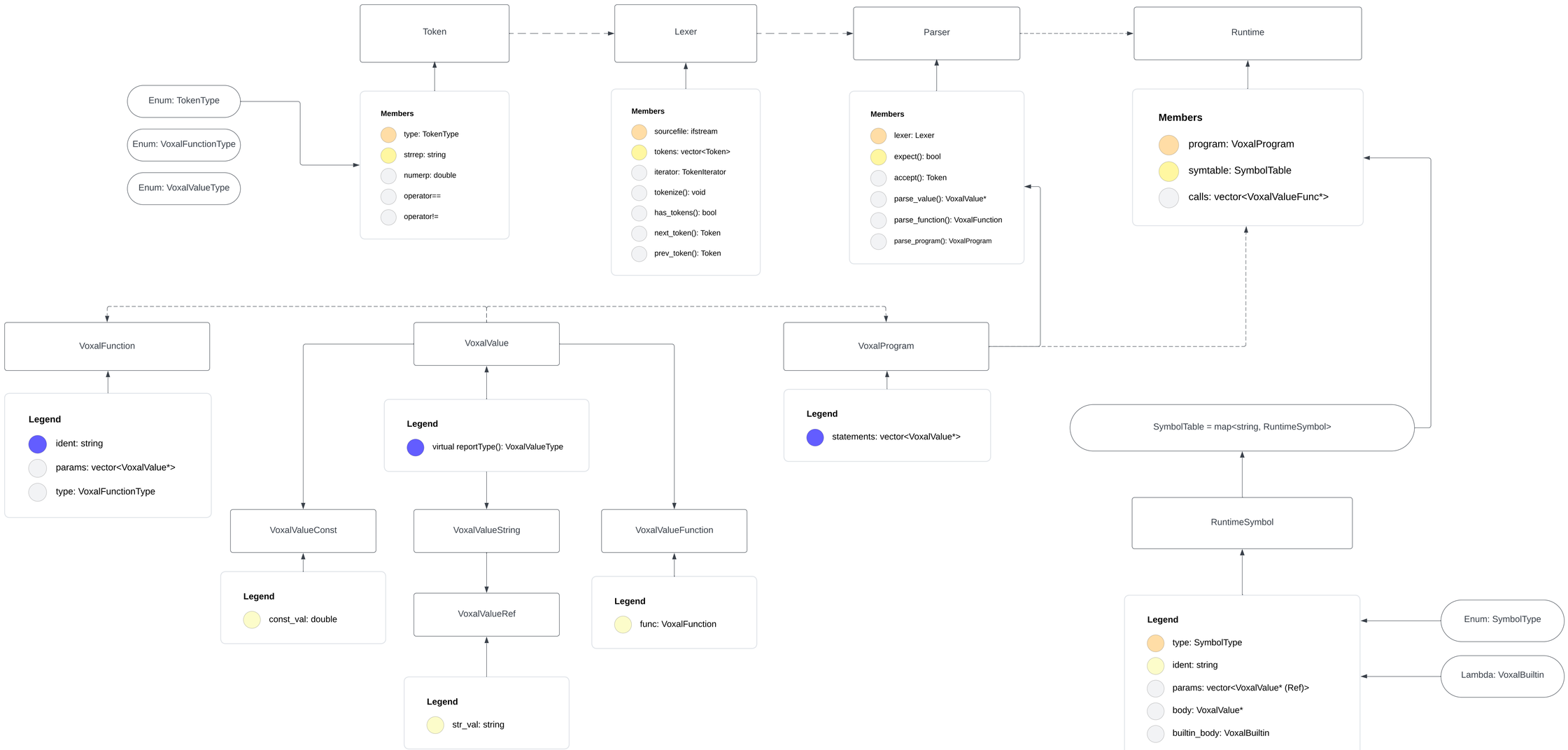
Token
Lexer
Parser
Runtime
Enum: TokenType
Enum: VoxalFunctionType
Enum: VoxalValueType
Members
type: TokenType
strrep: string
numerp: double
operator==
operator!=
Members
sourcefile: ifstream
tokens: vector<Token>
iterator: TokenIterator
tokenize(): void
has_tokens(): bool
next_token(): Token
prev_token(): Token
Members
lexer: Lexer
expect(): bool
accept(): Token
parse_value(): VoxalValue*
parse_function(): VoxalFunction
parse_program(): VoxalProgram
Members
program: VoxalProgram
symtable: SymbolTable
calls: vector<VoxalValueFunc*>
VoxalFunction
VoxalValue
VoxalProgram
Legend
ident: string
params: vector<VoxalValue*>
type: VoxalFunctionType
Legend
virtual reportType(): VoxalValueType
Legend
statements: vector<VoxalValue*>
SymbolTable = map<string, RuntimeSymbol>
VoxalValueConst
VoxalValueString
VoxalValueFunction
RuntimeSymbol
Legend
const_val: double
VoxalValueRef
Legend
func: VoxalFunction
Legend
str_val: string
Legend
type: SymbolType
ident: string
params: vector<VoxalValue* (Ref)>
body: VoxalValue*
builtin_body: VoxalBuiltin
Enum: SymbolType
Lambda: VoxalBuiltin