Модульна робота

<https://github.com/igoryartym/C>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. Описати клас, який містять вказані поля і методи.   |  |  | | --- | --- | | **Клас** “прямокутний трикутник ” – TPTriangle | | | **поля** | * для зберігання довжин катетів; | | **методи** | * конструктор без параметрів, конструктор з параметрами, конструктор копіювання; * введення/виведення даних; * визначення площі; * визначення периметру; * порівняння з іншим трикутником; * перевантаження операторів + (додавання довжин катетів), – (віднімання довжин відповідних катетів), \* (множення сторін на деяке число). | |
| 2. Створити клас-нащадок TPPiramid (прямокутна трикутна піраміда, у якій бічне ребро перпендикулярне до катетів) на основі класу TPTriangle. Додати метод знаходження об’єму піраміди та перевизначити відповідні методи. |
| 3. Створити програму-клієнт для тестування. |

using System;

namespace модуль

{

class Program

{

static void Main(string[] args)

{

double a,b,c,d,num,rebro;

Console.Write("a= ");

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

Console.Write("b= ");

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

Console.Write("a of another triangle= ");

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

Console.Write("b of another triangle= ");

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

Console.Write("enter num= ");

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

Console.Write("enter rebro of triangle= ");

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

TPTriangle t = new TPTriangle(a, b);

Console.WriteLine($"Area: {t.Square()}");

Console.WriteLine($"Perem: {t.Perumeter()}");

t.IsEqual(c, d);

Console.WriteLine("+: {0}",a+b);

Console.WriteLine("-: {0}",a-b);

Console.WriteLine("\*: {0}, {1}",a\*num,b\*num);

TPPiramid p = new TPPiramid(rebro);

Console.WriteLine($"Volume= {p.Volume()}");

}

}

}

Клас\_TPTriangle

using System;

using System.Collections.Generic;

using System.Text;

namespace модуль

{

class TPTriangle

{

double a;

double b;

double hypoten;

public double A

{

get

{

return a;

}

set

{

if (value > 0) a = value;

else

throw new Exception("Lenght of side must be positive value");

}

}

public double B

{

get

{

return b;

}

set

{

if (value > 0) b = value;

else

throw new Exception("Lenght of side must be positive value");

}

}

public TPTriangle(double a, double b)

{

A = a;

B = b;

}

public TPTriangle(double a)

{

A = a;

B = a;

}

public TPTriangle()

{

A = 3;

B = 4;

}

public TPTriangle(TPTriangle t)

{

A = t.A;

B = t.B;

}

public double Square()

{

return (a \* b) \* 0.5;

}

public double Perumeter()

{

hypoten = Math.Sqrt(A \* A + B \* B);

return Math.Round(A + B + hypoten,2);

}

public void IsEqual(double a, double b)

{

if (A == a && B == b)

{

Console.WriteLine("Triangles are equal");

}

else if (A == a && B > b || A > a && B == b || A > a && B > b)

{

Console.WriteLine("First triangle is bigger");

}

else

Console.WriteLine("Second triangle is bigger");

}

public static double operator +(TPTriangle t)

{

return t.A + t.B;

}

public static double operator -(TPTriangle t)

{

return Math.Abs(t.A - t.B);

}

public static TPTriangle operator \*(TPTriangle t, double n)

{

return new TPTriangle(t.A \* n, t.B \* n);

}

}

}

Клас\_TPPiramid

using System;

using System.Collections.Generic;

using System.Text;

namespace модуль

{

class TPPiramid:TPTriangle

{

double rebro;

public double Redro

{

get

{

return rebro;

}

set

{

if (value > 0) rebro = value;

else

throw new Exception("Lenght of side must be positive value");

}

}

public TPPiramid(double rebro)

{

this.rebro = rebro;

}

public double Volume()

{

return Math.Round(Square() \* rebro,2);

}

}

}

Контрольний приклад.

