

Lập trình hướng đối tượng

Operator overloading

TS. Lê Văn Vinh

Khoa CNTT – Trường ĐH Sư phạm Kỹ thuật Tp. Hồ Chí Minh

1. Ví dụ operator overloading

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Chapter_5___Operator_overloading
{
    public class PhanSo
    {
        public int iTuSo { get; set; }
        public int iMauSo { get; set; }

        public PhanSo()
        {
        }
        public PhanSo(int tuso, int mauso)
        {
            this.iTuSo = tuso;
            this.iMauSo = mauso;
        }
        //Cộng hai phân số
        public static PhanSo operator +(PhanSo a, PhanSo b)
        {
            if (a.iMauSo == b.iMauSo)
            {
                return new PhanSo(a.iTuSo + b.iTuSo, a.iMauSo);
            }
            int tu1 = a.iTuSo * b.iMauSo;
            int tu2 = b.iTuSo * a.iMauSo;

            return new PhanSo(tu1 + tu2, a.iMauSo * b.iMauSo);
        }
        //Cộng phân số và một số nguyên
        public static PhanSo operator +(PhanSo a, int num)
        {
            return new PhanSo(a.iTuSo + a.iTuSo*num, a.iMauSo);
        }
    }
}
```

```

}
//Tăng giá trị lên 1
public static PhanSo operator ++(PhanSo a)
{
    a = a + 1;
    return a;
}

//So sánh lớn hơn
public static bool operator >(PhanSo a, PhanSo b)
{
    double value1 = (double)a.iTuSo / a.iMauSo;
    double value2 = (double)b.iTuSo / b.iMauSo;
    return value1 > value2;
}

//So sánh nhỏ hơn
public static bool operator <(PhanSo a, PhanSo b)
{
    double value1 = (double)a.iTuSo / a.iMauSo;
    double value2 = (double)b.iTuSo / b.iMauSo;
    return value1 < value2;
}

//So sánh bằng
public static bool operator ==(PhanSo a, PhanSo b)
{
    double value1 = (double)a.iTuSo / a.iMauSo;
    double value2 = (double)b.iTuSo / b.iMauSo;
    return value1 == value2;
}

//So sánh khác
public static bool operator !=(PhanSo a, PhanSo b)
{
    double value1 = (double)a.iTuSo / a.iMauSo;
    double value2 = (double)b.iTuSo / b.iMauSo;
    return value1 != value2;
}

//Ép số nguyên về kiểu phân số
public static implicit operator PhanSo(int num)
{
    PhanSo kq = new PhanSo(num, 1);
    return kq;
}

//Ép phân số về kiểu double
public static explicit operator double(PhanSo ps)
{
    double kq = (double)ps.iTuSo / ps.iMauSo;
    return kq;
}

public void Xuat()
{

```

```

        Console.WriteLine(this.iTuSo + "/" + this.iMauSo);
    }
}
class Program
{
    static void Main(string[] args)
    {
        PhanSo a = new PhanSo(2, 5);
        PhanSo b = new PhanSo(4, 5);
        PhanSo c = a + b;
        c.Xuat();

        c++;
        c.Xuat();

        if (a>b)
        {
            Console.WriteLine("a lon hon b");
        }
        else
        {
            Console.WriteLine("a nho hon hoac bang b");
        }

        if (a == b)
        {
            Console.WriteLine("a bang b");
        }
        else
        {
            Console.WriteLine("a khong bang b");
        }

        Console.WriteLine("Nhap vao mot so nguyen: ");
        int num = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Kieu phan so la: ");
        PhanSo d = num; //Ép kiểu không tường minh
        d.Xuat();

        double kq = (double)a; //Ép kiểu tường minh
        Console.WriteLine("Gia tri cua phan so la: " + kq);
    }
}

```

Bài tập:

Định nghĩa các lớp sau, và xây dựng toán tử so sánh bằng, so sánh khác cho các lớp.

1. Lớp điểm (DIEM)
2. Lớp ngày (NGAY (ngày, tháng, năm))
3. Lớp thời gian (THOIGIAN (giờ, phút, giây, NGÀY))
4. Lớp đoạn thẳng
5. Lớp đường tròn
6. Lớp hình chữ nhật

Định nghĩa các lớp sau, và xây dựng toán tử so sánh lớn hơn, nhỏ hơn, lớn hơn hoặc bằng, nhỏ hơn hoặc bằng, tăng dần, giảm dần, so sánh khác cho các lớp.

1. Lớp ngày (NGAY (ngày, tháng, năm))
2. Lớp thời gian (THOIGIAN (giờ, phút, giây, NGÀY))
3. Lớp đường tròn
4. Lớp hình chữ nhật
5. Lớp hình tam giác

Thêm hàm ép kiểu: NGÀY, THOIGIAN → String, và ngược lại