Programmierung eines LISP-Interpreters

threez

IT-Systemhaus inovex GmbH

- Software Entwicklung, Business Integration, Systems Engineering
- Java, Ruby & .NET
- Linux, Windows, Mac OS

http://www.inovex.de/

Wir nutzen Technologien, um unsere Kunden glücklich zu machen.
Und uns selbst.

Vorgehen

Lexikalische Analyse

Syntaktische Analyse

Ausführung

LISP?

```
(+ (* 10 4) 2)
```

$$10 * 4 + 2 = 42$$

S-Expressions

List

Atom

(+ (* 10 4) 3)

Atom

List

Token

begin

number

(+ (* 10 4) 3)

atom

end

Token Stream

(	[:begin]
+	[:atom, :+]
(	[:begin]
*	[:atom, :*]
10	[:number, 10.0]
4	[:number, 4.0]
)	[:end]
2	[:number, 3.0]
)	[:end]

Der Lexer

DEMO

Der AST

(Abstract Syntax Tree)

Array

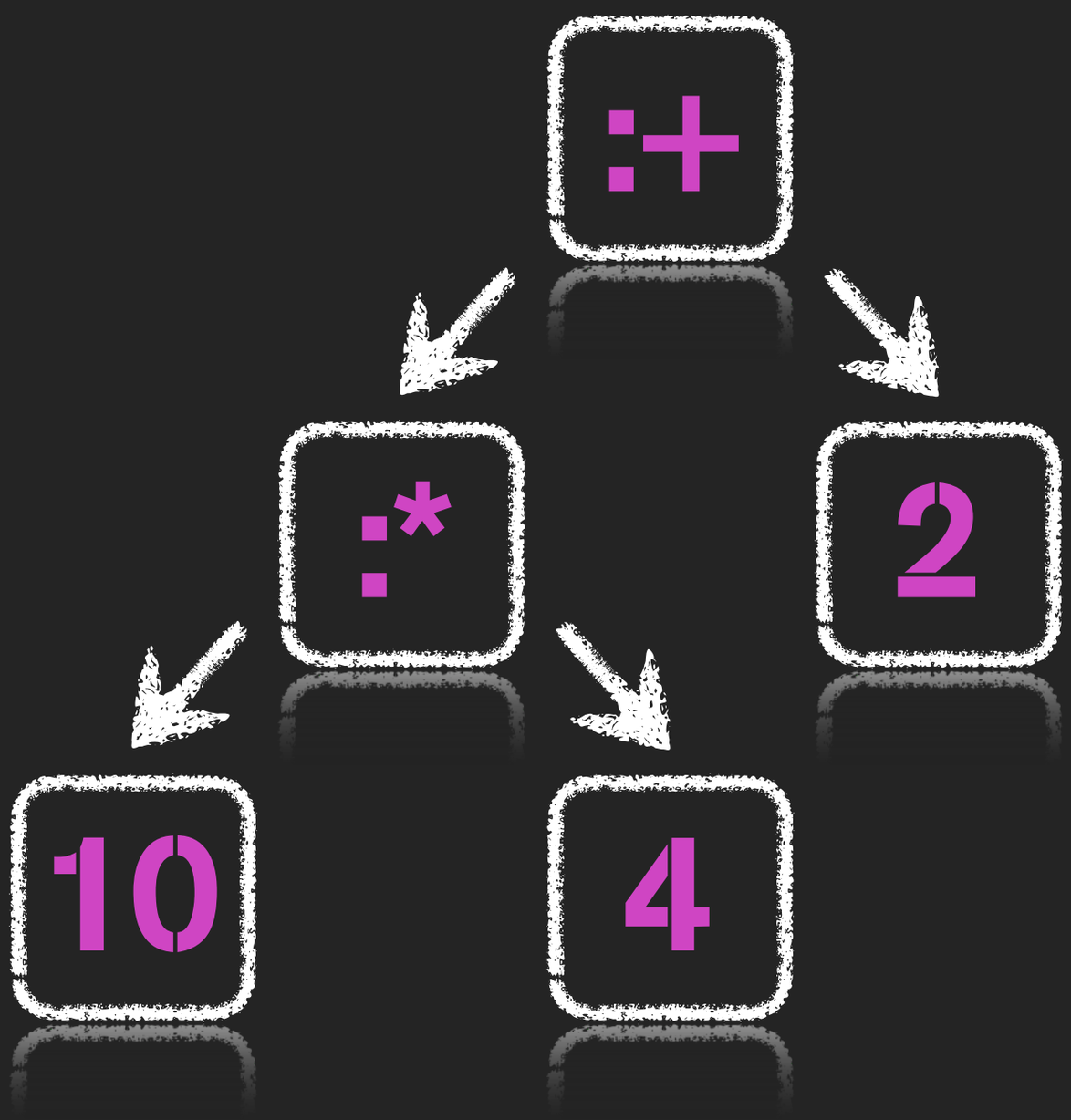
Symbol

[:+, [:*, 10, 4], 2]

Number

Der AST

(Abstract Syntax Tree)

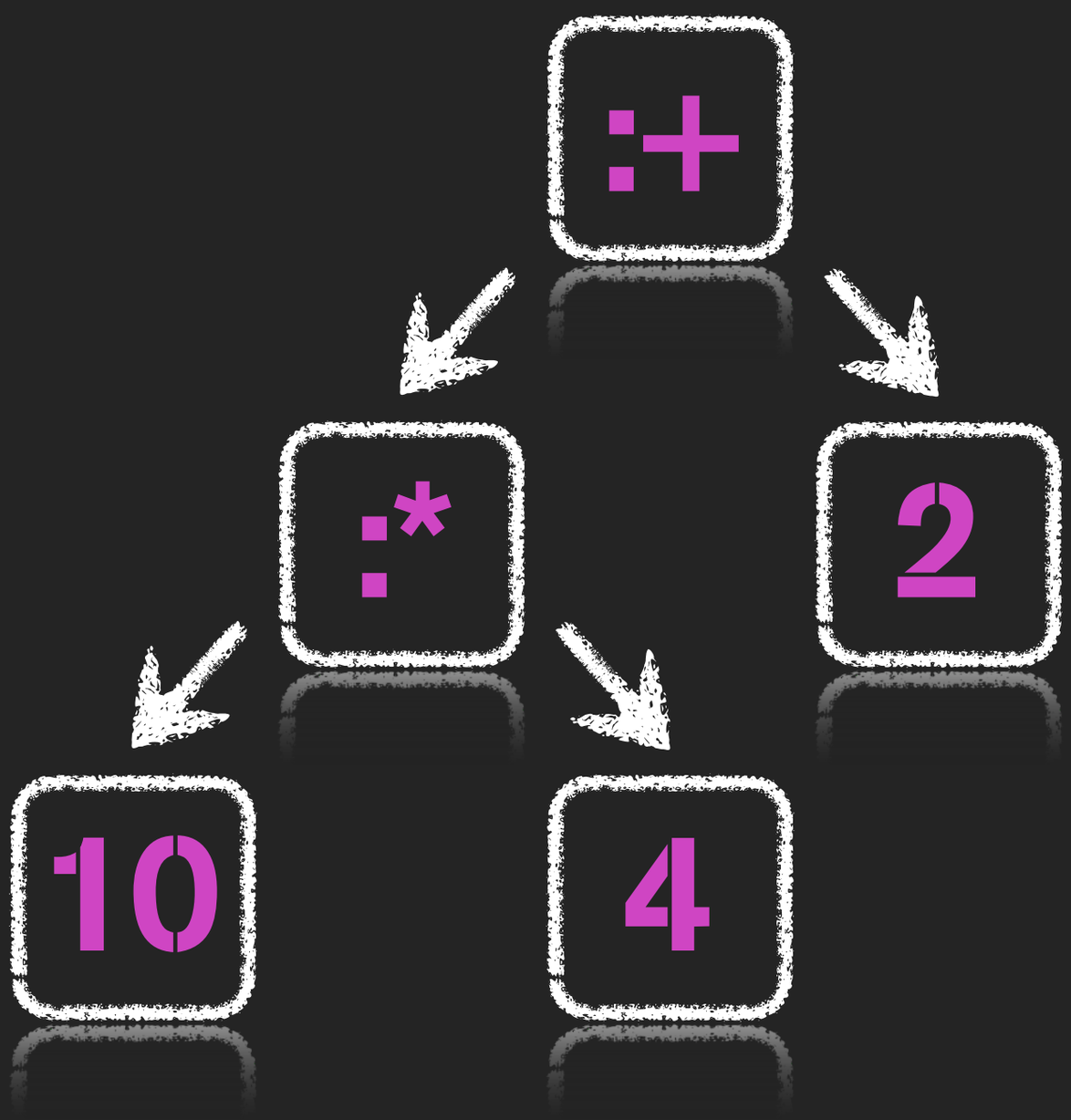


Der Parser

DEMO

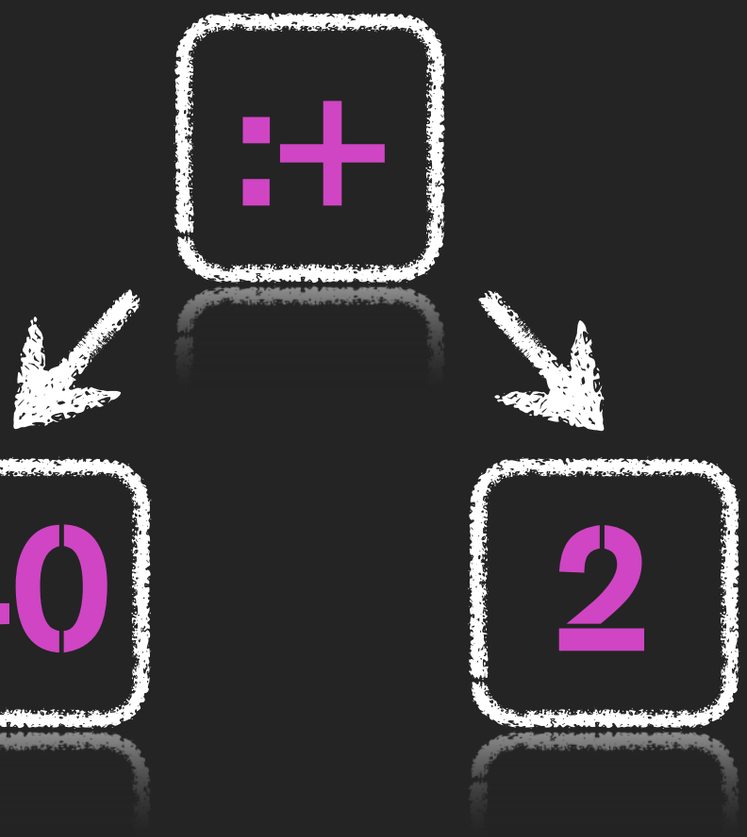
Der AST

(Abstract Syntax Tree)



Der AST

(Abstract Syntax Tree)



Der AST

(Abstract Syntax Tree)

eval & apply Schleife

eval  **liste rekursiv evaluieren &**

apply aufrufen

apply **funktion aufrufen**

Der Interpreter

DEMO