

## 第63章 演算子オーバーロード その2



今回は、[前章](#)で作ったMyPositionクラスにスカラー倍を定義します。スカラーとは大きさだけを持つ普通の数字です。

MyPositionクラスのオブジェクトAが(a,b)で表されるとき、 $A * n$ や $n * A$ (nはint型)を定義します。

```
A * n = (a * n, b * n)
n * A = (a * n, b * n)
```

とします。これは、簡単ですね。

```
// opover02.cs

using System;

class MyPosition
{
    int nX, nY;

    public int x
    {
        get
        {
            return nX;
        }
        set
        {
            nX = value;
        }
    }
    public int y
    {
        get
        {
            return nY;
        }
        set
        {
            nY = value;
        }
    }

    public static MyPosition operator +(MyPosition a, MyPosition b)
    {
        MyPosition c = new MyPosition();

        c.nX = a.nX + b.nX;
        c.nY = a.nY + b.nY;

        return c;
    }

    public static MyPosition operator -(MyPosition a, MyPosition b)
    {
        MyPosition c = new MyPosition();

        c.nX = a.nX - b.nX;
        c.nY = a.nY - b.nY;

        return c;
    }

    public static MyPosition operator *(MyPosition a, int b)
    {
        MyPosition c = new MyPosition();
        c.x = a.x * b;
        c.y = a.y * b;
        return c;
    }

    public static MyPosition operator *(int b, MyPosition a)
```

```

    {
        MyPosition c = new MyPosition();
        return a * b;
    }

    public MyPosition(int m, int n)
    {
        nX = m;
        nY = n;
    }

    public MyPosition()
    {
        nX = 0;
        nY = 0;
    }
}

class opover01
{
    public static void Main()
    {
        MyPosition A = new MyPosition(3, 5);
        MyPosition B = new MyPosition(4, 6);

        MyPosition C = A * 3;
        Console.WriteLine("A * 3 = ({0}, {1})", C.x, C.y);
        MyPosition D = 3 * A;
        Console.WriteLine("3 * A = ({0}, {1})", D.x, D.y);

        MyPosition E = (A + B) * 5;
        Console.WriteLine("(A + B) * 5 = ({0}, {1})", E.x, E.y);

        MyPosition F = A * 5 + B * 5;
        Console.WriteLine("A * 5 + B * 5 = ({0}, {1})", F.x, F.y);
    }
}

```

新たに、演算子のオーバーロードを行っています。

```
public static MyPosition operator *(MyPosition a, int b)
```

次に、引数の型の順番が逆になっている

```
public static MyPosition operator *(int b, MyPosition a)
```

も、定義しています。この時  $a * b$  も  $b * a$  も同じ結果になるので、

```

public static MyPosition operator *(int b, MyPosition a)
{
    MyPosition c = new MyPosition();
    return a * b;
}

```

と、直前の演算子オーバーロードを利用しています。

では、実行結果を見てみましょう。

