

DCC024 Linguagens de Programação  
2020/1

## Passagem de parâmetros

Haniel Barbosa



# Passagem de parâmetros

---

- ▶ Durante este curso temos lidado com declaração e chamada de funções
  
- ▶ Quais são diferentes formas em que a passagem de parâmetros para funções pode ser implementada?

# Alguns conceitos

---

- ▷ Parâmetros **formais** e parâmetros **reais**

```
void f(int x, int y) {}
```

```
...
```

```
f(1,0);
```

```
LetFun("f", "x", ..., Call(Var "f", IConst 1))
```

# Alguns conceitos

---

- ▷ Parâmetros **formais** e parâmetros **reais**

```
void f(int x, int y) {}
```

```
...
```

```
f(1,0);
```

```
LetFun("f", "x", ..., Call(Var "f", IConst 1))
```

- ▷ Como associar os parâmetros formais aos parâmetros reais?
  - ▶ Posicional
  - ▶ Nominal

## Exemplos: Python, C, C++

---

...

# Em resumo...

---

- ▷ Tipos de parâmetros:
  - ▶ *Formais*: aqueles declarados na lista de parâmetros da função
  - ▶ *Reais*: aqueles passados para a função, que substituirão os parâmetros formais
  
- ▷ Correspondência entre parâmetros reais e formais
  - ▶ *Posicional*: correspondência feita de acordo com a posição na chamada da função. Exemplo: quase todas as linguagens de programação.
  - ▶ *Nominal*: parâmetros são anexados a nomes, que os identificam durante a invocação da função. Exemplo: Python (também correspondência posicional).

# Em resumo...

---

- ▷ Tipos de passagem de parâmetros: avaliação gulosa (*eager*)
  - ▶ *Chamada por valor*: o parâmetro formal é como uma variável local no escopo da função, inicializada com o valor do parâmetro real no momento de chamada da função.
  - ▶ *Chamada por referência*: o parâmetro formal é um *alias* para o parâmetro real. Qualquer modificação afeta ambos, igual e simultaneamente.
  - ▶ Outros:
    - *Chamada por resultado*: o parâmetro formal é como uma variável local, porém não inicializada. Antes que a função retorne, o valor atual do parâmetro formal é copiado para o parâmetro real.
    - *Chamada por valor e resultado*: combinação de passagem por valor e por resultado: parâmetro formal como variável local, inicializada com o valor do parâmetro real, e, antes que a função termine, copia-se o valor atual do parâmetro formal para o real.

## Em resumo...

---

- ▷ Tipos de passagem de parâmetros: avaliação preguiçosa (*lazy*)
  - ▶ *Chamada por expansão de macros*: o corpo da macro é executado no escopo de chamada. Parâmetros formais são substituídos pelos parâmetros reais e reavaliados a cada ocorrência utilizada, no escopo daquela ocorrência na macro.
  - ▶ *Chamada por nome*: parâmetros formais são substituídos pelos parâmetros reais e reavaliados, no escopo de chamada, a cada ocorrência utilizada.
  - ▶ *Chamada por necessidade*: como por nome mas o parâmetro real é avaliado somente na primeira ocorrência utilizada do correspondente parâmetro formal. O resultado é armazenado para otimizar subseqüentes usos. A avaliação só é feita até onde é necessário pela chamada.