

---

Universidade Federal de Pelotas

## Um Compilador Simples

André Rauber Du Bois  
*dubois@inf.ufpel.edu.br*

---

## MOTIVAÇÃO

- Ilustrar de forma prática as fases de um compilador
- Um compilador para expressões aritméticas
- Gera código para uma máquina de pilha hipotética

---

## GRAMÁTICA DA LINGUAGEM

- Gramática:

$\text{Exp} \rightarrow \text{Num} \mid ' (' \text{Exp Op Exp} ') '$

$\text{Op} \rightarrow '+' \mid '*'$

$\text{Num} \rightarrow '0' \mid '1' \mid '2' \mid '3' \mid '4' \mid '5' \mid$   
 $'6' \mid '7' \mid '8' \mid '9'$

- Exemplos: 3, (5 + 8), (3\*(9+10)), (2\*((3\*4)+9))

---

# ESTRUTURA DO COMPILADOR

---

## EXEMPLO DE USO

- Usando o compilador

```
andre@andre:~$ cat teste
```

```
(3+(4*5))
```

```
andre@andre:~$ javac *.java
```

```
andre@andre:~$ java Compilador teste
```

```
PUSH 3
```

```
PUSH 4
```

```
PUSH 5
```

```
MULT
```

```
SUM
```

```
PRINT
```

---

## ARQUIVOS

- `Compilador.java`: principal, liga as partes
- `AnaliseLexica.java`: lê arquivo fonte e gera os tokens
- `Parser.java`: **Análise sintática** (gramática)
- `CodeGen.java`: gerador de código para máquina de pilha
- `ArvoreSintatica.java`: classes da árvore