**національний університет БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ україни**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Факультет інформаційних технологій

ЛАБОРАТОРНА РОБОТА №1  
СПАДКУВАННЯ І ПОЛІМОРФІЗМ У С#

Виконав:

студент групи ІПЗ-20006б

Симон Дмитрій Вікторович

Київ – 2023

**Мета роботи:** набуття вмінь і навичок роботи з конструювання ієрархії класів та використання віртуальних функцій.

**Призначення**: застосування динамічного поліморфізму, розуміння відмінності між раннім і пізнім зв'язуванням.

Завдання до лабораторної роботи

Описати базовий клас (можливо, абстрактний), у якому за допомогою віртуальних або абстрактних методів та властивостей задається інтерфейс для похідних класів.

В усіх класах потрібно перевизначити метод Equals, щоби порівнювались значення, а не посилання.

Визначити в класах усі можливі конструктори.

Метод Main повинен утримувати масив або список з елементів базового класу, заповнений посиланнями на похідні класи, та демонструвати використання усіх розроблених елементів класів відповідно варіанту.

Показати на прикладі одного з методів, присутніх у кожному класі, використання поліморфізму.

Обов’язково продемонструвати різницю між раннім та пізнім зв’язуванням

8. Базовий клас – трикутник (поля – довжини двох сторін і кут між ними), похідні – рівносторонній, рівнобедрений, прямокутний. Поліморфний метод виводить на екран значення площі та периметра.

Виконання завдання

using System;

abstract class Triangle

{

protected double side1, side2, angle;

public Triangle(double side1, double side2, double angle)

{

this.side1 = side1;

this.side2 = side2;

this.angle = angle;

}

public abstract double Area { get; }

public abstract double Perimeter { get; }

public abstract void Print();

}

class EquilateralTriangle : Triangle

{

public EquilateralTriangle(double side) : base(side, side, 60) { }

public override double Area => (Math.Sqrt(3) / 4) \* side1 \* side1;

public override double Perimeter => 3 \* side1;

public override bool Equals(object obj)

{

if (obj == null || GetType() != obj.GetType())

{

return false;

}

Triangle otherTriangle = (Triangle)obj;

return Area == otherTriangle.Area && Perimeter == otherTriangle.Perimeter;

}

public override void Print()

{

Console.WriteLine($"Рівносторонній трикутник:\nПлоща: {Area}\nПериметр: {Perimeter}\n");

}

}

class IsoscelesTriangle : Triangle

{

public IsoscelesTriangle(double baseSide, double equalSide, double angle) : base(baseSide, equalSide, angle) { }

public override double Area => 0.5 \* side1 \* side2 \* Math.Sin(angle);

public override double Perimeter => 2 \* side1 + side2;

public override bool Equals(object obj)

{

if (obj == null || GetType() != obj.GetType())

{

return false;

}

Triangle otherTriangle = (Triangle)obj;

return Area == otherTriangle.Area && Perimeter == otherTriangle.Perimeter;

}

public override void Print()

{

Console.WriteLine($"Рівнобедренний трикутник:\nПлоща: {Area}\nПериметр: {Perimeter}\n");

}

}

class RightTriangle : Triangle

{

public RightTriangle(double side1, double side2) : base(side1, side2, 90) { }

public override double Area => 0.5 \* side1 \* side2;

public override double Perimeter => side1 + side2 + Math.Sqrt(side1 \* side1 + side2 \* side2);

public override bool Equals(object obj)

{

if (obj == null || GetType() != obj.GetType())

{

return false;

}

Triangle otherTriangle = (Triangle)obj;

return Area == otherTriangle.Area && Perimeter == otherTriangle.Perimeter;

}

public override void Print()

{

Console.WriteLine($"Прямокутний трикутник:\nПлоща: {Area}\nПериметр: {Perimeter}\n");

}

}

class Program

{

static void Main(string[] args)

{

Triangle[] triangles = {

new EquilateralTriangle(5),

new IsoscelesTriangle(4, 6, 45),

new RightTriangle(3, 4)

};

foreach (Triangle triangle in triangles)

{

triangle.Print();

}

}

}

Результати виконання програмиЗображення, що містить текст, знімок екрана, програмне забезпечення, комп’ютер

Автоматично згенерований опис

Висновки

Навчився конструювати ієрархію класів, засвоїв поняття віртуальних функцій, динамічного поліморфізму та різницю між раннім і пізнім зв’язуванням. Застосував отриманні знання на практиці.