Grammer

+ startToken:Terminal
- productions:List<Production>

Production

+ index:Token
+ target: Vector<Token>

Parser

- grammerStack:Stack<Nonterminal>
- parsingTable:HashMap<Token, Production>

+ parse(Token t) : Production throw SyntaxErrorException
- reserve(Production p): void
- reserve(String productions): void

Lexer

- wordsMap:HashMap<String,Token>

+ scan():Token
+ Lexer():Lexer

Token

+ type : final String

+ Token(String type) : Token

Terminal

Nonterminal

- production:Production

Ope

+ name:String

NUM

+ value:int

Word

+ lexeme:String

Del

+ name:String