

КАМ'ЯНЕЦЬ-ПОДІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ОГІЄНКА

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

**ЛАБОРАТОРНА РОБОТА №4**  
З ДИСЦИПЛІНИ «ОБ'ЄКТНО-ОРІЄНТОВАНЕ  
ПРОГРАМУВАННЯ»

Поліморфізм

**Виконав:**

студент 1-го курсу

гр. KNms1-B21

Шевчук О. В.

**Прийняв:**

доцент,

Іванюк В. А.

Дата здачі «\_\_»\_\_\_\_\_2021 р

## ЗАВДАННЯ 1

### Код програми:

```
class Quadrate
{
    protected double a { get; set; }
    protected string color { get; set; }

    public Quadrate(double _a)
    {
        this.a = _a;
    }

    public virtual void Display()
    {
        color = "gray";
        Console.WriteLine($"Розмір квадрата: {a} та початковий колір {color}");
    }
}

class Point : Quadrate
{
    protected double x1 { get; set; }
    protected double y1 { get; set; }
    protected double x2 { get; set; }
    protected double y2 { get; set; }

    protected double[] x = new double[2];
    protected double[] y = new double[2];

    public Point(double a) : base(a)
    {
        this.x1 = -a;
        this.y1 = a;
        this.x2 = a;
        this.y2 = -a;
        x[0] = x1;
        x[1] = x2;
        y[0] = y1;
        y[1] = y2;
    }

    public override void Display()
    {
        Console.WriteLine("Координати: ");
        Console.WriteLine($"X = ({x[0]}, {x[1]}), Y = ({y[0]}, {y[1]})");
    }
}

class Segment : Point
{
    public Segment(double a) : base(a) { }

    private void Stretching()
    {
        Console.WriteLine("Розтягнення вдвічі");
        for (int i = 0; i < x.Length; i++)
        {
            x[i] = x[i] * 2;
        }
        for (int i = 0; i < y.Length; i++)
        {
            y[i] = y[i] * 2;
        }
    }
}
```

```

    }
    Console.WriteLine($"X = ({x[0]}, {x[1]}), Y = ({y[0]}, {y[1]})");
}

private void Compression()
{
    Console.WriteLine("Стиснення у півтора раза");
    for (int i = 0; i < x.Length; i++)
    {
        x[i] = Math.Round(x[i] / 1.5, 2);
    }
    for (int i = 0; i < y.Length; i++)
    {
        y[i] = Math.Round(y[i] / 1.5, 2);
    }
    Console.WriteLine($"X = ({x[0]}, {x[1]}), Y = ({y[0]}, {y[1]})");
}

public override void Display()
{
    Stretching();
    Compression();
    Console.WriteLine("Поворот ліворуч");
    Console.WriteLine($"X = ({x[0]}, {x[1]}), Y = ({y[0]}, {y[1]})");
    Console.WriteLine("Зміна кольору");
    color = Console.ReadLine();
    Console.WriteLine($"Колір фігури за таким координатами: X = ({x[0]},
{x[1]}), Y = ({y[0]}, {y[1]}) становить {color}");
}

}

class Program
{
    static void Main(string[] args)
    {
        double a = Convert.ToDouble(Console.ReadLine());
        Quadrate q = new Quadrate(a);
        q.Display();
        Point p = new Point(a);
        p.Display();
        Segment s = new Segment(a);
        s.Display();
    }
}

```

## Робота програми:

```

5,25
Розмір квадрата: 5,25 та початковий колір gray
Координати:
X = (-5,25, 5,25), Y = (5,25, -5,25)
Розтягнення вдвічі
X = (-10,5, 10,5), Y = (10,5, -10,5)
Стиснення у півтора раза
X = (-7, 7), Y = (7, -7)
Поворот ліворуч
X = (-7, 7), Y = (7, -7)
Зміна кольору
black
Колір фігури за таким координатами: X = (-7, 7), Y = (7, -7) становить black

```

## ЗАВДАННЯ 2

### Код програми:

```
abstract class TVector
{
    protected double a1 { get; set; }
    protected double a2 { get; set; }
    protected double b1 { get; set; }
    protected double b2 { get; set; }
    protected double sum { get; set; }

    protected double[] vA = new double[2];
    protected double[] vB = new double[2];

    public TVector(double a1_, double b1_, double a2_, double b2_)
    {
        this.a1 = a1_;
        this.a2 = a2_;
        this.b1 = b1_;
        this.b2 = b2_;
        vA[0] = a1;
        vA[1] = a2;
        vB[0] = b1;
        vB[1] = b2;
    }

    public abstract string Sum { get; }

    public override string ToString()
    {
        return Sum;
    }
}

class VectorTwo : TVector
{
    public VectorTwo(double a1, double b1, double a2, double b2) : base(a1, a2, b1,
b2) { }

    public override string Sum
    {
        get
        {
            double rez = 0, q = 0, a = 0, b = 0;
            for (int i = 0; i < vA.Length; i++)
            {
                rez += vA[i] * vB[i];
            }
            if (rez == 0)
            {
                for (int i = 0; i < vA.Length; i++)
                {
                    q += Math.Pow(vA[i], 2);
                }
                a = Math.Round(Math.Sqrt(q), 2);
                q = 0;
                for (int i = 0; i < vB.Length; i++)
                {
                    q += Math.Pow(vB[i], 2);
                }
                b = Math.Round(Math.Sqrt(q), 2);
                sum = a + b;
            }
        }
    }
}
```

```

        if (sum != 0)
        {
            return $"Суму паралельних довжин двовимірних векторів = {sum}";
        }
        return "Дані вектори не паралельні, тому їхню суму визначити
неможливо";
    }
}

```

```

class VectorThree : TVector
{
    protected double a3 { get; set; }
    protected double b3 { get; set; }

    protected double[] vC = new double[3];
    protected double[] vD = new double[3];

    public VectorThree(double a1, double b1, double a2, double b2, double a3,
double b3) : base(a1, a2, b1, b2)
    {
        this.a3 = a3;
        this.b3 = b3;
        vC[0] = a1;
        vC[1] = a2;
        vC[2] = a3;
        vD[0] = b1;
        vD[1] = b2;
        vD[2] = b3;
    }

    public override string Sum
    {
        get
        {
            double q = 0, a = 0, b = 0;
            int z = 0;
            for (int i = 1; i < vC.Length; i++)
            {
                double def = vC[i - 1] / vD[i - 1];
                if (def != vC[i] / vD[i])
                {
                    z--;
                    break;
                }
                else if (def == vC[i] / vD[i])
                {
                    z++;
                }
            }
            if (z > 0)
            {
                for (int i = 0; i < vC.Length; i++)
                {
                    q += Math.Pow(vC[i], 2);
                }
                a = Math.Round(Math.Sqrt(q), 2);
                q = 0;
                for (int i = 0; i < vD.Length; i++)
                {
                    q += Math.Pow(vD[i], 2);
                }
                b = Math.Round(Math.Sqrt(q), 2);
                sum = a + b;
            }
        }
    }
}

```

```

        if (sum != 0)
        {
            return $"Суму перпендикулярних довжин тривимірних векторів =
{sum}";
        }
    }
    return "Дані вектори не перпендикулярні, тому їхню суму визначити
неможливо";
}
}

class Program
{
    static void Main(string[] args)
    {
        TVector[] vector = {
            new VectorTwo(2.5, 4, -2, 1.25),
            new VectorTwo(2, 1, -1, 2),
            new VectorTwo(6, 5.5, 4, 0.75),
            new VectorThree(2.4, 1.2, 7.2, 3.6, 9.6, 4.8),
            new VectorThree(2, 2, 2, 2, 2, 2),
            new VectorThree(2, 7, 1, -6, 21, 3),
            new VectorThree(2.5, 2, 1.25, 1, 4, 3.2)
        };
        foreach (TVector str in vector)
        {
            Console.WriteLine(str);
        }
    }
}

```

## Робота програми:

Суму паралельних довжин двовимірних векторів = 7,08  
 Суму паралельних довжин двовимірних векторів = 4,48  
 Дані вектори не паралельні, тому їхню суму визначити неможливо  
 Суму перпендикулярних довжин тривимірних векторів = 18,36  
 Суму перпендикулярних довжин тривимірних векторів = 6,92  
 Дані вектори не перпендикулярні, тому їхню суму визначити неможливо  
 Суму перпендикулярних довжин тривимірних векторів = 8,78