用于Java程序的抽象语法树类（Abstract syntax tree classes for working with Java programs）

CodeQL有很多类来表示Java程序的抽象语法树。
抽象语法树（AST）表示程序的语法结构。 AST上的节点代表诸如语句和表达式之类的元素。

方法目录表（Statement classes）

下表列出了Stmt的所有子类。

Statement syntax	CodeQL class	Superclasses	Remarks
`;`	EmptyStmt		
Expr `;`	ExprStmt		
`{` Stmt `...` `}`	BlockStmt		
`if (` Expr `)` Stmt `else` Stmt	IfStmt	ConditionalStmt	
`if (` Expr `)` Stmt			
`while (` Expr `)` Stmt	WhileStmt	ConditionalStmt, LoopStmt	
`do` Stmt `while (` Expr `)`	DoStmt	ConditionalStmt, LoopStmt	
`for (` Expr `;` Expr `;` Expr `)` Stmt	ForStmt	ConditionalStmt, LoopStmt	

58安全应急响应中心

Statement syntax	CodeQL class	Superclasses	Remarks
`for (` VarAccess `:` Expr `)` Stmt	EnhancedForStmt	LoopStmt	
`switch (` Expr `) {` SwitchCase `... }`	SwitchStmt		
`try {` Stmt `... } finally {` Stmt `... }`	TryStmt		
`return` Expr `;`	ReturnStmt		
`return ;`			
`throw` Expr `;`	ThrowStmt		
`break ;`	BreakStmt	JumpStmt	
`break label ;`			
`continue ;`	ContinueStmt	JumpStmt	
`continue label ;`			
`label :` Stmt	LabeledStmt		
`synchronized (` Expr `)` Stmt	SynchronizedStmt		
`assert` Expr `:` Expr `;`	AssertStmt		
`assert` Expr `;`			
TypeAccess `name ;`	LocalVariableDeclStmt		
`class name {` Member `... } ;`	LocalClassDeclStmt		
`this (` Expr `, ... ) ;`	ThisConstructorInvocationStmt		
`super (` Expr `, ... ) ;`	SuperConstructorInvocationStmt		
`catch (` TypeAccess `name ) {` Stmt `... }`	CatchClause		can only occur as child of a TryStmt

58安全应急响应中心

Statement syntax	CodeQL class	Superclasses	Remarks
`case` Literal `:` Stmt `...`	ConstCase		can only occur as child of a SwitchStmt
`default :` Stmt `...`	DefaultCase		can only occur as child of a SwitchStmt

表达式类（Expression classes）

表达式类很多，因此我们按类别显示它们。 本节中的所有类都是Expr的子类。

常量（literal）

本小节中的所有类都是Literal的子类。

Expression syntax example	CodeQL class
`true`	BooleanLiteral
`23`	IntegerLiteral
`23l`	LongLiteral
`4.2f`	FloatingPointLiteral
`4.2`	DoubleLiteral
`'a'`	CharacterLiteral
`"Hello"`	StringLiteral
`null`	NullLiteral

一元表达式（Unary expressions）

本小节中的所有类都是UnaryExpr的子类。

Expression syntax	CodeQL class	Superclasses	Remarks
Expr `++`	PostIncExpr	UnaryAssignExpr	
Expr `--`	PostDecExpr	UnaryAssignExpr	

58安全应急响应中心

Expression syntax	CodeQL class	Superclasses	Remarks
`++` Expr	PreIncExpr	UnaryAssignExpr	
`--` Expr	PreDecExpr	UnaryAssignExpr	
`~` Expr	BitNotExpr	BitwiseExpr	see below for other subclasses of BitwiseExpr
`-` Expr	MinusExpr		
`+` Expr	PlusExpr		
`!` Expr	LogNotExpr	LogicExpr	see below for other subclasses of LogicExpr

二元表达式（Binary expressions）

本小节中的所有类都是BinaryExpr的子类。

Expression syntax	CodeQL class	Superclasses
Expr `*` Expr	MulExpr	
Expr `/` Expr	DivExpr	
Expr `%` Expr	RemExpr	
Expr `+` Expr	AddExpr	
Expr `-` Expr	SubExpr	
Expr `<<` Expr	LShiftExpr	
Expr `>>` Expr	RShiftExpr	
Expr `>>>` Expr	URShiftExpr	
Expr `&&` Expr	AndLogicalExpr	LogicExpr
Expr `||` Expr	OrLogicalExpr	LogicExpr
Expr `<` Expr	LTExpr	ComparisonExpr
Expr `>` Expr	GTExpr	ComparisonExpr
Expr `<=` Expr	LEExpr	ComparisonExpr
Expr `>=` Expr	GEExpr	ComparisonExpr

58安全应急响应中心

Expression syntax	CodeQL class	Superclasses
Expr `==` Expr	EQExpr	EqualityTest
Expr `!=` Expr	NEExpr	EqualityTest
Expr `&` Expr	AndBitwiseExpr	BitwiseExpr
Expr `|` Expr	OrBitwiseExpr	BitwiseExpr
Expr `^` Expr	XorBitwiseExpr	BitwiseExpr

赋值表达式（Assignment expressions）

该表中的所有类都是Assignment的子类。

Expression syntax	CodeQL class	Superclasses
Expr `=` Expr	AssignExpr	
Expr `+=` Expr	AssignAddExpr	AssignOp
Expr `-=` Expr	AssignSubExpr	AssignOp
Expr `*=` Expr	AssignMulExpr	AssignOp
Expr `/=` Expr	AssignDivExpr	AssignOp
Expr `%=` Expr	AssignRemExpr	AssignOp
Expr `&=` Expr	AssignAndExpr	AssignOp
Expr `|=` Expr	AssignOrExpr	AssignOp
Expr `^=` Expr	AssignXorExpr	AssignOp
Expr `<<=` Expr	AssignLShiftExpr	AssignOp
Expr `>>=` Expr	AssignRShiftExpr	AssignOp
Expr `>>>=` Expr	AssignURShiftExpr	AssignOp

访问入口（Accesses）

Expression syntax examples	CodeQL class
`this`	ThisAccess
`Outer.this`	

58安全应急响应中心

Expression syntax examples	CodeQL class
`super`	SuperAccess
`Outer.super`	
`x`	VarAccess
`e.f`	
`a[i]`	ArrayAccess
`f(...)`	MethodAccess
`e.m(...)`	
`String`	TypeAccess
`java.lang.String`	
`? extends Number`	WildcardTypeAccess
`? super Double`	

其他（Miscellaneous）

Expression syntax examples	CodeQL class	Remarks
`(int) f`	CastExpr	
`(23 + 42)`	ParExpr	
`o instanceof String`	InstanceOfExpr	
Expr `?` Expr `:` Expr	ConditionalExpr	
`String.class`	TypeLiteral	
`new A()`	ClassInstanceExpr	
`new String[3][2]`	ArrayCreationExpr	
`new int[] { 23, 42 }`		
`{ 23, 42 }`	ArrayInit	can only appear as an initializer or as a child of an ArrayCreationExpr

58安全应急响应中心

Expression syntax examples	CodeQL class	Remarks
`@Annot(key=val)`	Annotation	

58安全应急响应中心