Лабораторна робота №12

**Завдання 1.**

Реалізувати завдання попередньої лабораторної роботи з використанням властивостей для доступу до полів**.**

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

double r1, r2, num;

Console.Write("Enter radius1: ");

r1 = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter radius2: ");

r2 = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter num: ");

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

TCirle tcirle = new TCirle(r1, r2);

Console.WriteLine("Area: {0}", tcirle.Area());

Console.WriteLine("Len circle: {0}", tcirle.Len());

Console.WriteLine("r1>r2: {0}", tcirle.Diff());

Console.WriteLine("+: {0}",r1 + r2);

Console.WriteLine("-: {0}",r1 - r2);

Console.WriteLine("\*: {0}",r1 \* num);

}

}

}

Клас\_\_

using System;

namespace task1

{

class TCirle

{

public double radius1;

public double radius2;

public double R1

{

get

{

return radius1;

}

set

{

if (value > 0) radius1 = value;

else

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

}

}

public double R2

{

get

{

return radius2;

}

set

{

if (value > 0) radius2 = value;

else

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

}

}

public TCirle()

{

radius1 = 5;

radius2 = 7;

}

public TCirle(double radius1, double radius2)

{

this.radius1 = radius1;

this.radius2 = radius2;

}

public double Area()

{

return 2 \* Math.PI \* Math.Pow(radius1, 2);

}

public double Len()

{

return 2 \* Math.PI \* radius1;

}

public bool Diff()

{

bool a = false;

if (radius1 > radius2)

a = true;

return a;

}

public double Value { get; set; }

public static TCirle operator +(TCirle radius1, TCirle radius2)

{

return new TCirle { Value = radius1.Value + radius2.Value };

}

public static TCirle operator -(TCirle radius1, TCirle radius2)

{

return new TCirle { Value = radius1.Value - radius2.Value };

}

public static TCirle operator \*(TCirle radius1, double a)

{

return new TCirle { Value = radius1.Value \* a };

}

}

}

**Завдання 2.**

Створити клас TBankomat, який моделює роботу банкомата. Клас повинен містити поля для зберігання кількості купюр кожного із номіналів від 5 до 200 гривень. Реалізувати методи знаходження максимальної та мінімальної сум, які може видати банкомат, та метод зняття деякої суми.

Контрольний приклад.

**Завдання 3.**

Реалізувати клас, що представляє трикутник (трикутник задається координатами вершин) і містить опис індексатора для доступу до сторін трикутника( – перша сторона,  – друга сторона, – третя сторона). Передбачити методи введення/виведення, знаходження периметру та площі.

using System;

namespace task3

{

class Program

{

static void Main(string[] args)

{

double[] xn = new double[3];

double[] yn = new double[3];

for (int i = 0; i < 3; i++)

{

Console.Write("Enter x{0}: ",i+1);

xn[i] = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter y{0}: ", i+1);

yn[i] = Convert.ToInt32(Console.ReadLine());

}

TTriangle tr = new TTriangle(xn, yn);

Console.WriteLine("a = {0}, b ={1}, c = {2}",tr.a(),tr.b(),tr.c());

Console.WriteLine("Area: {0}"+"\n"+"Perum: {1}",tr.Square(),tr.Perum());

}

}

}

Клас\_

using System;

using System.Collections.Generic;

using System.Text;

namespace task3

{

class TTriangle

{

double[] xn;

double[] yn;

public TTriangle(double[] xn, double[] yn)

{

this.xn = xn;

this.yn = yn;

}

public double a()

{

return Math.Sqrt(Math.Pow((xn[1] - xn[0]), 2) + Math.Pow((yn[1] - yn[0]), 2));

}

public double b()

{

return Math.Sqrt(Math.Pow((xn[2] - xn[1]), 2) + Math.Pow((yn[2] - yn[1]), 2));

}

public double c()

{

return Math.Sqrt(Math.Pow((xn[2] - xn[0]), 2) + Math.Pow((yn[2] - yn[0]), 2));

}

double A, B, C;

TTriangle(TTriangle t)

{

A = t.a();

B = t.b();

C = t.c();

}

public double Square()

{

double p = (A+B+C)/2;

return Math.Round(Math.Sqrt(p\*(p-A)\*(p-B)\*(p-C)));

}

public double Perum()

{

return A + B + C;

}

}

}

Контрольний приклад.