CodeQL常见语法解释

基础词汇

词汇表	
短语	**意义**
注释 annotations 注解	用于修改声明的附加说明符，如private、override、deprecated、pragma、bindingset 或 cached.
body	{}、()或 if-then-else 或 from-where select 的每个部分中的文本.
二元运算符 binary operator	有两个操作数的运算符，如比较运算 符、and、or、IMPLICES 或算术运算 符.
调用 call	调用谓词的公式，例如.这是静态的()或呼叫(a、b).

结合 conjunct	是 and 的操作数的公式.
申明 declaration	类、模块、谓词、字段或新类型.
间断 disjunct	作为或的操作数的公式.
公式 formula	一种逻辑表达式，如 A=B，一个调用， 一个量词，and，or，not，in 或instanceof.

短语	**意义**
应该/不应该/避免/喜欢should/should not/avoid/prefer	只要有可能，只要有道理，就坚持这条规则.
may/can	这是一个合理的选择，要慎重使用.
必须/总是/不 must/always/do not	一定要遵守这条规则.
量词/聚合 quantifier/aggregation	exists，count，strictcount，any，forall，forex 等等.
变量 variable	谓词、字段、源变量或由量词或聚合引 入的变量的参数.

常见的使用语法

1.bindingset[i]，如果是类似这样的注解在一个函数上的时候，表示这个i是一个有限的集合，可能是一个数组。 如果没有这个注解，意思就是应用于全部的变量
2.Method是一个函数的原型，即public sink() ，MethodAccess是一个函数的调用，即sink()。
3.Construtor是一个构造方法的原型，ConstrutorCall是一个构造方法的调用。
4.dataflow::Node是一个基本节点， 你要获得这个节点的具体的代码的化需要Dataflow::Node node ，node.asExpr() 来访问。 这相当于ExprNode
5.this就是常见的this的意思
6.result是一个特殊的关键词 表示这个谓词会返回一个结果，这个结果就是result赋值的结果
7.强制类型转换 .(ConstructorCall)
8.获得一个方法调用的原型定义可以使用getSourceDeclaration
9.exits()一般是 这样用的，exits(int x | x > 6 and x < 10 and #some other expressions#)
10.hasQualifiedName("java.lang","ProcessBuilder") 限定包名
11.abstract方法可以用于扩展更多的可能性
例如：

```
abstract class SqlExpr extends Expr { ...
}
```

12.Expr模块是表达式模块，表达式可能是赋值，调用等情况。
13.stmt是语法模块，你可以使用xxx.getBasicBlock()获得某个表达式所在的作用域，然后你可以获得这个作用域的所对应的分支，有助于你更好的检查数据流
14.数组[0..10]
15.getSuccessor*()表示 递归调用这个方法，*是正则表达式那样的意思，当然还有一个getSuccessor+()这样的方法 也是一样的 匹配最少1次的递归
16.先睡觉了 这里暂停下 下次写