Лабораторная робота 9

Методы. Конструкторы. Перечисления

Паршин Олександр

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_9

{

public enum Color { Bely, Zelty, Krasny };

class Pryamougolnik

{

public double x1 { get; protected set; }

public double x2 { get; protected set; }

public double y1 { get; protected set; }

public double y2 { get; protected set; }

private Color colorFon;

private Color colorLine;

private double deafAngle = 0.0;

private double angle = 0.0;

public Pryamougolnik(double \_x1, double \_y1, double \_x2, double \_y2)

{

x1 = \_x1;

x2 = \_x2;

y1 = \_y1;

y2 = \_y2;

DeafAngle();

}

public Pryamougolnik(double height, double width)

{

x1 = -(width / 2);

y1 = (height / 2);

x2 = (width / 2);

y2 = -(height / 2);

DeafAngle();

}

private void DeafAngle()

{

double \_tempGip = Math.Sqrt((x2 - x1) \* (x2 - x1) + (y2 - y1) \* (y2 - y1));

double \_tempKat = Math.Sqrt((x2 - x1) \* (x2 - x1) + (y2 - y2) \* (y2 - y2));

deafAngle = 90 - ((Math.Asin(\_tempKat / \_tempGip) \* 180) / Math.PI);

}

public void info()

{

double \_tempX;

double \_tempY;

double length = Math.Sqrt((x2 - x1) \* (x2 - x1) + (y2 - y1) \* (y2 - y1));

\_tempX = length \* Math.Cos((deafAngle + angle) \* Math.PI / 180) + x1;

\_tempY = length \* Math.Sin((deafAngle + angle) \* Math.PI / 180 \* (-1)) + y1;

Console.WriteLine("x1 = {0,3:f2} y1 = {1,3:f2}", x1, y1);

Console.WriteLine("x2 = {0,3:f2} y2 = {1,3:f2}", \_tempX, \_tempY);

Console.WriteLine("Цвет фона {0}",colorFon);

Console.WriteLine("Цвет границ {0}\n",colorLine);

}

public double areaRectangle()

{

double width = Math.Sqrt( (x2-x1)\*(x2-x1) + (y1-y1)\* (y1 - y1));

double height = Math.Sqrt((x1 - x1) \* (x1 - x1) + (y2 - y1) \* (y2 - y1));

return width \* height;

}

public double perimeterRectangle()

{

double width = Math.Sqrt((x2 - x1) \* (x2 - x1) + (y1 - y1) \* (y1 - y1));

double height = Math.Sqrt((x1 - x1) \* (x1 - x1) + (y2 - y1) \* (y2 - y1));

return 2 \* (width + height);

}

public void moveX(double a)

{

x1 += a;

x2 += a;

}

public void moveY(double a)

{

y1 += a;

y2 += a;

}

public void scale(double a)

{

x2 \*= a;

y2 \*= a;

}

public void setColorFon(Color a)

{

colorFon = a;

}

public void setColorLine(Color a)

{

colorLine = a;

}

public double areaSquare()

{

double width = Math.Sqrt((x2 - x1) \* (x2 - x1) + (y1 - y1) \* (y1 - y1));

double height = Math.Sqrt((x1 - x1) \* (x1 - x1) + (y2 - y1) \* (y2 - y1));

return Math.Min(width, height) \* Math.Min(width, height);

}

public void Rotate(double Angle)

{

angle = Angle;

}

public void MoveTo(double \_tempAngle, double \_tempLength)

{

double \_tempX;

double \_tempY;

\_tempX = \_tempLength \* Math.Cos(\_tempAngle \* Math.PI / 180);

\_tempY = \_tempLength \* Math.Sin(\_tempAngle \* Math.PI / 180);

x1 += \_tempX;

y1 += \_tempY;

x2 += \_tempX;

y2 += \_tempY;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_9

{

class Program

{

static void Main(string[] args)

{

Pryamougolnik ABCD = new Pryamougolnik(0, 0, 2, 2);

Pryamougolnik EFGH = new Pryamougolnik(4,6);

ABCD.info();

Console.WriteLine("Площадь прямоугольника ABCD {0,4:f3}", ABCD.areaRectangle());

ABCD.Rotate(30);

ABCD.setColorFon(Color.Krasny);

ABCD.setColorLine(Color.Zelty);

ABCD.info();

Console.WriteLine("Площадь прямоугольника ABCD {0,4:f3}\n", ABCD.areaRectangle());

Console.WriteLine("Площадь прямоугольника EFGH {0,4:f3}", EFGH.areaRectangle());

EFGH.info();

Console.WriteLine("Для выхода нажмите Enter");

Console.ReadLine();

}

}

}

