Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №4

з дисципліни: “Програмування, ч.2 (ООП)”

на тему: “ Класи та об’єкти”

Варіант 3

Виконав: ст.гр. КІ-15

Блищак Р.С.

Прийняв: Козак Н.Б.

Львів 2020

Мета: познайомитися із класами та об‘єктами.

Теоретичні відомості :

Основна відмінність будь-якої об‘єктно-орієнтованої мови програмування від інших не об‘єктно-орієнтованих мов програмування є можливість створення нових визначених користувачем типів, що називаються класами. Клас – це визначений користувачем тип з даними-елементами (властивостями) та функціями-елементами (методами), що являються членами класу. Він зазвичай описує певну абстракцію якоїсь сутності реального світу з її властивостями і можливими діями. Змінна типу клас називається об‘єктом. Об‘єкт – це вже не абстракція, а елемент реального світу, який може володіти певними характеристиками, які задаються властивостями в класі, та діяти згідно операцій заданих методами у класі. Оголошення класу в мові С++ має наступний синтаксис (не обов‘язково щоб клас мав всі секції чи спадкував базовий клас)

Оголошення класу містить оголошення даних-елементів (властивостей) та функцій-елементів (методів) класу. Одна з фундаментальних основ ООП передбачає інкапсуляцію даних, тобто дані мають бути недоступними ззовні, а 49 лише за посередництвом загальнодоступних методів класу. Оголошення методів має вигляд оголошення прототипу функції в середині однієї з секцій класу, зазвичай у секції public. Якщо методи є службовими і не мають бути доступні ззовні, тоді їх розміщують в секції private або protected. Сама ж реалізація методу може знаходитись як усередині класу (тоді оголошення методу в класі має вигляд оголошення функції з тілом), так і поза оголошенням класу (тоді у класі оголошується прототип функції, а її тіло визначається за межами класу). Але один з найфундаментальніших принципів розробки програмного забезпечення мовою С++ полягає у розмежуванні інтерфейсу класу від його реалізації. Тому при побудові програми мовою C++ кожне оголошення класу зазвичай розміщують у заголовочний файл \*.h назва якого співпадає з назвою класу, а реалізацію методів цього класу – у файл \*.cpp із тим ж іменем, що й \*.h файл. Заголовочні файли включаються (за допомогою директиви #іnclude) у кожен файл, у якому використовується клас, а файли з вихідними кодами компілюються і компонуються з файлом, що містить головну програму (main-функцію).

Завдання (Варіант 3) -

Клас CQuadrangle (Чотирикутник). Клас зберігає декартові координати чотирьох кутів чотирикутника. Конструктор приймає чотири групи координат. Повинні бути передбачені методи, що обчислюють периметр і площу, а також метод, що перевіряє чи передані координати визначають прямокутник.

Код для вирішення задачі даного варіанту -

#include <iostream>

#include <String>

#include <cmath>

using namespace std;

void clear() {

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

}

void empty() {

cout << endl;

}

class MyVector {

public:

void xyz(int x, int y, int z) {

coord[0] = x;

coord[1] = y;

coord[2] = z;

}

void x(int ix) {

coord[0] = ix;

}

void y(int iy) {

coord[1] = iy;

}

void z(int iz) {

coord[2] = iz;

}

float rx() {

return coord[0];

}

float ry() {

return coord[1];

}

float rz() {

return coord[2];

}

float coord[3] = { 0,0,0 };

MyVector(int x, int y, int z) {

coord[0] = x;

coord[1] = y;

coord[2] = z;

}

MyVector() {

coord[0] = 0;

coord[1] = 0;

coord[2] = 0;

}

MyVector\* operator =(MyVector v1){

this->coord[0] = v1.coord[0];

this->coord[1] = v1.coord[1];

this->coord[2] = v1.coord[2];

return this;

}

};

class CQuad {

public:

CQuad(MyVector v1, MyVector v2, MyVector v3, MyVector v4) {

vector1 = v1;

vector2 = v2;

vector3 = v3;

vector4 = v4;

a = sqrt(pow(v1.rx() - v2.rx(), 2) + pow(v1.ry() - v2.ry(), 2) + pow(v1.rz() - v2.rz(), 2));

b = sqrt(pow(v2.rx() - v3.rx(), 2) + pow(v2.ry() - v3.ry(), 2) + pow(v2.rz() - v3.rz(), 2));

c = sqrt(pow(v3.rx() - v4.rx(), 2) + pow(v3.ry() - v4.ry(), 2) + pow(v3.rz() - v4.rz(), 2));

d = sqrt(pow(v4.rx() - v1.rx(), 2) + pow(v4.ry() - v1.ry(), 2) + pow(v4.rz() - v1.rz(), 2));

vA.xyz(v2.rx() - v1.rx(), v2.ry() - v1.ry(), v2.rz() - v1.rz());

vB.xyz(v2.rx() - v3.rx(), v2.ry() - v3.ry(), v2.rz() - v3.rz());

vC.xyz(v4.rx() - v3.rx(), v4.ry() - v3.ry(), v4.rz() - v3.rz());

vD.xyz(v4.rx() - v1.rx(), v4.ry() - v1.ry(), v4.rz() - v1.rz());

s1 = (vA.rx() \* vD.rx() + vA.ry() \* vD.ry() + vA.rz() \* vD.rz());

s2 = (vB.rx() \* vC.rx() + vB.ry() \* vC.ry() + vB.rz() \* vC.rz());

al = (s1) / (a \* d);

bt = (s2) / (b \* c);

sinal = sqrt(1 - pow(al, 2));

sinbt = sqrt(1 - pow(bt, 2));

p = (a + b + c + d) / 2;

t = (1 + (al \* bt - sinal \* sinbt)) / 2;

sized = sqrt((p - a) \* (p - b) \* (p - c) \* (p - d) - a \* b \* c \* d \* t);

}

float size() {

return sized;

}

bool square() {

if (al == 0 && bt == 0) {

return true;

}

else return false;

}

float perimetr() {

return a + b + c + d;

}

private:

MyVector vector1, vector2, vector3, vector4, vA, vB, vC, vD;

float a, b, c, d, d1, d2, p, t, al, bt, s1, s2, sinal, sinbt, sized = 0;

};

int main() {

MyVector vA, vB, vC, vD;

int x, y, z = 0;

cout << "Enter a x parameter to A point - " << endl;

cin >> x;

cout << "Enter a y parameter to A point - " << endl;

cin >> y;

cout << "Enter a z parameter to A point - " << endl;

cin >> z;

vA.xyz(x,y,z);

clear();

x, y, z = 0;

cout << "Enter a x parameter to B point - " << endl;

cin >> x;

cout << "Enter a y parameter to B point - " << endl;

cin >> y;

cout << "Enter a z parameter to B point - " << endl;

cin >> z;

vB.xyz(x, y, z);

clear();

x, y, z = 0;

cout << "Enter a x parameter to C point - " << endl;

cin >> x;

cout << "Enter a y parameter to C point - " << endl;

cin >> y;

cout << "Enter a z parameter to C point - " << endl;

cin >> z;

vC.xyz(x, y, z);

clear();

x, y, z = 0;

cout << "Enter a x parameter to D point - " << endl;

cin >> x;

cout << "Enter a y parameter to D point - " << endl;

cin >> y;

cout << "Enter a z parameter to D point - " << endl;

cin >> z;

vD.xyz(x, y, z);

clear();

CQuad myQuad(vA,vB,vC,vD);

cout << "Okay, all has been created and completed";

if (myQuad.square()) {

cout << ", ur Quad is square-like" << endl;

}

else cout << ", ur Quad is not a square-like" << endl;

cout << "So, it takes place of " << myQuad.size() << ";" << endl;

cout << "And its perimeter is " << myQuad.perimetr() << ";" << endl;

}

Скріншот виконання програми –



Висновок : в даній лабораторній роботі я опрацював роботу з классами, об’єктами классів, застосуванням їх і використанням геттерів і сеттерів.