返回结果而不产生副作用的操作称为函数

let arr = [1, 2, 3, 4]

// filter 是无副作用函数

let newArr = arr.filter(e=>e > 2);

// push 不是无副作用函数

arr.push(5);

如果一个操作既有读又有写，那么应该重构为两个函数，读写分离

注：从上述2条规则我们可以看出，无副作用函数即为读函数

值对象的操作均为函数，应为值对象不能改变，想改变一个值对象，只有替换，如果值对象想改变自身属性，则应该在函数中返回一个新的值对象，然后调用者替换掉这个值对象，基于上述方法，我们可以将一些复杂的概念封装为值对象，如下Color便是一个值对象

// 将油桶颜色混合

paint.Color = Color.MixedWith(otherPaint.Color)