Program	::= **[** ε | (DeclarationList | Statement)* **]**
DeclarationList	::= Declaration *
Declaration	::= VarDeclaration | OprDeclaration | ε
VarDeclaration	::= **Identifier:** (Type (ε | **=** Literal) | (**col <**Type**>** ((**{** NumberLiteral | **Identifier }**) | **=** CollectionLiteral)) **;**
OprDeclaration	::= Type **opr Identifier (** Parameter ((**,** Parameter)*|ε) **)** Block (**send** (Literal | **Identifier**) | ε) **;**
Block	::= **[** ε | (VarDeclaration | Statement)* **]**
Parameter	::= (**Identifier :** Type) | ε
Statement	::= singleStatement (**;** singleStatement)*
singleStatement	::=Identifier (**=** Expression | **(** Expression **));** | **? (** Expression **)** Block | **until (** Expression **)** Block | **out <=** Expression **;** | **in =>** Expression **;** | Block
Expression	::= PrimaryExpression (**Operator** PrimaryExpression)*
PrimaryExpression	::= **Identifier** (ε | **(** ExpressionList **)**) | **Operator** PrimaryExpression | Literal | **(** Expression **)**
Literal	::= **NumberLiteral** | **LetterLiteral** | **StateLiteral** | CollectionLiteral
CollectionLiteral	::= **<** (**Identifier |** Literal) (ε | (**,** (**Identifier |** Literal)*) **>**
ExpressionList	::= Expression (**,** Expression) * | ε
Type	::= **number** | **letter** | **state** | **col** | **void**

Tokens	
Identifier	**Letter** (**Letter** | **Digit**)*
NumberLiteral	**Digit** (**Digit**)*
LetterLiteral **StateLiteral** **Digit** **Letter**	**Letter** ::= **true** | **false** ::= **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** := **a** | **b** | … | **z** | **A** | **B** | … |**Z**
Operator	::= **+** | **-** | ***** | **/** | **>** | **<** | **=** | **==** | **!** | **&** | **|** | **--** | **<<** | **>>** | **//** | **`**
opr	
send	
Until	
Out	
In	
?	
(	
)	
[	
]	
{	
}	
<	
>	
;	
:	
,	