std::abs(key))) { } // Bullshit for numeric types(Overloaded Longint?)

};

template<class T>

class Tetraedr : public NAedr<T> {

private:

using NAedr<T>::side;

public:

Tetraedr() : NAedr<T>(DEFAULT) { }

Tetraedr(T a) : NAedr<T>(a) { }

T area() override { return std::sqrt(3)\*side\*side; }

T volume() override { return std::pow(side, 3)/(6\*std::sqrt(2)); }

};

template<class T>

class Dodekaedr : public NAedr<T> {

private:

using NAedr<T>::side;

public:

Dodekaedr() : NAedr<T>(DEFAULT) { }

Dodekaedr(T a) : NAedr<T>(a) { }

T area() override { return 3\*std::pow(side, 2)\*std::sqrt(25+10\*std::sqrt(5)); }

T volume() override { return 0.25\*(15+7\*std::sqrt(5))\*pow(side, 3); }

};

template<class T>

class Icosaedr : public NAedr<T> {

private:

using NAedr<T>::side;

public:

Icosaedr() : NAedr<T>(DEFAULT) { }

Icosaedr(T a) : NAedr<T>(a) { }

T area() override { return pow(side, 2) \* 5 \* std::sqrt(3); }

T volume() override { return 5\*(3+std::sqrt(5))\*pow(side, 3)/12; }

};

#endif

// Makefile

CPPFLAGS=-Wpedantic -std=c++14 -g

LDFLAGS=

OFNAME=main

SRC=main.cpp

build:

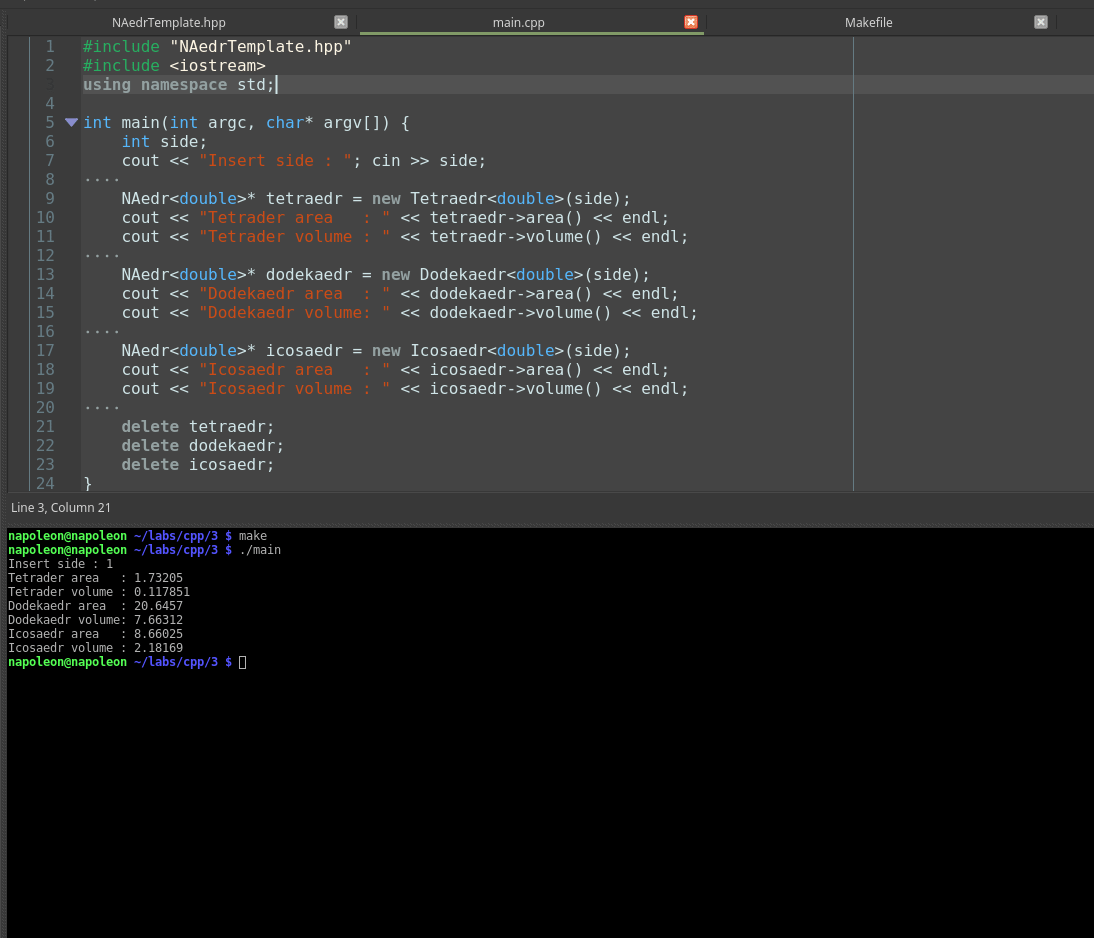
@g++ ${CPPFLAGS} ${SRC} -o ${OFNAME} ${LDFLAGS}

run: build

@./${OFNAME}

**ПРИКЛАД РОБОТИ ПРОГРАМИ**

Программа компілюється командою *make* (за допомогою *Makefile*), та запускається в консолі *./main*.



**ВИСНОВОК**

При виконанні лабораторної роботи було, згідно умови, створено відповідні класи. Для них було реалізовано усі визначені умовою методи. При виконанні було використано механізм шаблонів. Було використано елементи стандартної бібліотеки C++, зокрема функції для виводу результатута інші.

В результаті було створено ефективну програму для розв'язку поставленої задачі, згідно з заданими умов.