

Programação Funcional

Apresentação da disciplina

Prof. Giovanni Lucero

DCOMP/UFS

# Programação Funcional

- O que é Programação?
- O que é Funcional?

# Programação

Programming is a way to “instruct the computer to perform various tasks”

[hackr.io/blog/what-is-programming](https://hackr.io/blog/what-is-programming)

- Programação não é somente codificar
- Instruct the computer → Linguagem
- Que tipo de tarefas?

“Computer programming is the process of designing and building an executable computer program for accomplishing a specific computing task”

Wikipedia

- Implicitamente especificação
- Projetar e construir
- Programa

“Programming is coding, modeling, simulating or presenting the solution to a problem, by representing facts, data or information using pre-defined rules and semantics, on a computer or any other device for automation”

Wikiversity

- Solução de problemas
- Usando uma linguagem de programação
- Paradigma de programação

- Computador executa instruções muito simples tais como somar ou comparar dois números (instruções de máquina)
  - Muita velocidade
  - Memória muito grande para armazenamento de resultados intermediários
  - Computador “entende” linguagem de máquina
- Tarefas complexas são feitas combinando muitas instruções de máquina
- Linguagens de programação de alto nível
  - Oferece um modelo de programação que abstrai a linguagem de máquina
  - Linguagens imperativas (como fazer a tarefa)
  - Linguagens declarativas (qual é a tarefa)
    - Linguagens funcionais (Haskell, Lisp, ...), Lógicas (Prolog, Mercury, ..), ...

Programa  
Linguagem de alto nível



Compilador



Programa  
Linguagem de Máquina

Programa  
Linguagem de alto nível

```
graph LR; A[Programa Linguagem de alto nível] --> B[Interpretador]; subgraph C [Computador]; B; end
```

The diagram illustrates the execution of a high-level language program. On the left, a blue rectangular box contains the text "Programa Linguagem de alto nível". A blue arrow points from this box to a smaller blue rectangular box labeled "Interpretador". This "Interpretador" box is positioned inside a larger blue rectangular box labeled "Computador" at its bottom right corner.

Interpretador

Computador

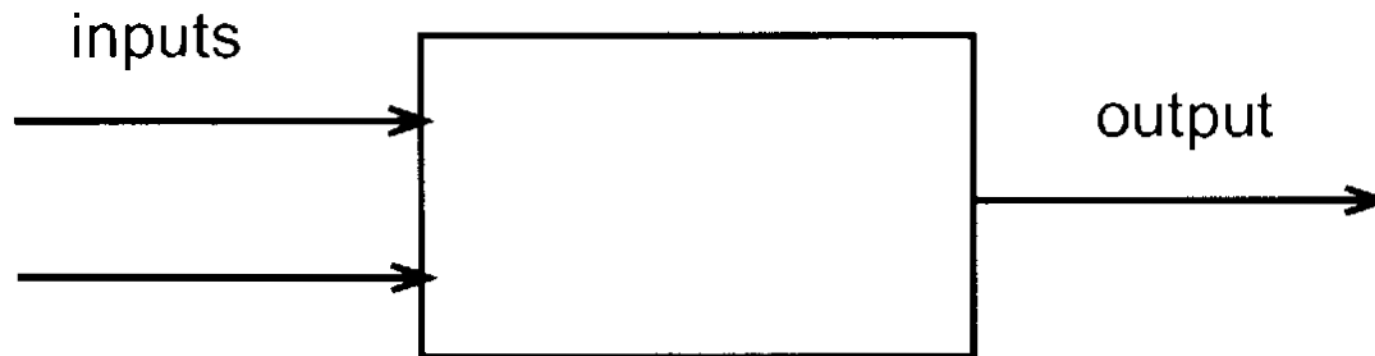


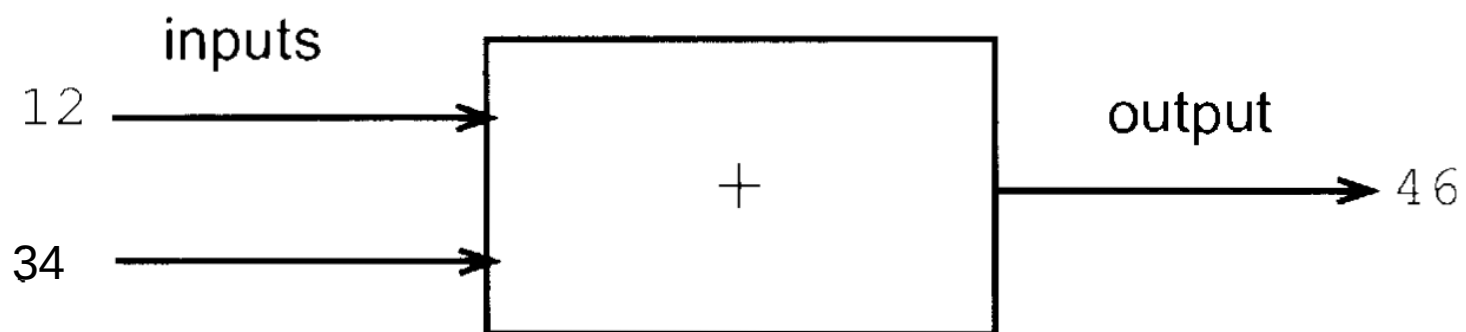
# Funcional

- Programa: Conjunto de definições de dados e funções
- Computação: avaliação de funções
- Avaliação: similar como fazemos em álgebra
- Não há estado

# Função

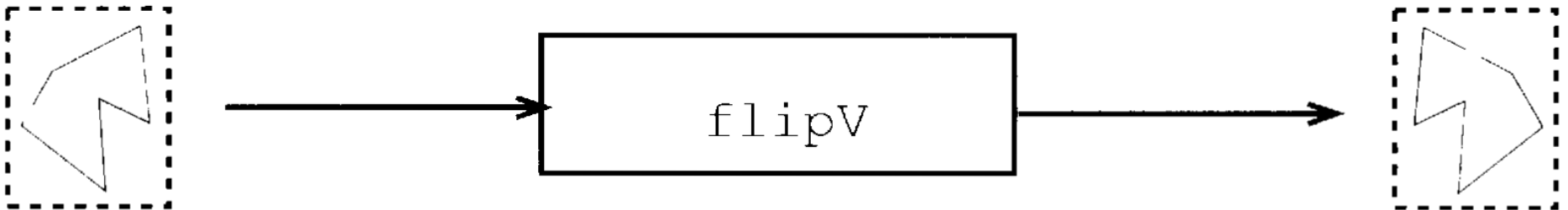
- Relação binária tal que ....
- “Caixas pretas” que recebem valores de entrada e produzem um valor que depende dos valores da entrada





Valores não precisam ser somente números

Podem ser entidades mais elaboradas, como por exemplo figuras



Em Haskell, definições de funções são dadas usando equações

Por exemplo

```
somaDobro x y = 2*(x+y)
```

O modelo computacional de cômputo é muito simples e claro

- A avaliação de expressões é feita usando as equações no sentido de esquerda para a direita
- Por exemplo

```
3 + somaDobro 4 5 → 3 + 2*(4+5)
                  → 3 + 2*9
                  → 3 + 18
                  → 21
```

- Programação funcional é só isso?
- Há outros conceitos importantes
  - Funções de alta ordem
  - Tipos e polimorfismo paramétrico
  - Tipos compostos: listas
  - Avaliação preguiçosa

# Objetivos da Disciplina

- Aprender o paradigma da Programação Funcional.
  - Dominar os elementos de estilo funcional
  - Saber aplicá-los na solução de problemas e na construção de softwares
  - Raciocinar sobre programas funcionais
    - Desenvolver a habilidade de realizar provas matemáticas envolvendo programas.

# Ementa

- Indução Matemática.
- Resolução de problemas por indução.
- Paradigma funcional.
  - Expressões. Valores. Funções e tipos.
  - Tipos primitivos de dados e tipos derivados.
  - Recursividade.
  - Definições por casamento de padrão.
  - Polimorfismo paramétrico e funções de alta ordem.
  - Avaliação de expressões lazy e eager.
  - Programação com listas.
  - Tipos algébricos.
- Solução de problemas de baixa complexidade.
- Raciocínio com programas.



# Metodologia

- Aulas teóricas e práticas
- Exercícios
  - Nos laboratórios
  - Extra-aula

- Aluno deve ser pró-ativo
  - Pelo menos 4hs de estudo extra-aula
  - Ler fontes bibliográficas adotadas e complementares
  - Fazer todos os exercícios propostos
  - Discutir soluções com colegas e monitores
  - Dúvidas? Procurar monitores e professor
  - Participar em sala de aula
- O professor não está para ensinar. Vocês estão para aprender, com o professor como guia.

# Avaliação

- Três testes
  - cada um com peso 30%
- Exercícios extra-aula
  - com peso 10%

# Bibliografia

- Livros principais
  - Simon Thompson, Haskell: The Craft of Functional Programming, Addison-Wesley, 2nd Ed.
  - Miran Lipovaca, Learn You a Haskell for Great Good, 2011
- Site principal:
  - <http://www.haskell.org>