Лабораторна робота №11

Варіант 4

Завдання 1

**4.** Описати клас, який містять вказані поля і методи.

|  |  |
| --- | --- |
| **Клас** “Трикутник ” – TTriangle | |
| **поля** | * для зберігання довжин сторін; |
| **методи** | * конструктор без параметрів, конструктор з параметрами, конструктор копіювання; * введення/виведення даних; * визначення площі; * визначення периметру; * порівняння з іншим трикутником; * перевантаження операторів + (додавання довжин сторін), – (віднімання довжин відповідних сторін), \* (множення сторін на деяке число). |

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

try

{

Triangle tr1 = new Triangle(3, 4, 5);

Triangle tr2 = new Triangle(6, 8, 10);

Console.Write("Введіть число: ");

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

Console.WriteLine($"Площа(S) = {tr1.Area()}");

Console.WriteLine($"Периметр(P) = {tr1.Perimeter()}");

if (tr1 == tr2)

Console.WriteLine("Рівний");

else

Console.WriteLine("Не рівний");

Console.WriteLine($"Додавання {num}: {tr1 + num}");

Console.WriteLine($"Віднімання{num}: {tr1 - num}");

Console.WriteLine($"Множення {num}: {tr1 \* num}");

Triangle copiedTriangle = new Triangle(tr2);

Console.WriteLine(copiedTriangle);

}

catch

{

Console.WriteLine("Сторона має додатнє число");

}

}

}}

using System;

namespace task1

{

class Triangle

{

private double a;

private double b;

private double c;

public double A

{

get { return a; }

set

{

if (value < 0)

throw new Exception("Сторона має додатнє число");

a = value;

}

}

public double B

{

get { return b; }

set

{

if (value < 0)

throw new Exception("Сторона має додатнє число");

b = value;

}

}

public double C

{

get { return c; }

set

{

if (value < 0)

throw new Exception("Сторона має додатнє число");

c = value;

}

}

public Triangle(double userA, double userB, double userC)

{

A = userA;

B = userB;

C = userC;

}

public override string ToString()

{

return $"a = {a}, b = {b}, c = {c}";

}

// Конструктор копіювання

public Triangle(Triangle previousTriangle)

{

A = previousTriangle.a;

B = previousTriangle.b;

C = previousTriangle.c;

}

// Площа

public double Area()

{

double p = (a + b + c) / 2;

return Math.Sqrt(p \* (p - a) \* (p - b) \* (p - c));

}

// периметр

public double Perimeter()

{

return a + b + c;

}

//Порівняння трикутників

public static bool operator ==( Triangle tr1, Triangle tr2)

public static bool operator ==(Triangle tr1, Triangle tr2)

{

return tr1.a == tr2.a && tr1.b == tr2.b && tr1.c == tr2.c;

}

public static bool operator !=(Triangle tr1, Triangle tr2)

{

return !(tr1 == tr2);

}

// Перевантаження операції +

public static Triangle operator +(Triangle triangle, double num)

{

return new Triangle(triangle.a + num, triangle.b + num, triangle.c + num);

}

// Перевантаження операції-

public static Triangle operator -(Triangle triangle, double num)

{

return new Triangle(triangle.a - num, triangle.b - num, triangle.c - num);

}

// Перевантаження операції \*

public static Triangle operator \*(Triangle triangle, double num)

{

return new Triangle(triangle.a \* num, triangle.b \* num, triangle.c \* num);

}

}

}