# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №3**

# 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**

Перевантаження операторів

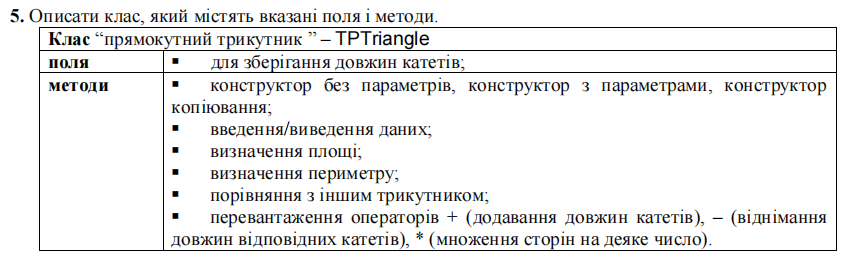
1. **Тема роботи**

Перевантаження операторів

1. **Мета роботи**

Набуття навичок розробки класів з використанням перевантажених операторів.

1. **Умова завдання**



1. **Рішення**

Об’являємо класс Трикутник. Оголошуємо змінні двох катетів. Описуємо метод, що дозволяє ввести довжини катетів, виводячи також запрошення. Описуємо метод, що обчислює площу трикутника за формулою двох катетів ((a \* b/2), де a і b - катети). Описуємо метод, що обчислює периметр після обчислення гіпотенузи за теоремою Піфагора. Описуємо метод, що порівнює два трикутники. Описуємо метод, що виводить усі дані про трикутник. Перевантажуємо унарні оператори “+” та “-”, бінарний оператор “\*” згідно з вищевказаною умовою завдання, після чого описуємо всі конструктори (стандартний за замовчуванням, конструктор з параметрами, конструктор копіювання (той, що копіює дані з іншого класу). У функції main створюємо екземпляр классу, скориставшись конструктором без параметрів, після чого викликаємо методи классу.

Код:

using System;

using System.Text;

namespace Fr\_223

{

class Program

{

public class Triangle

{

double legOne;

double legTwo;

public void basicInputSystem()

{

Console.WriteLine("Please enter the length of the first leg below:");

legOne = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Please enter the length of the second leg below:");

legTwo = Convert.ToDouble(Console.ReadLine());

}

public double areaCalculator()

{

double area = (legOne \* legTwo) / 2;

return area;

}

public double perimeterCalculator()

{

double perimeter = Math.Sqrt((Math.Pow(legOne, 2) + Math.Pow(legTwo, 2))) + legOne + legTwo;

return perimeter;

}

public void comparator(Triangle heavensGate)

{

if (legOne == heavensGate.legOne && legTwo == heavensGate.legTwo)

{

Console.WriteLine("Triangles are equal.");

} else

{

Console.WriteLine("Triangles are not equal.");

}

}

public void basicOutputSystem()

{

Console.WriteLine("First leg is " + legOne + " miles long; second leg is " + legTwo + " miles long.");

Console.WriteLine("Area is " + areaCalculator() + " square miles; perimeter is " + perimeterCalculator() + " miles.");

}

public static double operator + (Triangle heavensGate)

{

return (heavensGate.legOne + heavensGate.legTwo);

}

public static double operator -(Triangle heavensGate)

{

if (heavensGate.legOne > heavensGate.legTwo)

{

return heavensGate.legOne - heavensGate.legTwo;

} else

{

return heavensGate.legTwo - heavensGate.legOne;

}

}

public static Triangle operator \*(Triangle heavensGate, double multiplier)

{

heavensGate.legOne \*= multiplier;

heavensGate.legTwo \*= multiplier;

return heavensGate;

}

public Triangle()

{

}

public Triangle(double legOneInput, double legTwoInput)

{

legOne = legOneInput;

legTwo = legTwoInput;

}

public Triangle(Triangle heavensGate)

{

legOne = heavensGate.legOne;

legTwo = heavensGate.legTwo;

}

}

static void Main(string[] args)

{

string logo = @" x

x x

x x

x x

x x x

x xxx x

x x x

x x

xxxxxxxxxxxxxxxxx";

Console.WriteLine(logo);

Triangle joyDivision = new Triangle();

joyDivision.basicInputSystem();

joyDivision.basicOutputSystem();

Triangle newOrder = new Triangle(joyDivision);

Console.WriteLine("Enter the multiplier below:");

double fiatMultipla = Convert.ToDouble(Console.ReadLine());

newOrder \*= fiatMultipla;

newOrder.basicOutputSystem();

joyDivision.comparator(newOrder);

Console.WriteLine("+ overload results: " + +newOrder);

Console.WriteLine("- overload results: " + -newOrder);

}

}

}

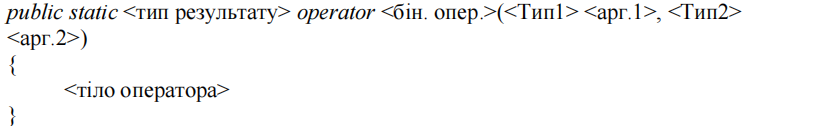
Результат роботи програми:

[screenshot]

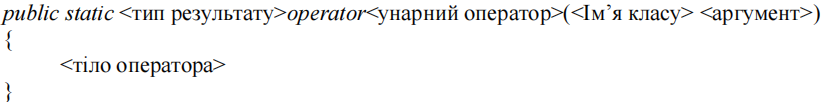
[screenshot]

1. **Контрольні запитання**
2. Опишіть загальний формат перевантаження операторів

Для бінарних операторів:



Для унарних операторів:



1. Які оператори можуть бути перевантажені?

Перевантаженими можуть бути унарні “+”, “-”, “!”, ” ~”, “++”, “*true”, “false”* табінарні “+”, “-”, “\*”, “/”, “%”, “&”, “|”, “^”, “<<”, “>>”, “==”, “!=”, “>”, “<”, “>=”, “<=”.

1. Що може бути результатом перевантаження оператора?

Результатом перевантаження оператора може бути змінна або екземпляр класу чи об’єкту.

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 площі та периметра прямокутного трикутника та набув навичок створення класів з використанням перевантаження операторів. Вважаю дану лабораторну роботу виконаною в повному обсязі.