重载算术操作符

- **不解析，看代码**

```
class add
{
public:
    int i;
    int j;

    add(int i, int j)
    {
        this->i = i;
        this->j = j;
    }

    add operator + (const add &other)   /* operator + 重载算术运算符---------成员操作符 */
    {
        add m1;
        m1.i = i + other.i;
        m1.j = j + other.j;

        return m1;
    }
}

 add operator + (const add &a, const add &b)   /* 全局操作符， */
{
    add m1;                                 /* 注意，操作类的成员，需要声明为朋友（friend） */
    m1.i = a.i + b.i;
    m1.j = a.j + b.j;

    return m1;
}

int main ()
{
    add a(1, 2);
    add b(2, 3);

    add c = a + b;

    return 0;
}
```

- **子类中未实现赋值重载时，会调用父类的赋值重载（无论实现或未实现）**
- **子类一旦实现赋值重载，不会主动调用父类的赋值重载**

- **子类中会把父类重名的成员**shadow **(** father :: dis()**携带命名空间)**