D lang Interpreter

Gleb Popov, Timur Usmanov
Compiler Construction
Innopolis University
Fall 2025

Project Context

Dynamic Lang

object types are not specified and can change while program execution

the language assumes **interpretation**

C++ Language

the implementation language is C++

it provides extensive memory management and optimization features

Personal parser

hand-written parser

if you want a thing done well, do it yourself :)

Recall: Lexer

```
var x := 5
print x
```

```
tkVar tkIdent(”x”) tkAssign tkIntLiteral(5) tkNewLine
tkPrint tkIdent tkNewLine
```

```
var t := {x:=1}
t := t + {y:=2}
```

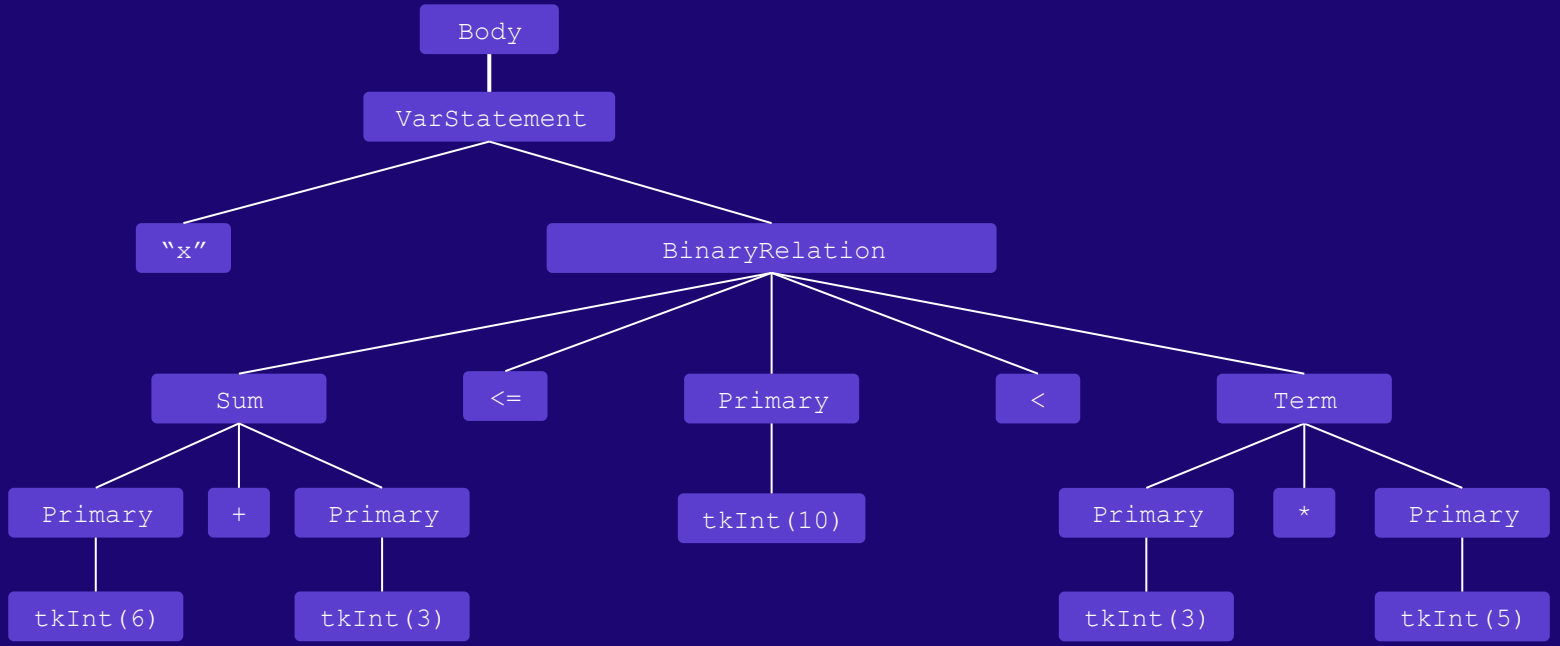
```
tkVar tkIdent(t) tkAssign tkOpenCurlyBrace tkIdent(“x”)
tkAssign tkIntLiteral(1) tkClosedCurlyBrace tkNewLine
tkIdent(“t”) tkAssign tkIdent(“t”) tkPlus tkOpenCurlyBrace
tkIdent(“y”) tkAssign tkIntLiteral(2) tkClosedCurlyBrace
```

```
var x := 3
if x < 10 then
    print "small"
else
    print "big"
end
```

```
tkVar tkIdent(“x”) tkAssign tkIntLiteral(3) tkNewLine tkIf
tkIdent(“x”) tkLess tkIntLiteral(10) tkThen tkNewLine
tkPrint tkStringLiteral(“small”) tkNewLine tkElse tkNewLine
tkPrint tkStringLiteral(“big”) tkNewLine tkEnd tkNewLine
```

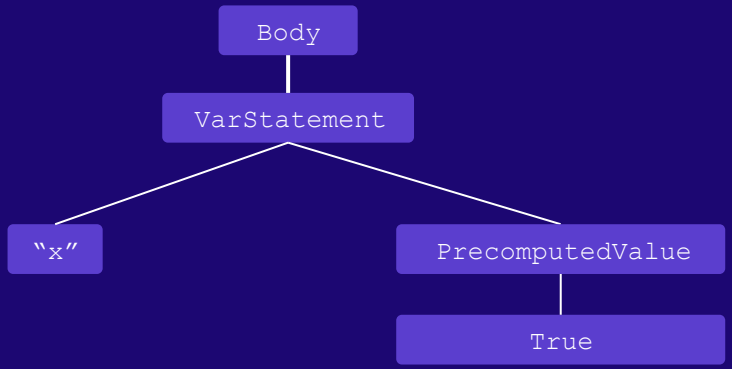
Recall: Syntax Analyzer

```
var x := 6 + 3 <= 10 < 3 * 5
```

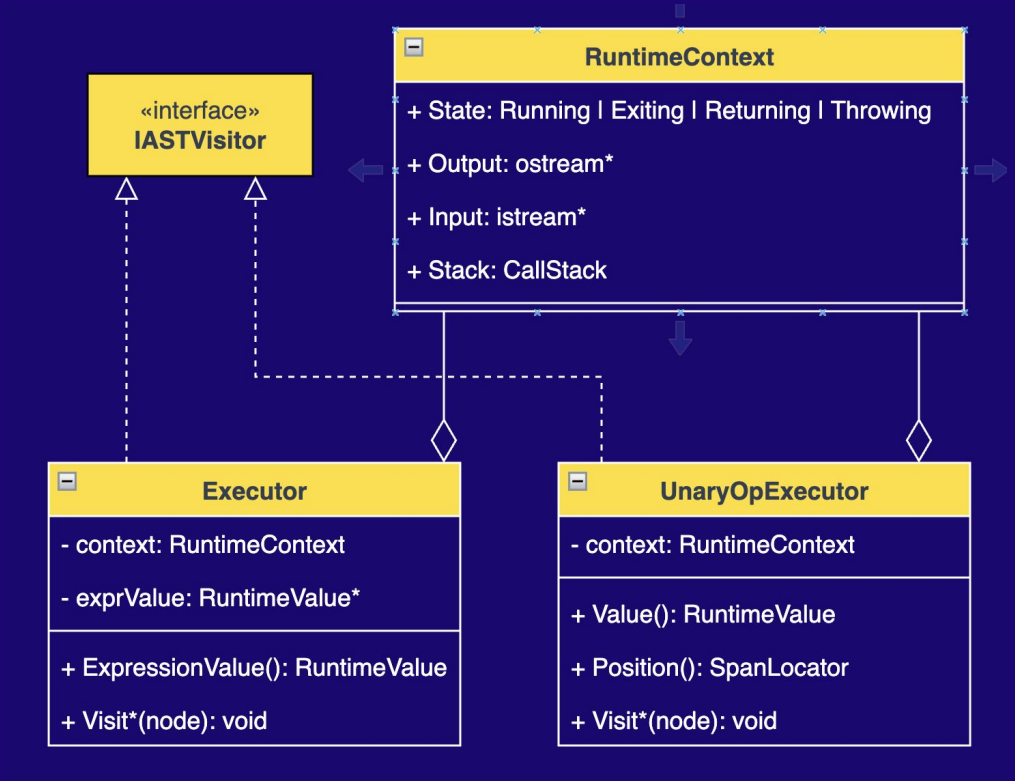


Recall: Semantic Analyzer

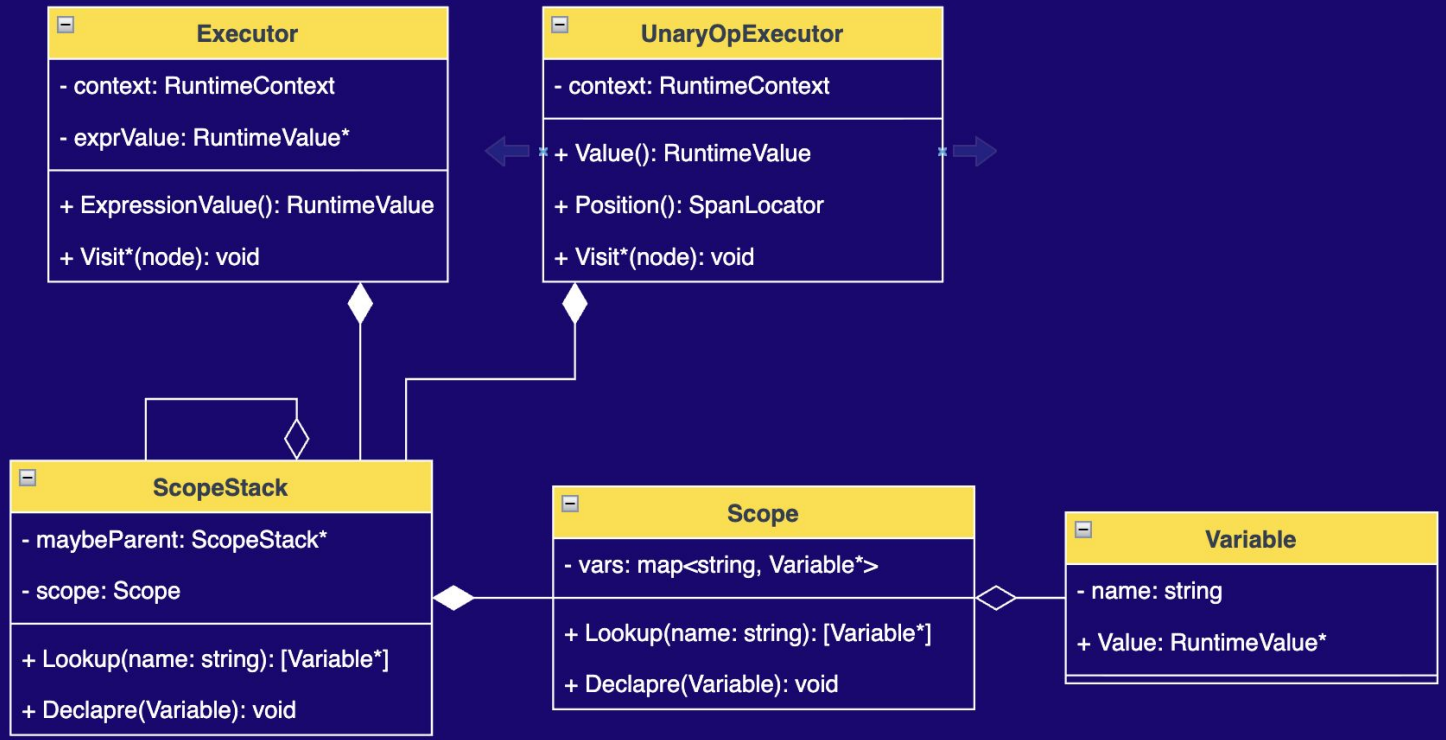
```
var x := 6 + 3 <= 10 < 3 * 5
```



Interpreter: Implementation Details



Interpreter: Implementation Details



Interpreter: Features

- everything from project description
- the input function
- BigInt custom implementation (Karatsuba algorithm)
- reference counting
- informative error reports (error location, stack trace)
- . . .

Interpreter: Features

```
dinterp - an interpreter for the D language.

Usage: dinterp [OPTIONS] [--] [file1.d file2.d ...]

Options:
    --help       -h  Show this text.
    --check      -c  Only check for errors, do not run.
    --examples       Show some usage examples.
    --lexer      -l  Stop after lexical analysis, output the tokens.
    --syntaxer   -s  Stop after syntactic analysis, start interactive AST traversal.
    --semantics  -S  Stop after semantic analysis, start interactive AST traversal.
    --nocontext  -C  Do not show code excerpts below errors.
    --notrace    -T  Do not show the call stack traceback on error.
Parameter options:
    --callstack <nonnegative integer>  Set the call stack capacity (default = 1024).
    --tracelen  <nonnegative integer>  On error, output at most this many call stack entries
(default = 50).

Every argument after -- is assumed to be a file name.
```