**мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»**

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

**Кафедра ІСМ**

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



**ЗВІТ**

**Про виконання лабораторної роботи № 3.3A**

**«Успадкування замість композиції»**

**З дисципліни**

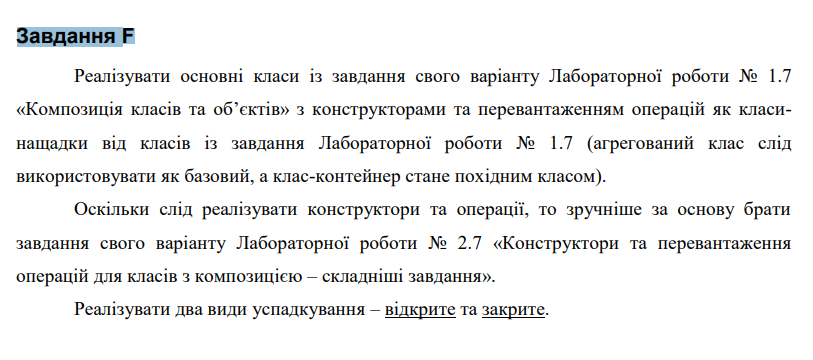
**«Об’єктно-орієнтоване програмування»**

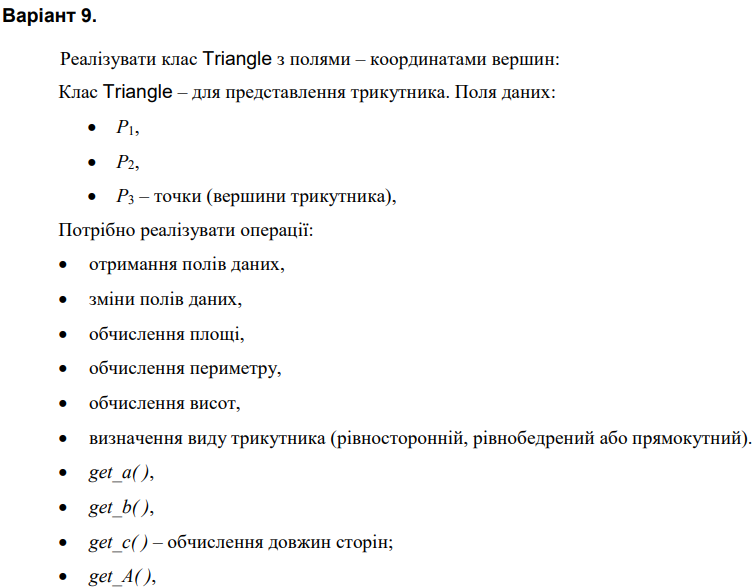
**Студента групи ІТ-11**

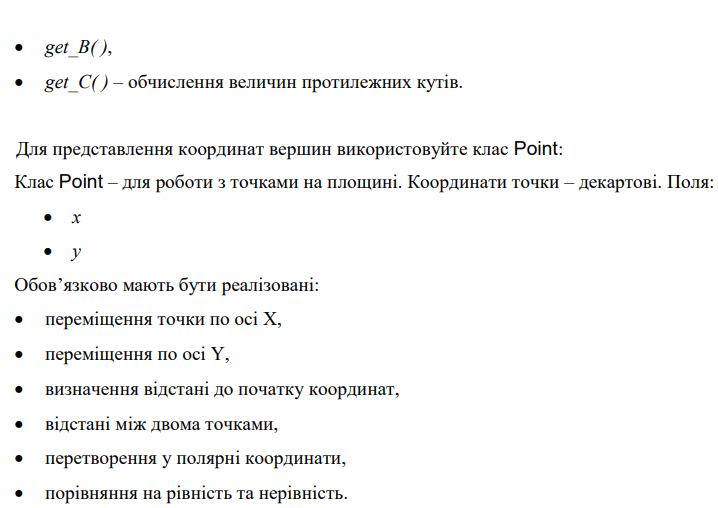
**Лисецького Володимира Любомировича**

**Мета роботи:** Освоїти використання успадкування.

**Умова завдання:**

****

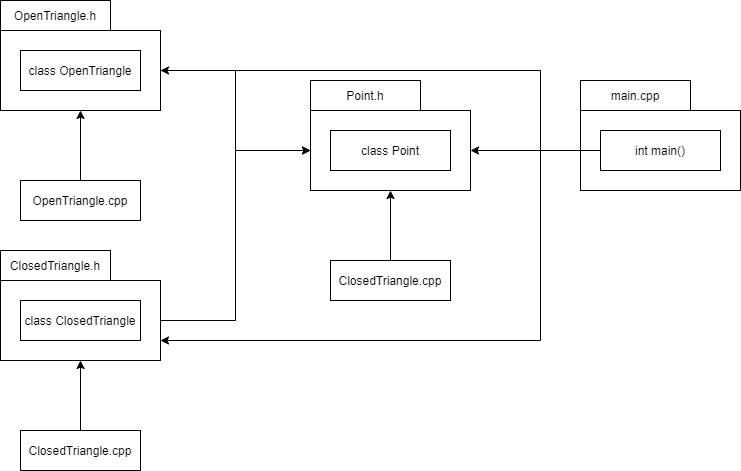
****

****

**UML-діаграма класів:**



**Структурна схема програми:**



**Текст програми:**

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

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

//

#include <iostream>

#include "OpenTriangle.h"

#include "ClosedTriangle.h"

#include <string>

using namespace std;

int main()

{

cout << "Open inheritance" << endl;

OpenTriangle 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 << "Closed inheritance" << endl;

ClosedTriangle XYZ;

cin >> XYZ;

cout << XYZ;

cout << "XY(P1P2) = " << XYZ.get\_a() << endl;

cout << "YZ(P2P3) = " << XYZ.get\_b() << endl;

cout << "XZ(P1P3) = " << XYZ.get\_c() << endl;

cout << endl;

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

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

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

}

#include "Point.h"

#include <iostream>

#include <string>

class OpenTriangle :

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); }

OpenTriangle();

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

OpenTriangle(const OpenTriangle& obj);

OpenTriangle& operator = (const OpenTriangle& 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, OpenTriangle& obj);

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

OpenTriangle& operator ++();

OpenTriangle& operator --();

OpenTriangle operator ++(int);

OpenTriangle operator --(int);

double perimeter()const;

string triangleType()const;

double square() const;

};

#pragma once

#include "Point.h"

class ClosedTriangle :

private 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); }

ClosedTriangle();

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

ClosedTriangle(const ClosedTriangle& obj);

ClosedTriangle& operator = (const ClosedTriangle& 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, ClosedTriangle& obj);

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

ClosedTriangle& operator ++();

ClosedTriangle& operator --();

ClosedTriangle operator ++(int);

ClosedTriangle operator --(int);

double perimeter()const;

string triangleType()const;

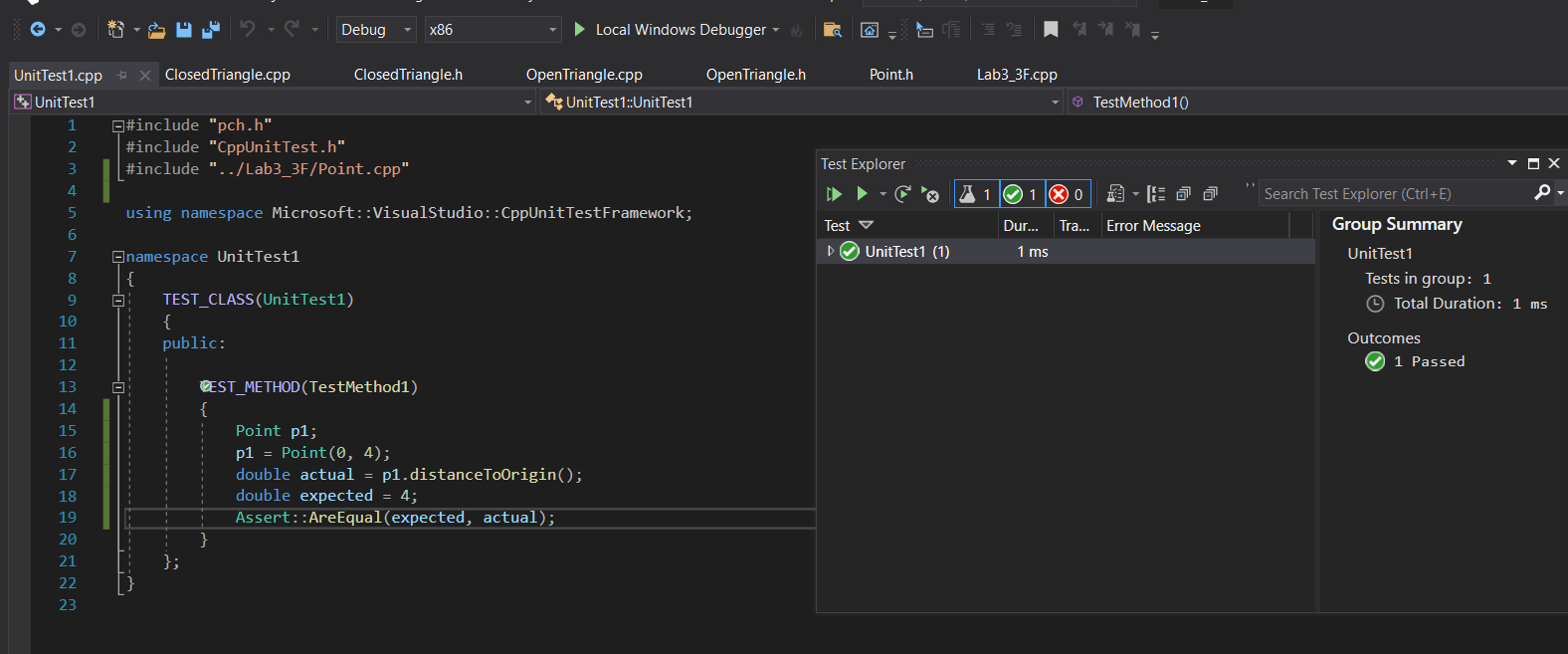
double square() const;

};

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

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

**Результати unit-тесту:**

****

**Висновки:6.05 5.20 закінчив писати звіт.**