**TestLang grammar**

Start symbol: Program

|  |  |
| --- | --- |
| Program | ::= DeclarationList |
| DeclarationList | ::= DeclarationList declaration  | Declaration |
| Declaration | ::= VarDeclaration  | OprDeclaration  | ε |
| VarDeclaration | ::= **Identifier: Type** **;**  | **Identifier: Type** = Literal**;**  | CollectionDeclaration |
| CollectionDeclaration | ::= **Identifier:** col <Type> { NumberLiteral };  | **Identifier:** col <Type> { **Identifier** };  | **Identifier:** col <Type> = CollectionLiteral; |
| OprDeclaration | ::= **opr** **Identifier** **(** IdList **)** Block **send** Expression **;** |
| Type | ::= number  | letter  | sate  |col |
| Block | ::= [ VarDeclarations StatementsList ] |
| IdList | ::= **Identifier** IdListTail  | Literal IdListTail  | ε |
| IdListTail | ::= **,** **Identifier** IdListTail  |, Literal IdListTail  | ε |
| StatementsList | ::= Statement  | StatementsList Statement |
| Statement | ::= Expression **;**  | ? ( Expression ) **Block**  | **until** ( Expression ) **Block**  | **out** <= Expression **;**  | **in** => Expression **;**  | ε |
| Expression | ::= Primary  | Expression **Operator** Primary |
| Primary | ::= **Identifier**  | **Identifier** **(** ExpressionList **)**  | **Operator** Primary  | **Literal**  | **(** Expression **)** |
| Literal | ::= NumberLiteral  | LetterLiteral  | StateLiteral  | CollectionLiteral |
| CollectionLiteral | ::= < IdList> |
| ExpressionList | ::= Expression ExpressionListTail  | ε |
| ExpressionListTail | ::= **,** Expression ExpressionListTail  | ε |
| Operator | ::= +  | -  | \*  | /  | >  | <  | =  | ==  | !  | &  | |  | --  | <<  | >>  | //  | ‘ |