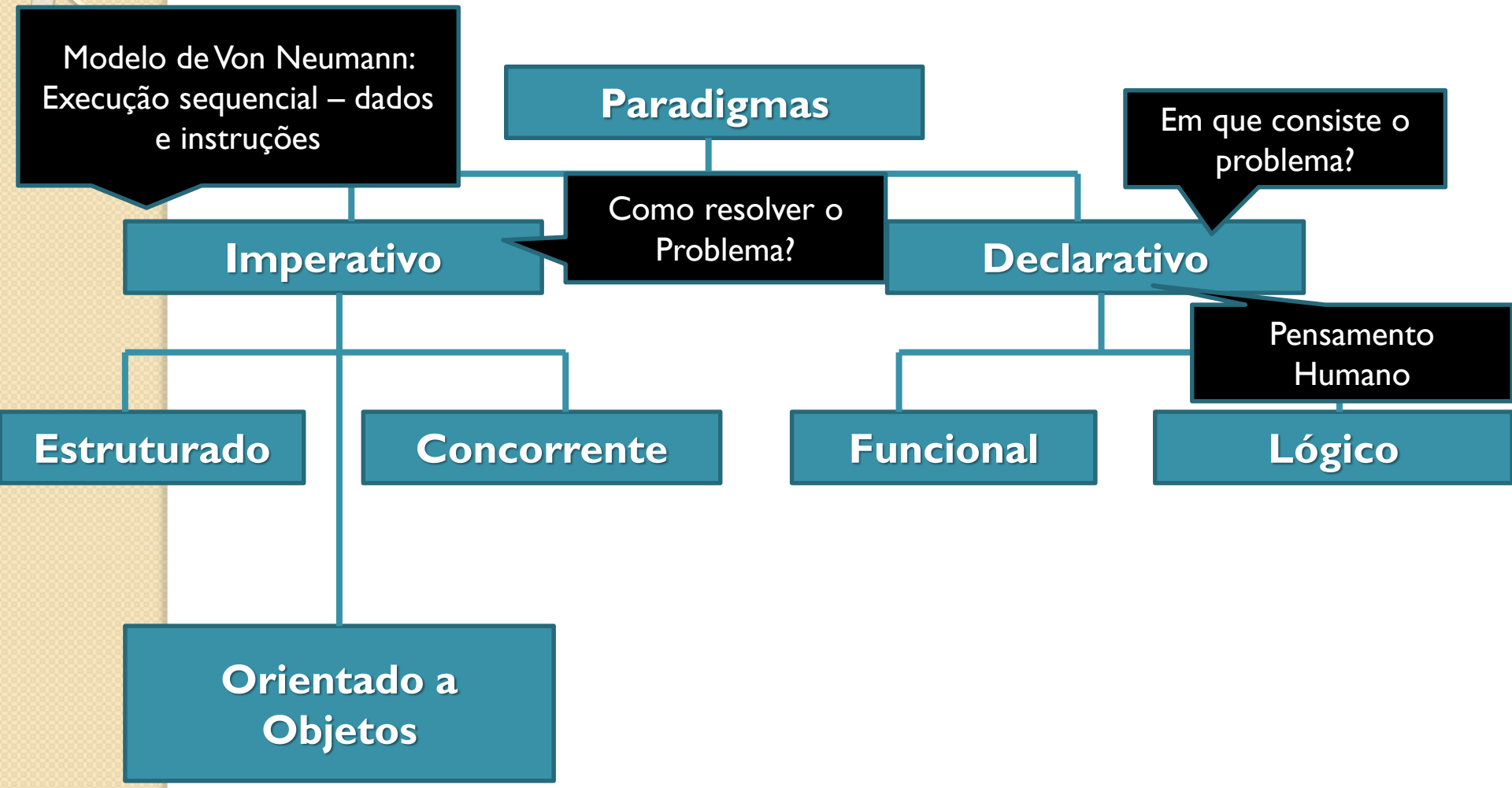
Paradigma Funcional

Profª Maria Adelina Raupp Sganzerla

Paradigmas de Linguagem de Programação

Ulbra – Gravataí – 2015/1

Paradigmas de Linguagem de Programação



Modelo Funcional de Programação

- Fundamentos de LP Funcional:
  - Conceito matemático de função
  - Programas formados por funções
- Amplamente utilizadas na área de Inteligência Artificial;
- Manipulação de estruturas de dados complexas:
  - Usualmente simbólicas

Programação Funcional

- É um Paradigma de Programação que:
  - Enfatiza a avaliação de expressões;
  - Não utiliza comandos e algoritmos (sequência de comandos);
  - Não utiliza variáveis e atribuições (transparência referencial)
    - O valor de uma variável nunca muda com o fluxo de controle (conceito de empilhamento e recursividade).

Programação Funcional

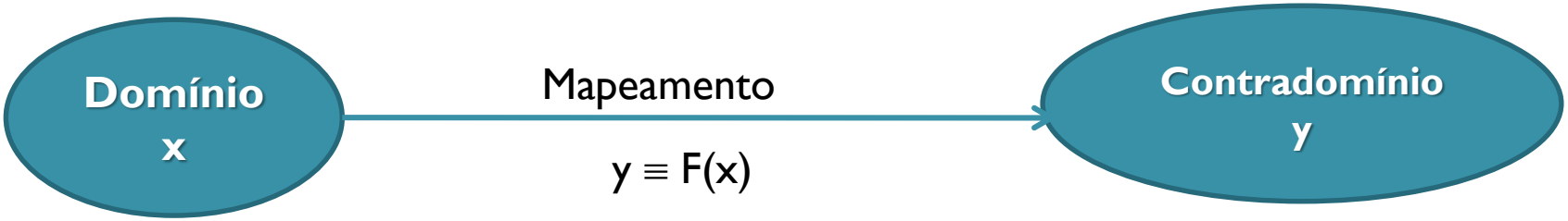
- Exige disciplina de programação;
- Produz programas que possuem facilidade de verificação;
- Facilita a programação modular:
  - Funções complexas podem ser construídas a partir de funções simples
  - Programas podem ser construídos a partir de outros programas:
    - Programas menores e mais eficientes que na programação procedural.

Algumas LPs Funcionais

- Uma breve lista:
  - FP (*Functional Programming*): Backus, 1978
  - LISP (*List Processing*): Mac Carthy, 1960
  - APL (*A Programming Language*): Iverson, 1960
  - Haskell: Hudak, 1988
  - Scheme (Dialeto Lisp): MIT, por volta de 1975
  - ML: Milner, 1980 a 1990

Base: Conceito de Função

- Conceito matemático de Função:



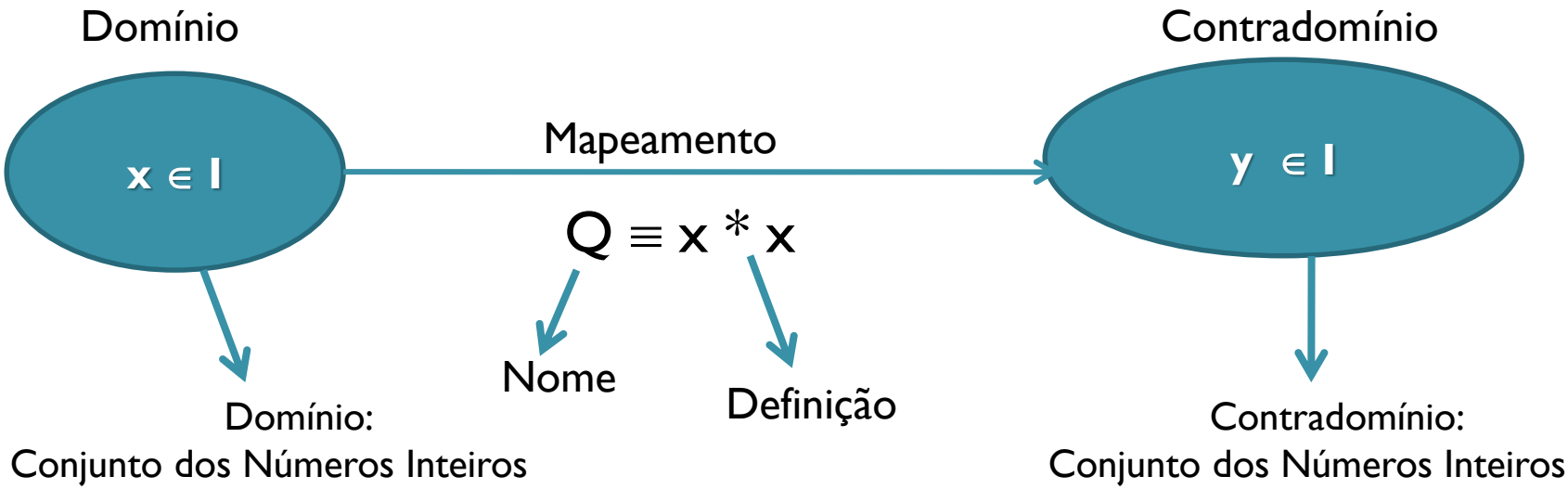
  - **Domínio:** conjunto de objetos sobre os quais a função pode ser aplicada
  - **Contradomínio:** conjunto de objetos resultantes da aplicação da função
  - **Definição da Função:** especifica a regra de mapeamento

Linguagens de Programação Funcional – LPF

- Objetivo:
  - Assemelhar-se ao máximo possível as funções matemáticas

- Em uma LPF, a avaliação de uma função sempre produz o mesmo resultado a partir dos mesmos parâmetros:
  - Independente de contexto e execuções anteriores

Base: Conceito de Função
Exemplo 1 – Quadrado de x



- Uma Função é uma regra para mapear (associar) elementos de um conjunto (o domínio) em elementos de outros conjuntos (o contradomínio).

- Exemplo do Cubo (Função para calcular o cubo):

Cubo(x)

```
X          →   x * x * x
(2  3  4)  →  (8  27  64)
```

Domínio → Contradomínio

`Cubo(x) ≡ x* x * x` (símbolo ≡ é equivalência)

Componentes da LPF

- **Conjunto de Função Primitivas**: são pré-definidas pela Linguagem;
- **Conjunto de Formas Funcionais**: são mecanismos pelos quais se podem combinar funções para criar novas funções;
- **Operação de Aplicação**: é o mecanismo embutido para aplicar uma função a seus argumentos e produzir um valor;
- **Conjunto de Dados (ou objetos):** são elementos permitidos nos domínios.

Scheme

- Dialeto do LISP:
  - Projetada para ser simples, de fácil aprendizagem e apropriada para o ensino de programação

- Tudo o que é escrito em Scheme é chamado, genericamente, de expressão:
  - Notação pré-fixada:

$$5 + 3 \rightarrow (+\ 5\ 3)$$

$$(5 + 3) * 2 \rightarrow (*\ (+\ 5\ 3)\ 2)$$

Scheme

- Expressões podem ser comentadas se precedidas de “;” (ponto e vírgula);

- Expressões podem ser de dois tipos:
  - Átomos
  - Listas.

Scheme - Exemplos

Expressões: Resultantes	Valores
`42`	
`(* 3 7)`	
`(+ 5 7 8)`	
`(- 5 6)`	
`(-15 7 2)`	
`(-24(* 4 3))`	

Scheme - Sintaxe

- Átomos são objetos simples que podem ser de três tipos:
  - Numerais:
    - `4`
    - `-3.65`
  - Textos (strings) escritos entre aspas:
    - `"Um exemplo de texto"`
  - Símbolos:
    - `a`
    - `define`
    - `+`

Scheme - Sintaxe

- Uma lista é uma sequência de objetos entre parênteses, separados por espaços
  - `(+ 3 4)`
  - `(define (soma  x) (+ 3 x))`
  - `(display "Ola Turma!")`

Scheme - Sintaxe

- Um Programa Scheme é uma sequência de definições e aplicações de funções
  - Expressões entre parênteses
  - Exemplo: Função para calcular a área de um cubo

```
(define (cubo x) (* x x x))
(cubo 3)
 27
```

Passos para resolver uma Expressão em Scheme

- Expressão

```
(+ (* 8 -5) 7.3 (* 2 7.3))
```

Executa a multiplicação mais à esquerda

```
(+ -40 7.3 (* 2 7.3))
```

Executa multiplicação

```
(+ -40 7.3 14.6)
```

Executa a soma

-18.1 → resultado final

Definindo Funções - define

- Exemplo de Funções definidas por outras Funções

```
(define (quadrado x) (* x x))
        (quadrado 2)
             4

(define (areacirculo raio)
 (*3.1416 (quadrado raio)))
        (areacirculo 5)
          78.539999999
```

Definindo Funções – define
Exercício

- Defina uma Função que a partir da base **a** e da altura, calcule a área de uma triângulo.

Definindo Funções – define

- Expressão: definições globais
  - `(define a 7)`
  - `(define b -5)`
  - `(+ (* 8 b) a (* 2 a))`

- Resolva a Função acima (teste de mesa)