

# Lógica da Computação

## Aula 16

Raul Ikeda

1º semestre de 2020

# Esta Aula

- Declaração de Variáveis com tipagem forte

# Nova Raiz do DS

Vamos trocar a raiz do DS para o modelo final:

```
/* Mínimo programa viável */  
  
<?php  
    /* Command */  
  
?>
```

DS:



# Sem tipagem?

```
<?php
    $i = 3;
    echo gettype($i)
    $i = True;
    echo gettype($i)
?>
```

- Assim como o Python, a linguagem PHP atribui o tipo da variável durante a atribuição. Permitindo inclusive a troca do tipo em tempo de execução.
- Como faríamos então para abstrair e armazenar o tipo da variável?

# Inserindo Tipagem

- Criar um novo nó chamado BoolVal, semelhante ao IntVal, mas que guarda valores Booleanos: **True**, **False**.
- Na SymbolTable, além de guardar nome e valor da variável, agora tem que guardar o tipo, que poderá ser **int** ou **bool**, por enquanto.
- Todos os métodos Evaluate() que possuem *return*, deverão retornar um valor e um tipo. Usar preferencialmente tupla ou struct.
- Durante a Evaluate() do Assignment, é preciso atualizar a SymbolTable (**setter**) com o tipo do retorno do filho 1.

# Ajustando os tipos

- O resultado dos operadores devem ser:
  - Relacionais: bool
  - Booleanos: bool
  - Aritméticos: int
- No caso das folhas da AST no Factor:
  - IntVal: int
  - BoolVal: bool
  - Identifier: o tipo atual na SymbolTable
  - Readline: int
- Não haverá restrições em operar **int** com **bool**, realizando o mapeamento:
  - False é mapeado para o zero, e vice-versa
  - True é mapeado para o um
  - Qualquer inteiro diferente de zero é mapeado para True

# Strings!

- Vamos implementar strings:
  - Tokenizer: A string estará sempre entre "aspas duplas"
  - Tokenizer: Adicionar o operador de concatenação: .
  - Parser: Adicionar string no Factor
  - Parser: Adicionar . no Expression
  - AST: novo nó StringVal, semelhante a InVal. Deve retornar tipo "string"

```
{  
  $x = "abcd";  
  $y = "e";  
  echo $x.$y;  
}
```

# Semântico de verdade

- Seu compilador não deverá permitir operações booleanas e aritméticas com strings.
- MAS, deve permitir a concatenação de booleanos e inteiros, convertendo-os em string antes de executar a operação.
- Portanto, **ANTES** de realizar uma operação, é preciso verificar se os tipos de retornos dos filhos é plausível.

Por Exemplo

```
<?php
  $x = 1 + True; /* Ok */
  $x = 1 and True; /* Ok */
  $x = "a" . 1 . True; /* Ok */
  $x = "a" == "b"; /* Ok, resultado bool: False */
  $x = 1 + "a"; /* ERROR: Incompatible types */
  $x = True and "a"; /* ERROR: Incompatible types */
?>
```



# Atividade: Roteiro 7

- Roteiro Impresso ou PDF no Blackboard.
- **Atualizar EBNF e DS!**

# Próxima Aula

- Computabilidade e Decidibilidade
- Problema da Parada

## Referências:

- Boolos et al. Cap. 3
- Hopcroft et al. Cap. 7
- Sipser Cap 2.3