# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №2**

# з курсу

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

*Студента 2 курсу*

*групи ПП-21 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

к.ф.-м.н., доц. Шолохов О.В.

## Київ – 2022

**1.Назва роботи**

Дослідження внутрішнього устрою класів

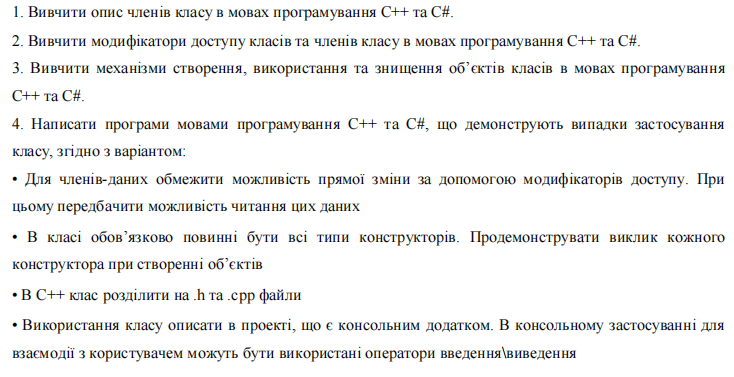
1. **Тема роботи**

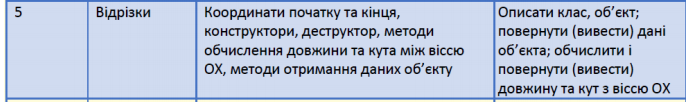
Дослідження внутрішнього устрою класів

1. **Мета роботи**

Вивчити опис членів класу в мовах програмування С++ та С#, вивчити модифікатори доступу класів та членів класу в мовах програмування С++ та С#, вивчити механізми створення, використання та знищення об’єктів класів в мовах програмування С++ та С#, написати програми мовами програмування С++ та С#, що демонструють випадки застосування класу згідно з варіантом.

1. **Умова завдання**





1. **Рішення**

Об’являємо класс Відрізок. Оголошуємо змінні початку та кінця. Описуємо метод, що обчислює довжину відрізка, приймаючи дві точки (початок та кінець), використовуючи формулу AB = √(xb - xa)2 + (yb - ya)2, де А - початок, В - кінець. Описуємо метод, що обчислює кут нахилу відрізка до осі ОХ шляхом знаходження відношення протилеглого катета до гіпотенузи (якщо вірізок не паралельний чи перпендикулярний осі ОХ). Описуємо метод, що виводить всі дані (точки, довжину, кут нахилу) на екран консолі, після чого описуємо всі конструктори (стандартний за замовчуванням, конструктор з параметрами, конструктор копіювання (той, що копіює дані з іншого класу), статичний конструктор, приватний конструктор та деструктор). У функції main створюємо екземпляр классу, скориставшись параметризованим конструктором, після чого викликаємо метод printRedacted().

Код (C#):

using System;

using System.Drawing;

namespace At\_210

{

class Program

{

public class Segment

{

public readonly Point start;

public readonly Point end;

public double lengthMeasurer (Point start, Point end)

{

double length = Math.Sqrt(Math.Pow((end.X - start.X), 2) + Math.Pow((end.Y - start.Y), 2));

return length;

}

public double angleMeasurer(Point start, Point end)

{

if (start.X == end.X)

{

return 90;

} else if (start.Y == end.Y)

{

return 0;

} else

{

double hypotenuse = lengthMeasurer(start, end);

Point ninetyDegreeAngle = new Point();

ninetyDegreeAngle.X = start.X;

ninetyDegreeAngle.Y = end.Y;

double leg = lengthMeasurer(start, ninetyDegreeAngle);

return Math.Tan(leg/hypotenuse);

}

}

public void printRedacted()

{

Console.WriteLine("Starting point: " + start);

Console.WriteLine("Ending point: " + end);

Console.WriteLine("Length: " + lengthMeasurer(start, end));

Console.WriteLine("Angle to OX: " + angleMeasurer(start, end));

}

public Segment()

{

start.X = 1;

start.Y = 4;

end.X = 8;

end.Y = 8;

}

public Segment(int x1, int y1, int x2, int y2)

{

start.X = x1;

start.Y = y1;

end.X = x2;

end.Y = y2;

}

public Segment(Segment anotherSegment)

{

start.X = anotherSegment.start.X;

start.Y = anotherSegment.start.Y;

end.X = anotherSegment.end.X;

end.Y = anotherSegment.end.Y;

}

static Segment() { }

//private Segment() { }

~Segment() { }

}

static void Main(string[] args)

{

Console.WriteLine("Please enter the starting point's X, then Y coordinates: ");

int x1 = Convert.ToInt32(Console.ReadLine());

int y1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter the ending point's X, then Y coordinates: ");

int x2 = Convert.ToInt32(Console.ReadLine());

int y2 = Convert.ToInt32(Console.ReadLine());

Segment instanceOfASegment = new Segment(x1, y1, x2, y2);

instanceOfASegment.printRedacted();

}

}

}

Код(С++):

#define \_USE\_MATH\_DEFINES

#define M\_PI 3.14159265358979323846

class Segment {

struct Point

{

int X;

int Y;

};

public:

Point start;

Point end;

double lengthMeasurer(Point start, Point end) {

double length = sqrt(pow((end.X - start.X), 2) + pow((end.Y - start.Y), 2));

return length;

}

double angleMeasurer(Point start, Point end)

{

if (start.X == end.X)

{

return 90;

}

else if (start.Y == end.Y)

{

return 0;

}

else

{

double hypotenuse = lengthMeasurer(start, end);

Point ninetyDegreeAngle = Point();

ninetyDegreeAngle.X = start.X;

ninetyDegreeAngle.Y = end.Y;

double leg = lengthMeasurer(start, ninetyDegreeAngle);

return tan(leg / hypotenuse);

}

}

void printRedacted()

{

double length = lengthMeasurer(start, end);

double angle = angleMeasurer(start, end);

std::cout << "Starting point's X: " << start.X << "starting point's Y: " << start.Y;

std::cout << "\nEnding point's X: " << end.X << "ending point's Y: " << end.Y;

std::cout << "\nLength: " << length;

std::cout << "\nAngle to OX (in degrees): " << ((180 \* angle) / M\_PI);

}

Segment() {

start.X = 1;

start.Y = 4;

end.X = 8;

end.Y = 8;

}

Segment(int x1, int y1, int x2, int y2) {

start.X = x1;

start.Y = y1;

end.X = x2;

end.Y = y2;

}

Segment(const Segment& p1) {

start = p1.start;

end = p1.end;

}

};

int main()

{

std::cout << "Please enter the starting point's X, then Y coordinates: ";

int x1;

int y1;

std::cin >> x1;

std::cin >> y1;

std::cout << "Please enter the ending point's X, then Y coordinates: ";

int x2;

int y2;

std::cin >> x2;

std::cin >> y2;

Segment instanceOfASegment = Segment(x1, y1, x2, y2);

instanceOfASegment.printRedacted();

}

Результат роботи програми:

[screenshot]

1. **Висновки**

В результаті виконання даної лабораторної роботи я оволодів практичними навичками створення алгоритмів для виконання типових завдань на створення класу з заданими полями та методами, пошуку довжини відрізка та його куту нахилу до осі ОХ з використанням кординат початку та кінця вищезгаданого відрізка. Вважаю дану лабораторну роботу виконаною в повному обсязі.