rules.sonarsource.com

Swift static code analysis: Precedence and associativity of standard operators should not be changed

2 minutes

It is acceptable to override standard operators to provide appropriate behaviors for your classes. But it is not appropriate to change those operators' associativity or precedence from the standard. Doing so will inevitably lead to misuse and mistakes for users of the class.

Instead of overriding an existing operator’s associativity or precedence, you should either let them use the default values or define a completely new operator.

Noncompliant Code Example

```
infix operator - : CustomAdditionPrecedence   // Noncompliant. For a different behavior create a different operator

precedencegroup CustomAdditionPrecedence {
  associativity: right
}

func - (lhs: MyInt, rhs: MyInt) -> MyInt {
  // ...
}

var a = MyInt(10), b = MyInt(5), c = MyInt(5)
print(a - b - c) // against expectations, this outputs 10
```

Compliant Solution

```
infix operator <- : CustomAdditionPrecedence

precedencegroup CustomAdditionPrecedence {
  associativity: right
}

func <- (lhs: MyInt, rhs: MyInt) -> MyInt {
  // ...
}

var a = MyInt(10), b = MyInt(5), c = MyInt(5)

var a = MyInt(10), b = MyInt(5), c = MyInt(5)
print(a - b - c) // prints 0 as expected
print(a <- b <- c) // prints 10
```