

# COMPILADORES 2019/2020

## aula 0x04 - A Linguagem YA!

12/03/2020

Pedro Patinho <pp@di.uevora.pt>

Universidade de Évora - Departamento de Informática



1. A Linguagem Ya! (YaLang - Yet Another Language)
2. Exemplos

A LINGUAGEM YA! (YALANG - YET  
ANOTHER LANGUAGE)

# A LANGUAGE YA! (YALANG - YET ANOTHER LANGUAGE)

- Linguagem para aprendizagem de compiladores
- Sintaxe “fácil”
- Implementável por fases
- Nem alto nível, nem baixo nível

- Separador de instrução: ; (ponto e vírgula)
- Programa → Declarações
- Declaração de variável
  - Formato `id : tipo`
  - *ou* `id : tipo = valor`
  - *múltiplas variáveis* `id1, id2, ... : tipo`
  - *ou* `id1, id2, ... : tipo = valor`
- Declaração de função
  - `nome ( ) : tipo {...}`
  - *ou* `nome (arg1:tipo, arg2:tipo, ...):tipo {...}`
- Definição de novo tipo
  - `define nome tipo`

- Espaço em branco é ignorado (`\n`, `\r`, `\t`, [espaço])
- Statements (no corpo das funções)
  - declarações (variáveis locais)
  - expressões (afecções; outras expressões são simplesmente ignoradas)

`id = valor`

`id[intval] = valor`

`id1 = id2 = ... = idn = valor`

- `return valor` (pode não ter valor  $\rightarrow$  void)
- `if booleano then { stms }`
- `if booleano then { stms } else { stms }`
- `while booleano do { stms }`

- `int`
- `float`
- `string`
- `bool`
- `tipo[intval] → array`
- `void → para funções`

- inteiros (1, 2, 30000, ...)
- vírgula flutuante (1.2, 0.1, .23, .22E-20, ...)
- strings ("olá", "1", ...)
- booleanos (true, false)



- Numéricas

- $, -, *, /$
- $\text{mod}, ^$  (potenciação)

- Booleanas

- $=, <, >, <=, >=, !=$
- $\text{not}, \text{and}, \text{or}$

## YA! - FUNÇÕES PRÉ-DEFINIDAS

- Output ( `print(expressão)` )
- Input ( `input(lvalue)` )

EXEMPLOS

# EXEMPLO 1

---

```
1  a : int = 0;
2  b : string = "ola";
3
4  f (b: int) : void {
5      print(b);
6  };
7
8  main () : void {
9      a : bool = true;
10
11      if a then {
12          b = "hello, world";
13          f(b);
14      };
15  };
```

---

## EXEMPLO 2 - FACTORIAL

```
1  fact (n: int) : int {
2      f, i : int = 1;
3
4      while i <= n do {
5          f = f * i;
6          i = i + 1;
7      };
8
9      return f;
10 };
11
12 main () : void {
13     x : int;
14     print("Introduza um inteiro: ");
15     input(x);
16     print(fact(x));
17 };
```

## EXEMPLO 3 - FACTORIAL RECURSIVO

```
1  factrec (n: int) : int {
2      a,b: int;
3
4      if n == 1 then {
5          b = 1;
6      }
7      else {
8          a = n - 1;
9          b = n * factrec(a);
10     };
11     return b;
12 };
13
14 main () : void {
15     print(factrec(3));
16 };
```

# TRABALHO (PRIMEIRA PARTE)

## 1. **Implementar** analisador para Ya!

- Analisador lexical
- Analisador sintático (gramática + APT)

## 2. **Pensar** nos casos especiais

- Funções pré-definidas com argumentos de tipos variáveis

`print()`

`input()`

- Conversão de tipos implícita  
(ex: `'2 + 3.1'`, `'if ("ola") ...'`)