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 TCirle(TCirle c)

{

this.radius1 = c.radius1;

this.radius2 = c.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.

using System;

namespace task2

{

class Program

{

static void Main(string[] args)

{

int a5, a10, a20, a50, a100, a200,sum;

Console.Write("Enter number of 5: ");

a5 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter number of 10: ");

a10 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter number of 20: ");

a20 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter number of 50: ");

a50 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter number of 100: ");

a100 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter number of 200: ");

a200 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the amount you want to withdraw: ");

sum = Convert.ToInt32(Console.ReadLine());

TBankomat bank = new TBankomat(a5, a10, a20, a50, a100, a200,sum);

Console.WriteLine("Min = {0}"+"\n"+"Max = {1}",bank.Minsum(),bank.Maxsum());

Console.WriteLine("Cash = {0}",bank.Amount());

}

}

}

**Клас\_** TBankomat

using System;

namespace task2

{

class TBankomat

{

int a5;

int a10;

int a20;

int a50;

int a100;

int a200;

int sum;

public TBankomat(int a5, int a10, int a20, int a50, int a100, int a200, int sum)

{

this.a5 = a5;

this.a10 = a10;

this.a20 = a20;

this.a50 = a50;

this.a100 = a100;

this.a200 = a200;

this.sum = sum;

}

public int Minsum()

{

if (a5 > 0) return 5;

else if (a10 > 0) return 10;

else if (a20 > 0) return 20;

else if (a50 > 0) return 50;

else if (a100 > 0) return 100;

else if (a200 > 0) return 200;

else return 0;

}

public uint Maxsum()

{

uint a = Convert.ToUInt32(5 \* a5 + 10 \* a10 + 20 \* a20 + 50 \* a50 + a100 \* 100 + 200 \* a200);

return a;

}

string s;

int a= 0;

public string Amount()

{

if (Maxsum() > sum)

{

if (sum < 5) return sum.ToString();

while (sum > 5)

{

if (sum / 200 > 0)

{

a = sum / 200;

sum -= 200 \* a;

s += "200\*" + a;

}

else if (sum / 100 > 0)

{

a = sum / 100;

sum -= 100 \* a;

s += "+100\*" + a;

}

else if (sum / 50 > 0)

{

a = sum / 50;

sum -= 50 \* a;

s += "+50\*" + a;

}

else if (sum / 20 > 0)

{

a = sum / 20;

sum -= 20 \* a;

s += "+20\*" + a;

}

else if (sum / 10 > 0)

{

a = sum / 10;

sum -= 10 \* a;

s += "+10\*" + a;

}

else

{

a = sum / 5;

sum -= 5 \* a;

s += "+5\*" + a;

}

}

return s;

}

else

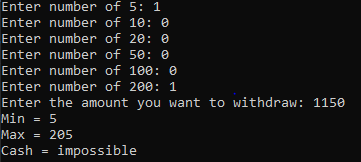
return "impossible";

}

}

}

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

}

public double b()

{

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

}

public double c()

{

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

}

public double A

{

get

{

return a();

}

}

public double B

{

get

{

return b();

}

}

public double C

{

get

{

return c();

}

}

public double Square()

{

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

return Math.Round(Math.Sqrt(p \* (p - A) \* (p - B) \* (p - C)),2);

}

public double Perum()

{

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

}

}

}

Контрольний приклад.

