

## BÀI 4. OPERATOR OVERLOADING

### Bài 1. Xây dựng lớp Vector

Một **vector** trong không gian 2 chiều gồm có 2 phần **x** và **y**.

Ví dụ:

$$\vec{a} = (1, 2)$$

$$\vec{b} = (3, 4)$$

**Tổng** 2 vector được định nghĩa :  $\vec{c} = \vec{a} + \vec{b} = (1 + 3, 2 + 4) = (4, 6)$

**Hiệu** của 2 vector được định nghĩa :  $\vec{d} = \vec{a} - \vec{b} = (1 - 3, 2 - 4) = (-2, -2)$

Hãy xây dựng lớp **Vector**:

- Khai báo cáo **fields**
- Viết **default constructor** tạo vector (0, 0)
- Viết **constructor** tạo vector có 2 thành phần x và y
- Cài đặt các **properties**
- **Overload** phép toán cộng 2 vector
- **Overload** phép toán trừ 2 vector

Viết phương thức **Main()** tạo 2 đối tượng vector, tính tổng và hiệu 2 vector đó, sau đó xuất kết quả ra màn hình để kiểm tra kết quả đúng hay không.

**File Vector.cs**

```
using System;

namespace ThuchanhOOP
{
    class Vector
    {
        private int x;
        private int y;

        public int X { get => x; set => x = value; }
        public int Y { get => y; set => y = value; }

        public Vector()
        {
            X = 0;
            Y = 0;
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 4

```
public Vector(int _x, int _y)
{
    X = _x;
    Y = _y;
}

public static Vector operator +(Vector a, Vector b)
{
    Vector c = new Vector();
    c.X = a.X + b.X;
    c.Y = a.Y + b.Y;

    return c;
}

public static Vector operator -(Vector a, Vector b)
{
    Vector c = new Vector();
    c.X = a.X - b.X;
    c.Y = a.Y - b.Y;

    return c;
}

public string Output()
{
    return $"({X},{Y})";
}
}
```

### File Program.cs

```
using System;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            Vector a = new Vector(2, 3); // a = (2,3)
            Vector b = new Vector(1, -7); // b = (1,-7)

            Console.WriteLine($"Vector a = {a.Output()}");
            Console.WriteLine($"Vector b = {b.Output()}");
            Console.WriteLine();

            Vector c = a + b;

            Console.WriteLine("Tong cua vector a va b:");
            Console.WriteLine($"{a.Output()} + {b.Output()} = {c.Output()}");
            Console.WriteLine();

            Console.WriteLine("Hieu cua vector a va b:");
            Console.WriteLine($"{a.Output()} - {b.Output()} = {(a - b).Output()}");
            Console.WriteLine($"{b.Output()} - {a.Output()} = {(b - a).Output()}");

            Console.ReadKey();
        }
    }
}
```

## Bài 2. Overload phép toán so sánh

Hai vector được gọi là bằng nhau khi các thành phần tương ứng của chúng bằng nhau, nghĩa là với:  $\vec{a} = (a_1, a_2)$ ;  $\vec{b} = (b_1, b_2)$ ;

$$\vec{a} == \vec{b} \Leftrightarrow \begin{cases} a_1 = b_1 \\ a_2 = b_2 \end{cases}$$

Trong lớp Vector hãy overload phép toán so sánh bằng (==) giữa hai vector.

Viết phương thức **Main()** kiểm tra phép toán == vừa overload

### File Vector.cs

```
namespace ThuchanhOOP
{
    class Vector
    {
        private int x;
        private int y;

        public int X { get => x; set => x = value; }
        public int Y { get => y; set => y = value; }

        public Vector()
        {
            X = 0;
            Y = 0;
        }
        public Vector(int _x, int _y)
        {
            X = _x;
            Y = _y;
        }

        public static Vector operator +(Vector a, Vector b)
        {
            Vector c = new Vector();
            c.X = a.X + b.X;
            c.Y = a.Y + b.Y;

            return c;
        }
        public static Vector operator -(Vector a, Vector b)
        {
            Vector c = new Vector();
            c.X = a.X - b.X;
            c.Y = a.Y - b.Y;

            return c;
        }
        public static bool operator ==(Vector a, Vector b)
        {
            if (a.X == b.X && a.Y == b.Y)
                return true;
            else
                return false;
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 4

```
public static bool operator !=(Vector a, Vector b)
{
    if (a.X != b.X || a.Y != b.Y)
        return true;
    else
        return false;
}

public string Output()
{
    return $"({X},{Y})";
}
}
```

### File Program.cs

```
using System;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            Vector a = new Vector(2, 3); // a = (2,3)
            Vector b = new Vector(1, -7); // b = (1,-7)

            Console.WriteLine($"Vector a = {a.Output()}");
            Console.WriteLine($"Vector b = {b.Output()}");
            Console.WriteLine();

            if (a == b)
                Console.WriteLine("Vector a bang vector b.");
            else
                Console.WriteLine("Vector a khong bang vector b.");

            Console.ReadKey();
        }
    }
}
```

### Bài 3. Cộng số nguyên với vector

Cộng vector với một số nguyên được định nghĩa như sau:

Cho  $\vec{a} = (a_1, a_2)$  và số nguyên  $k$

$$\vec{a} + k = (a_1 + k, a_2 + k)$$

Trong lớp vector hãy

- **Overload** phép toán cộng vector với 1 số nguyên
- **Overload** phép toán cộng 1 số nguyên với 1 vector

#### File Vector.cs

```
namespace ThuchanhOOP
{
    class Vector
    {
        private int x;
        private int y;

        public int X { get => x; set => x = value; }
        public int Y { get => y; set => y = value; }

        public Vector()
        {
            X = 0;
            Y = 0;
        }
        public Vector(int _x, int _y)
        {
            X = _x;
            Y = _y;
        }

        public static Vector operator +(Vector a, Vector b)
        {
            Vector c = new Vector();
            c.X = a.X + b.X;
            c.Y = a.Y + b.Y;

            return c;
        }
        public static Vector operator -(Vector a, Vector b)
        {
            Vector c = new Vector();
            c.X = a.X - b.X;
            c.Y = a.Y - b.Y;

            return c;
        }
        public static Vector operator +(Vector a, int k)
        {
            Vector c = new Vector();
            c.X = a.X + k;
            c.Y = a.Y + k;
            return c;
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 4

```
public static Vector operator +(int k, Vector a)
{
    Vector c = new Vector();
    c.X = a.X + k;
    c.Y = a.Y + k;

    return c;
}
public static bool operator ==(Vector a, Vector b)
{
    if (a.X == b.X && a.Y == b.Y)
        return true;
    else
        return false;
}
public static bool operator !=(Vector a, Vector b)
{
    if (a.X != b.X || a.Y != b.Y)
        return true;
    else
        return false;
}

public string Output()
{
    return $"({X},{Y})";
}
}
```

### Bài 4. Lớp Phân số

Phân số (**PhanSo**) gồm có tử số và mẫu số. Hãy xây dựng lớp Phân số:

- Khai báo các **fields**
- **Constructor** không tham số để tạo phân số 0/1
- **Constructor** có tham số để tạo phân số có 2 thông tin trên
- Cài đặt các **Properties**
- Phương thức nhập phân số từ bàn phím
- Phương thức xuất phân số ra màn hình
- **Overload** các phép toán cộng (+), trừ (-) hai phân số
- **Overload** các phép toán nhân (\*), chia (/) hai phân số

Viết phương thức **Main()** nhập vào 2 phân số, tính tổng, hiệu, tích, thương của 2 phân số.

VD:

```
Nhap phan so thu 1:
tu = 1
mau = 2

Nhap phan so thu 2:
tu = 3
mau = 2

1/2 + 3/2 = 4/2
1/2 - 3/2 = -2/2
1/2 * 3/2 = 3/4
1/2 / 3/2 = 2/6
```

### File PhanSo.cs

```
using System;

namespace ThuchanhOOP
{
    class PhanSo
    {
        private int ts;
        private int ms;

        public int TS { get => ts; set => ts = value; }
        public int MS
        {
            get => ms;
            set
            {
                if (value!=0)
                    ms = value;
            }
        }

        public PhanSo()
        {
            TS = 0;
            MS = 1;
        }
        public PhanSo(int _ts, int _ms)
        {
            TS = _ts;
            MS = _ms;
        }
        public PhanSo(int _ts)
        {
            TS = _ts; MS = 1;
        }

        public void Input()
        {
            Console.Write("tu = ");
            TS = int.Parse(Console.ReadLine());
            Console.Write("mau = ");
            MS = int.Parse(Console.ReadLine());
        }
    }
}
```

```
public string Output()
{
    string tmp = "";
    if (TS == 0)
        tmp = "0";
    else if (MS == 1)
        tmp = TS.ToString();
    else
        tmp = $"{TS}/{MS}";

    return tmp;
}

public static PhanSo operator + (PhanSo a, PhanSo b)
{
    PhanSo c = new PhanSo();
    c.TS = (a.TS*b.MS) + (b.TS*a.MS);
    c.MS = a.MS * b.MS;

    return c;
}

public static PhanSo operator - (PhanSo a, PhanSo b)
{
    PhanSo c = new PhanSo();
    c.TS = (a.TS * b.MS) - (b.TS * a.MS);
    c.MS = a.MS * b.MS;

    return c;
}

public static PhanSo operator *(PhanSo a, PhanSo b)
{
    PhanSo c = new PhanSo();
    c.TS = a.TS * b.TS ;
    c.MS = a.MS * b.MS;

    return c;
}

public static PhanSo operator /(PhanSo a, PhanSo b)
{
    PhanSo c = new PhanSo();
    c.TS = a.TS * b.MS;
    c.MS = a.MS * b.TS;

    return c;
}

public void Toigian()
{
    int ucln = UCLN(TS, MS);

    TS = TS / ucln;
    MS = MS / ucln;
}
```



## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 4

```
private int UCLN(int a, int b)
{
    a = Math.Abs(a);
    b = Math.Abs(b);

    while (a != b)
        if (a > b)
            a = a - b;
        else
            b = b - a;

    return a;
}
```

### File Program.cs

```
using System;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhap phan so thu 1:");
            PhanSo a = new PhanSo();
            a.Input();
            Console.WriteLine("Nhap phan so thu 2:");
            PhanSo b = new PhanSo();
            b.Input();

            Console.WriteLine($"Phan so a = {a.Output()}");
            Console.WriteLine($"Phan so b = {b.Output()}");
            Console.WriteLine();

            PhanSo c = a + b;
            c.Toigian();
            Console.WriteLine($"a + b = {c.Output()}");
            Console.WriteLine();

            c = a - b;
            c.Toigian();
            Console.WriteLine($"a - b = {c.Output()}");
            Console.WriteLine();

            c = b - a;
            c.Toigian();
            Console.WriteLine($"b - a = {c.Output()}");
            Console.WriteLine();

            c = a * b;
            c.Toigian();
            Console.WriteLine($"a * b = {c.Output()}");
            Console.WriteLine();

            c = a / b;
            c.Toigian();
            Console.WriteLine($"a / b = {c.Output()}");
            Console.WriteLine();
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 4

```
        c = b / a;
        c.Toigian();
        Console.WriteLine($"b / a = {c.Output()}");
        Console.WriteLine();

        Console.ReadKey();
    }
}
```

## Bài 5. Sử dụng lớp Phân số

Sử dụng lớp Phân số trên thực hiện các chức năng

- Nhập n phân số (n nhập từ bàn phím) (Có thể dùng **List** nếu đã học)
- Tính tổng n phân số và xuất kết quả ra màn hình

### File Program.cs

```
using System;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Nhap vao so luong phan so can tao: ");
            int n = int.Parse(Console.ReadLine());

            PhanSo[] PSArray = new PhanSo[n];

            for (int i = 0; i < PSArray.Length; i++)
            {
                PSArray[i] = new PhanSo();
                PSArray[i].Input();
            }

            PhanSo tong = PSArray[0];
            for (int i = 1; i < PSArray.Length; i++)
                tong += PSArray[i];

            tong.Toigian();
            Console.WriteLine($"Tong cac phan so la: {tong.Output()}");

            Console.ReadKey();
        }
    }
}
```

## Bài 6. Các phép toán so sánh trong phân số

Trong lớp Phân số, hãy cài đặt thêm các phép toán so sánh

Trong phương thức **Main()**

- **Nhập** n phân số
- **Sắp xếp** các phân số tăng dần và xuất ra màn hình để kiểm chứng
- **Tìm** phân số có giá trị lớn nhất và xuất ra màn hình phân số này

### File PhanSo.cs

```
using System;

namespace ThuchanhOOP
{
    class PhanSo
    {
        private int ts;
        private int ms;

        public int TS { get => ts; set => ts = value; }
        public int MS
        {
            get => ms;
            set
            {
                if (value!=0)
                    ms = value;
            }
        }

        public PhanSo()
        {
            TS = 0;
            MS = 1;
        }
        public PhanSo(int _ts, int _ms)
        {
            TS = _ts;
            MS = _ms;
        }
        public PhanSo(int _ts)
        {
            TS = _ts;
            MS = 1;
        }

        public void Input()
        {
            Console.Write("tu = ");
            TS = int.Parse(Console.ReadLine());
            Console.Write("mau = ");
            MS = int.Parse(Console.ReadLine());
        }
    }
}
```

```
public string Output()
{
    string tmp = "";
    if (TS == 0)
        tmp = "0";
    else if (MS == 1)
        tmp = TS.ToString();
    else
        tmp = $"{TS}/{MS}";

    return tmp;
}

public static PhanSo operator + (PhanSo a, PhanSo b)
{
    PhanSo c = new PhanSo();
    c.TS = (a.TS*b.MS) + (b.TS*a.MS);
    c.MS = a.MS * b.MS;

    return c;
}

public static PhanSo operator - (PhanSo a, PhanSo b)
{
    PhanSo c = new PhanSo();
    c.TS = (a.TS * b.MS) - (b.TS * a.MS);
    c.MS = a.MS * b.MS;

    return c;
}

public static PhanSo operator *(PhanSo a, PhanSo b)
{
    PhanSo c = new PhanSo();
    c.TS = a.TS * b.TS ;
    c.MS = a.MS * b.MS;

    return c;
}

public static PhanSo operator /(PhanSo a, PhanSo b)
{
    PhanSo c = new PhanSo();
    c.TS = a.TS * b.MS;
    c.MS = a.MS * b.TS;

    return c;
}

public void Toigian()
{
    int ucln = UCLN(TS, MS);

    TS = TS / ucln;
    MS = MS / ucln;
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 4

```
private int UCLN(int a, int b)
{
    a = Math.Abs(a);
    b = Math.Abs(b);

    while (a != b)
        if (a > b)
            a = a - b;
        else
            b = b - a;

    return a;
}

public static bool operator >(PhanSo a, PhanSo b)
{
    if (a.TS * b.MS > b.TS * a.MS)
        return true;
    else
        return false;
}

public static bool operator <(PhanSo a, PhanSo b)
{
    if (a.TS * b.MS < b.TS * a.MS)
        return true;
    else
        return false;
}
}
```

### File Program.cs

```
using System;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Nhap vao so luong phan so can tao: ");
            int n = int.Parse(Console.ReadLine());

            PhanSo[] PSArray = new PhanSo[n];

            for (int i = 0; i < PSArray.Length; i++)
            {
                PSArray[i] = new PhanSo();
                PSArray[i].Input();
            }
            Console.WriteLine();

            for (int i = 0; i < PSArray.Length - 1; i++)
                for (int j = i + 1; j < PSArray.Length; j++)
                    if (PSArray[i] > PSArray[j])
                    {
                        PhanSo tmp = new PhanSo();
                        tmp = PSArray[i];
                        PSArray[i] = PSArray[j];
                        PSArray[j] = tmp;
                    }
            }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 4

---

```
        Console.WriteLine("Mang cac phan so da sap xep tang dan:");
        foreach (var a in PSArray)
            Console.Write($"{a.Output()} ");
        Console.WriteLine();

        //Tim phan so co gia tri lon nhat
        PhanSo max = PSArray[0];
        foreach (var a in PSArray)
            if (a > max)
                max = a;
        Console.WriteLine($"Phan so lon nhat la {max.Output()}");

        Console.ReadKey();
    }
}
```