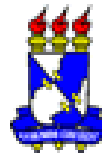


Framework para uma Família de LP

Prof. Alberto Costa Neto
alberto@ufs.br

Linguagens de Programação



Departamento de Computação
Universidade Federal de Sergipe

Linguagens

- Linguagem de Expressões 1 (LE1)
- Linguagem de Expressões 2 (LE2)
- Linguagem Funcional 1 (LF1)
- Linguagem Funcional 2 (LF2)
- Linguagem Funcional 3 (LF3)
- Linguagem Imperativa 1 (LI1)
- Linguagem Imperativa 2 (LI2)
- Linguagem Orientada a Objetos 1 (LO1)



LE1

Linguagem de Expressões 1

(Expressões constantes)



LE1 - Características

- Uma **expressão** é uma construção que é avaliada para produzir um **valor**
- Um **programa** é uma **expressão**
 - Sua execução (avaliação) produz um valor
- Envolve conceitos de **tipo** (inteiro, booleano e string), **valores constantes** e **operadores** (unários e binários) sobre expressões



LE1 - Sintaxe

Programa ::= Expressão

Expressao ::= Valor | ExpUnaria | ExpBinaria

Valor ::= ValorConcreto

ValorConcreto ::= ValorInteiro | ValorBooleano | ValorString

ExpUnaria ::= “-” Expressao | “not” Expressao
| “length” Expressao

ExpBinaria ::= Expressao “+” Expressao
| Expressao “-” Expressao
| Expressao “and” Expressao
| Expressao “or” Expressao
| Expressao “==” Expressao
| Expression “++” Expression



LE1 - Semântica

- Não foi definida uma semântica formal
- A implementação do interpretador serve como uma semântica operacional (em Java)
 - Java é usada como uma meta-linguagem



LE1 - Exemplos

- 2 + 5 - 10

"expressions" ++ " " ++ "1"

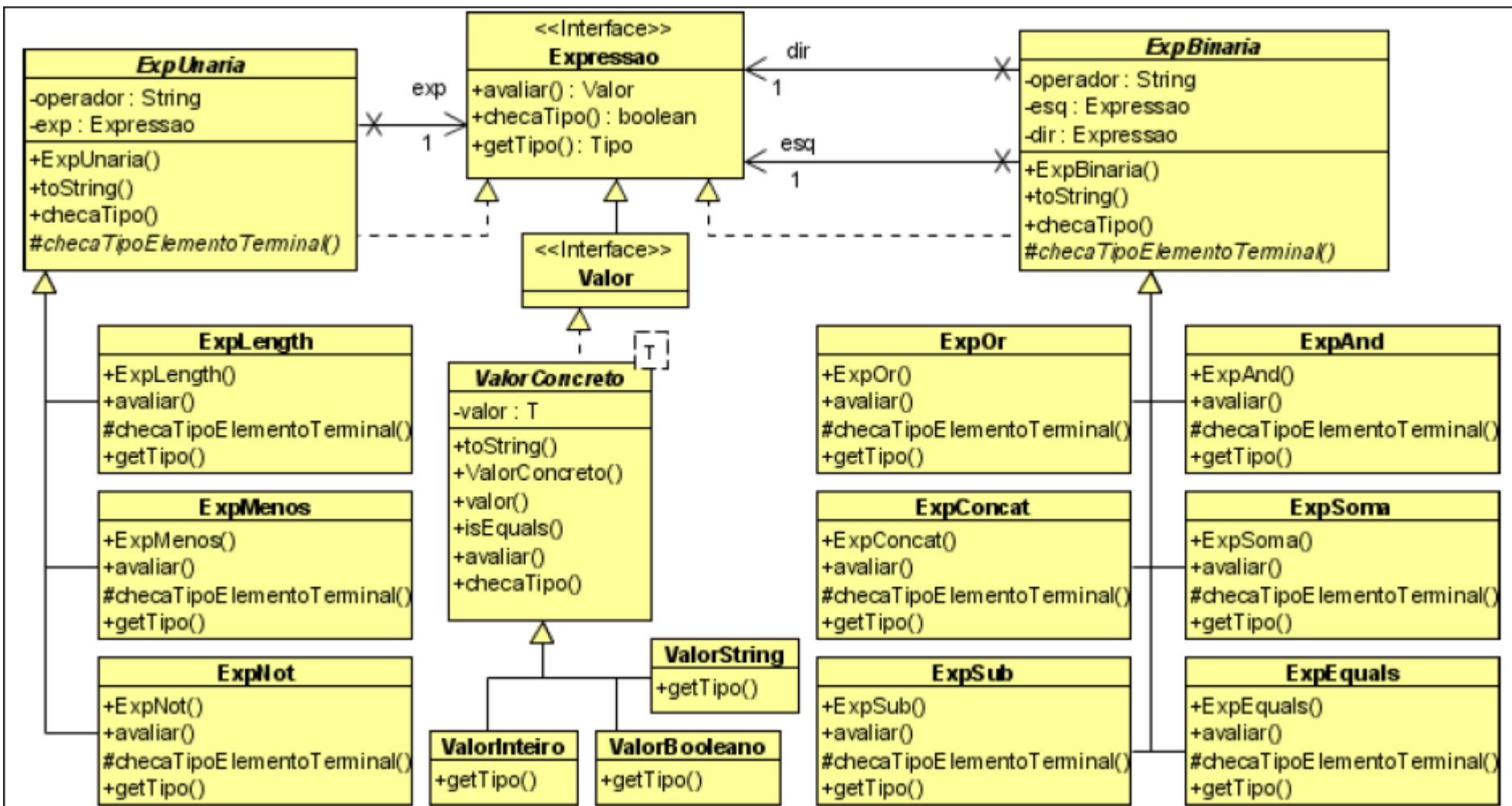
length "abc"

false or true and not false

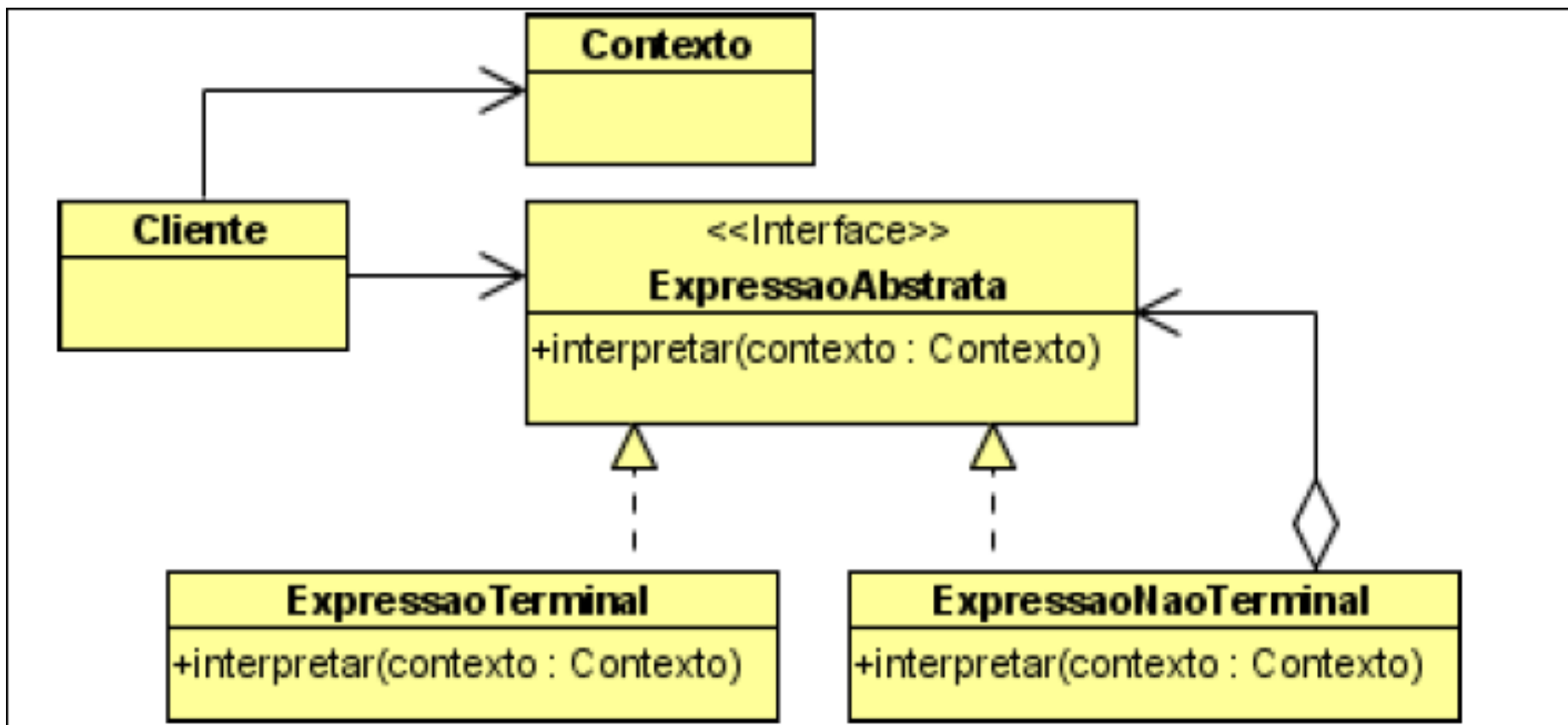
10 == 5 + 4 + 1



LE1 – Diagrama de Classes



Padrão de Projeto *Interpreter*



Considerações de Implementação

- Estrutura da implementação é semelhante à gramática da linguagem
 - Elementos da BNF são implementados por classes, interfaces e classes abstratas
- Parser (gerado pelo JavaCC)
- Cada unidade provê métodos:
 - **checaTipo()** // Checar tipo
 - **avaliar()** // Avaliar a expressão
 - **executar()** // Executar o comando (imperativa)



Exercícios

1. Estender a LE1 com ValorChar (character) e uma operação que retorna a ordem (ascii) do character
2. Implementar ValorReal e as operações aritméticas usuais sobre os reais
3. Implementar operadores relacionais como menor que, maior que, ..., para valores inteiros



LE2

Linguagem de Expressões 2

(Expressões com Declarações)



LE2 - Características

- LE2 estende LE1 com **identificadores** que possuem um **valor** (constante)
- Identificadores são declarados usando **let**
- As declarações no **let** são **colaterais**
- Um programa continua sendo uma expressão
- Durante a interpretação e verificação de tipos, surge a necessidade de contextos com:
 - Um mapeamento entre identificadores e valores
 - Um mapeamento entre identificadores e tipos

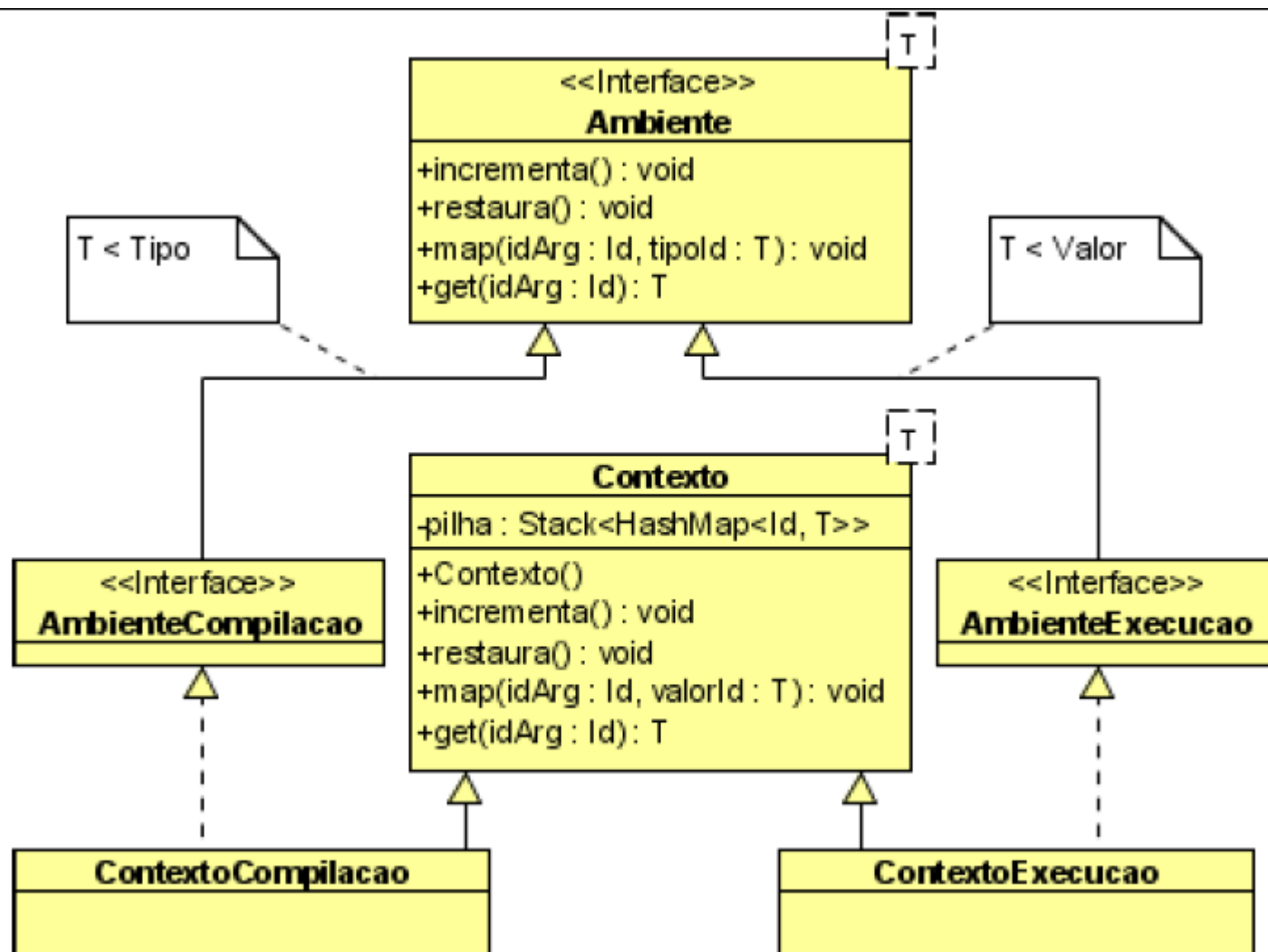


LE2 – Características (cont.)

- Na avaliação de uma expressão, a ocorrência de um identificador é substituída pelo valor associado ao identificador
- Durante a verificação de tipos, o tipo de um identificador que ocorre em uma expressão precisa ser conhecido para que a verificação da expressão seja analisada.



LE2 – Contextos e Ambientes



LE2 - Sintaxe

Programa ::= Expressão

Expressao ::= Valor | ExpUnaria | ExpBinaria
| ExpDeclaracao | Id

ExpDeclaracao ::= "let" DecVariavel "in" Expressao

DecVariavel ::= "var" Id "=" Expressao
| DecVariavel "," DecVariavel

Valor ::= ...

ExpUnaria ::= ...

ExpBinaria ::= ...



LE2 - Exemplos

let var x = 10 in x

let var x = 1 in
 let var x = 2 in
 x + 1

let var x = 10, var y = 5 in
 let var z = x + y in
 x + y + z



LE2 - Exemplos

Encontre o erro:

```
let var x = 10, var y = x + 5 in  
  let var z = x + y in  
    x + y + z
```



Exercícios

- Modifique o interpretador de LE2 de forma que as declarações em um bloco let sejam sequenciais ao invés de colaterais

