

演算子のオーバーロードしてみよう

そのままいきなりソースコード

```
#include <iostream>

//オペレータ 演算子 +-*/% sizeof () = * &
// int a = 1 + 2 二項演算子 1とか2のことをオペランド (右オペランド、左オペランド)
// int *b = &a; オペランド 1個のやつ=単項演算子

using std::endl;
using std::cout;

class Vec2
{
public:
    int x;
    int y;
    Vec2(int _x, int _y):x(_x),y(_y){}
    Vec2() {}
    ~Vec2() {};
    void PrintVal() { cout << "(x, y) = (" << x << ", " << y << ")" << endl; }
};

//
//Vec2 PlusVec2AndVec2(const Vec2 &_v1, const Vec2 &_v2);
//
//Vec2 PlusVec2AndVec2(const Vec2 &_v1, const Vec2 &_v2)
//{
//    return(Vec2(_v1.x + _v2.x, _v1.y + _v2.y));
//}

//プロトタイプ宣言
Vec2& operator+(const Vec2& _v1, const Vec2& _v2);

//定義
Vec2& operator+(const Vec2& _v1, const Vec2& _v2) {
    Vec2 ret;

    ret = Vec2(_v1.x + _v2.x, _v1.y + _v2.y);

    return (ret);
}

int main()
{
    Vec2 pos1 = { 10,10 };
    pos1.PrintVal();
    Vec2 pos2(20, 30);
    pos2.PrintVal();

    //int a = 10;
    //int b = 20;
    //int c = a + b;
    //cout << c << endl;
    //res.x = pos1.x + pos2.x;
    //res.y = pos1.y + pos2.y;

    Vec2 res = pos1 + pos2;
```

```
//Vec2 res;  
//res = PlusVec2AndVec2(pos1, pos2);  
res.PrintVal();  
}
```

🕒Revision #2

★Created 9 May 2025 09:13:14 by youe2

✎Updated 2 June 2026 17:30:04 by youe2