**мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»**

**ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра ІСМ**

|  |
| --- |
| **2022** |



**ЗВІТ**

**Про виконання лабораторної роботи № 3.3G**

**«Успадкування замість композиції»**

**З дисципліни**

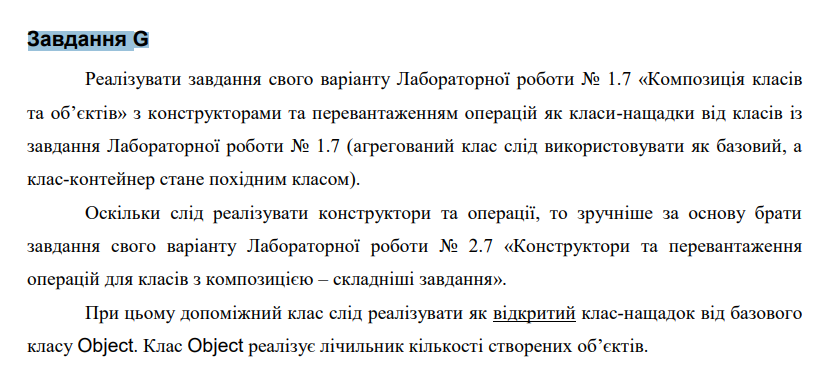
**«Об’єктно-орієнтоване програмування»**

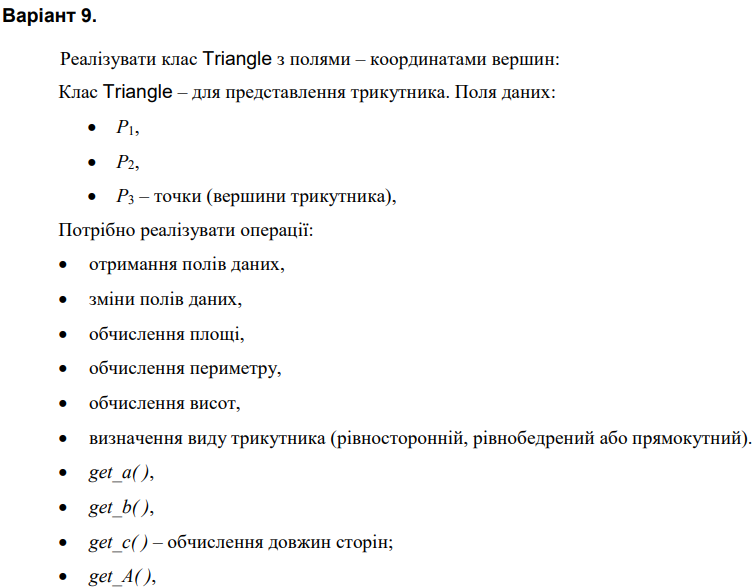
**Студента групи ІТ-11**

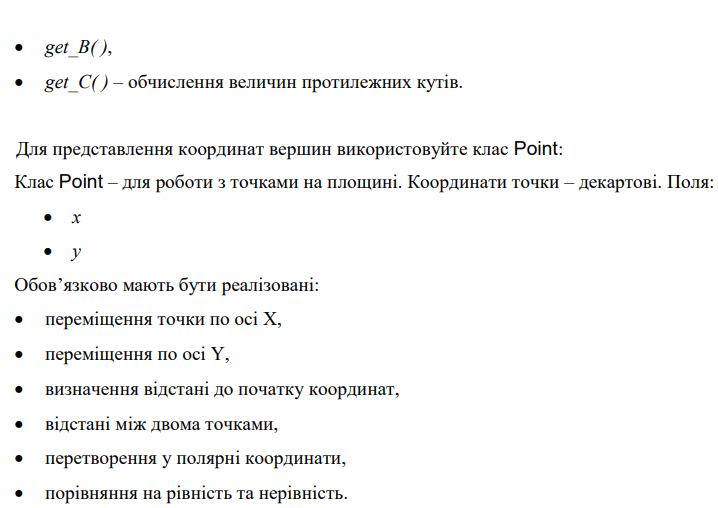
**Лисецького Володимира Любомировича**

**Мета роботи:** Освоїти використання успадкування.

**Умова завдання:**

****

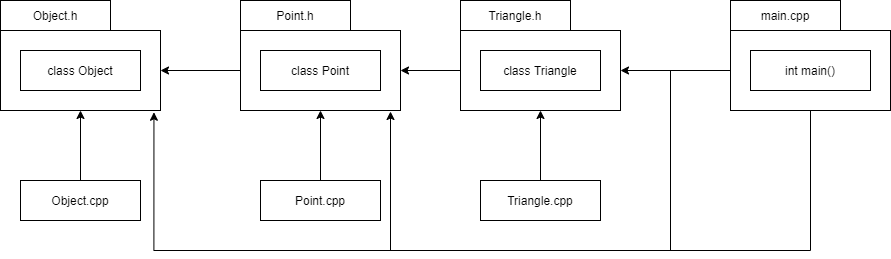
****

****

**UML-діаграма класів:**

**C:\Users\lyset\AppData\Local\Microsoft\Windows\INetCache\Content.Word\UML.png**

**Структурна схема програми:**



**Текст програми:**

#include "Point.h"

#include <iostream>

#include <string>

class Triangle :

public Point

{

private:

Point P1, P2, P3; // A B C

public:

Point getP1() { return P1; }

Point getP2() { return P2; }

Point getP3() { return P3; }

void setP1(double x, double y) { P1.setX(x); P1.setY(y); }

void setP2(double x, double y) { P2.setX(x); P2.setY(y); }

void setP3(double x, double y) { P3.setX(x); P3.setY(y); }

Triangle();

Triangle(double x1, double y1, double x2, double y2, double x3, double y3);

Triangle(const Triangle& obj);

Triangle& operator = (const Triangle& obj);

operator string() const;

double getObjX(Point obj) const { return obj.getX(); }

double getObjY(Point obj) const { return obj.getY(); }

void setObjX(Point obj, double value) { obj.setX(value); }

void setObjY(Point obj, double value) { obj.setY(value); }

double get\_a()const { return distanceOfPoints(P1, P2); }; // Side AB P1-P2 a

double get\_b() const { return distanceOfPoints(P2, P3); }; // Side BC P2-P3 b

double get\_c() const { return distanceOfPoints(P1, P3); }; // Side AC P1-P3 c

double get\_A() const; // Angle A

double get\_B()const; // Angle B

double get\_C()const; // Angle C

double hA() const; // Height AH

double hB() const; // Height BH

double hC()const; // Height CH

friend istream& operator >> (istream& in, Triangle& obj);

friend ostream& operator << (ostream& in, const Triangle& obj);

Triangle& operator ++();

Triangle& operator --();

Triangle operator ++(int);

Triangle operator --(int);

double perimeter()const;

string triangleType()const;

double square() const;

};

///////////////////////////////////////////////////////////////

// Point.h

#pragma once

#include "Object.h"

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

class Point: public Object

{

private:

double x, y;

public:

double getX() { return x; }

double getY() { return y; }

void setX(double value) { x = value; }

void setY(double value) { y = value; }

Point();

Point(double a, double b);

Point(const Point& obj);

Point& operator = (const Point& obj);

operator string() const;

double distanceToOrigin();

void moveX(double value) { x += value; }

void moveY(double value) { y += value; }

void polarizeCoords();

friend istream& operator >> (istream& in, Point& obj);

friend ostream& operator << (ostream& in, const Point& obj);

Point& operator ++();

Point& operator --();

Point operator ++(int);

Point operator --(int);

friend double distanceOfPoints(Point p1, Point p2);

friend bool operator ==(Point p1, Point p2);

friend bool operator != (Point p1, Point p2);

};

#pragma once

class Object

{

protected:

static unsigned int count;

public:

Object();

~Object();

unsigned int get\_count() { return count; };

};

///////////////////////////////////////////////////////////////

// Lab3\_3F.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

#include <iostream>

#include "Triangle.h"

#include <string>

using namespace std;

int main()

{

cout << "Open inheritance" << endl;

Triangle ABC;

cin >> ABC;

cout << ABC;

cout << "AB(P1P2) = " << ABC.get\_a() << endl;

cout << "BC(P2P3) = " << ABC.get\_b() << endl;

cout << "AC(P1P3) = " << ABC.get\_c() << endl;

cout << endl;

cout << "Type of ABC = " << ABC.triangleType() << endl;

cout << "Perimeter ABC = " << ABC.perimeter() << endl;

cout << "Square ABC = " << ABC.square() << endl;

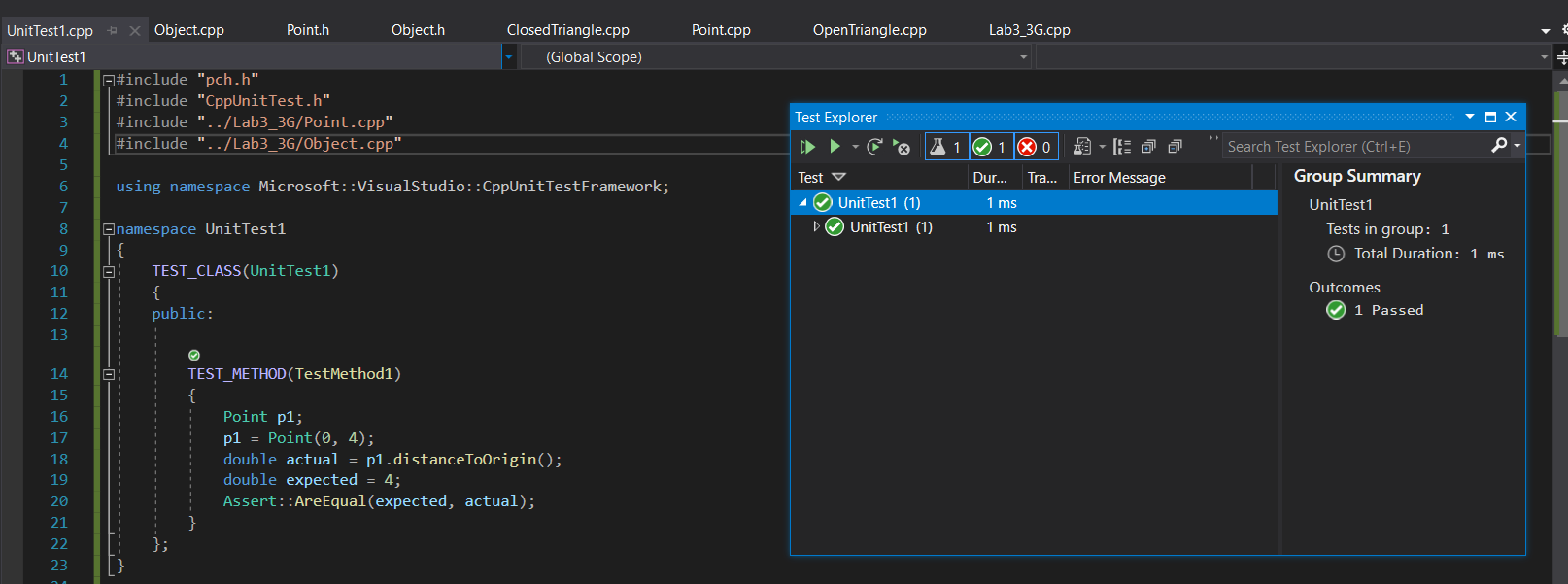
cout << "total objects = " << ABC.get\_count() << endl;

}

**Посилання на git-репозиторій з проектом:**

**https://github.com/MarshmallowSoup/oop-lab-3-3G**

**Результати unit-тесту:**

****

**Висновки:6.05 o 5.37 закінчив писати звіт.**