Introdução

Programação Funcional

Marco A L Barbosa
malbarbo.pro.br

Departamento de Informática
Universidade Estadual de Maringá

Este trabalho está licenciado com uma Licença Creative Commons - Atribuição-CompartilhaIgual 4.0 Internacional.
http://github.com/malbarbo/na-progfun

O que é?

O que é?

O que é programação imperativa?

- Um paradigma de programação onde os programas são descritos com sentenças que modificam o estado do programa.

O que é programação funcional?

- Um paradigma de programação onde os programas são descritos com aplicação e composição de funções.
- Evita mudança de estado (mudança do valor das variáveis)
- Evita efeitos colaterais (qualquer efeito que seja observável além do valor de saída da função, como a mudança dos parâmetro e variáveis global, exceções, entrada e saída, etc).

Por que?

Por que?

Um paradigma (linguagem) de programação é uma ferramenta.

Conhecer várias ferramentas permite utilizar a mais adequada para cada problema.

Compartilhamento de dados junto com mudança de estado é difícil!

Mudança de estado

Qual o valor de `lst`?

```
>>> lst = [0] * 3
>>> lst
[0, 0, 0]
>>> lst[1] = 10
>>> lst
```

```
[0, 10, 0]
```

Qual o valor de `lst`?

```
>>> lst = [[]] * 3
>>> lst
[[], [], []]
>>> lst[1].append(2)
>>> lst
```

```
[[2], [2], [2]]
```

```
def adiciona_todos(
        dest: list[int],
        fonte: list[int]):
    '''
    Adiciona todos os elementos
    de *fonte* no final
    de *dest*.
    '''
    for x in fonte:
        dest.append(x)
```

Qual o valor de `lst`?

```
>>> lst = [4, 3, 1]
>>> adiciona_todos(lst, [6, 2])
>>> lst
```

```
[4, 3, 1, 6, 2]
```

```
>>> adiciona_todos(lst, lst)
>>> lst
```

A execução não para!

As duas definições a seguir são equivalentes?

```
def soma_indices(lst: list[int], a: int, b: int) -> int:
    return indice(lst, b) + indice(lst, a)

def soma_indices(lst: list[int], a: int, b: int) -> int:
    return indice(lst, a) + indice(lst, b)
```

Não é possível afirmar que as duas definições são equivalentes sem olhar o código da função `indice`. Se a função `indice` tem efeitos colaterais, então as definições podem não ser equivalentes.

Efeitos colaterais

A possibilidade de efeitos colaterais **dificulta pensar localmente** sobre o funcionamento do código.

A ausência de efeitos colaterais **permite pensar localmente** sobre o funcionamento do código.

Como?

Como?

1) Escolher uma linguagem.
2) Estudar as construções do paradigma e as referências da linguagem.
3) Praticar lendo e escrevendo código.

1) Escolher uma linguagem
    - Student Gleam
    - Simples
    - Bom suporte ao paradigma funcional
    - Inferência de tipo
    - Fácil instalação

2) Estudar as construções do paradigma e as referências da linguagem

- A Tutorial Introduction to the Lambda Calculus
- Livro How to Design Programs
- Livro Structure and Interpretation of Computer Programs
- Tour da linguagem Gleam
- Documentação da biblioteca padrão da linguagem Gleam
- Página do sgleam

3) Praticar lendo e escrevendo código
    - Muitos exemplos
    - Muitos exercícios

Primeiros passos

Instalação

No Linux

```
$ curl -s -L https://malbarbo.pro.br/sgleam/sgleam.tar.gz | tar xvz
```

ou

```
$ wget -qO- https://malbarbo.pro.br/sgleam/sgleam.tar.gz | tar xvz
```

Em outros sistemas

Acesse https://malbarbo.pro.br/sgleam/ e faça o download e descompactação manualmente.

Execução

Considere o arquivo `dobro.gleam` com o conteúdo

```
import gleam/io

pub fn dobro(x: Int) -> Int {
    x * 2
}

pub fn main() {
  io.debug(dobro(4))
}
```

Para executar o arquivo no Linux digite

```
$ ./sgleam dobro.gleam
8
```

No Windows

```
$ .\sgleam dobro.gleam
8
```

Execução do repl

REPL (*Read Eval Print Loop*)

- A expressão é lida (*Read*)
- A expressão é avaliada (*Eval*)
- O resultado da avaliação é exibido (*Print*)
- O processo é repetido (*Loop*)

Para iniciar o repl

```
$ ./sgleam
Welcome to sgleam.
Type "quit" or CTRL-D to exit.
> 2 + 5
7
```

Para carregar um arquivo e iniciar o repl

```
$ ./sgleam -i dobro.gleam
Welcome to sgleam.
Type "quit" or CTRL-D to exit.
> dobro.dobro(4)
8
```

Leitura

Leitura

Recomendada

- Tour da linguagem Gleam
- Programação funcional

Extra

- The Python paradox
- Revenge of the Nerds
- Beating the averages