

Paradigmas de Programação: Uma Introdução

Profa. Juliana Vizzotto

`juvizzotto@inf.ufsm.br`

Disciplina de Paradigmas de Programação

Roteiro

- Objetivos da disciplina
- Programa da disciplina
- Informações gerais
- Uma breve introdução
- Bibliografia

Objetivos

- Introduzir os conceitos fundamentais sobre *linguagens de programação*
- Estudar os principais paradigmas de programação incluindo imperativo, orientado a objetos, lógico, funcional e concorrente
- A Ênfase será nos seguintes pontos:
 - Estudo dos principais paradigmas e o processo de raciocínio associado com cada um.
 - Estudar como características específicas das linguagens contribuem para prática de programação.

Programa

- 1 Programação Funcional
 - *Haskell*: <http://www.haskell.org>
- 2 Programação Lógica
 - *KANREN*: <http://kanren.sourceforge.net> ou programação em lógica em haskell...
- 3 Programação Orientada a Objetos
 - Java, C++, Ruby
- 4 Programação Concorrente

Avaliação

- Trabalhos
- Provas

Paradigmas de Linguagens de Programação

“Padrões de raciocínio para resolução de problemas”
[Peter Wegner, 1998]

- Como os paradigmas de programação foram descobertos?
- Uma breve história sobre linguagens de programação
- Desde antes de 1940 até os desafios da atualidade

Paradigmas de Linguagens de Programação

Linguagens Imperativas

- Atribuição: variáveis guardam valores
 - Iteração: *loops*
- 1 Linguagens procedurais: estruturadas em blocos
 - 2 Linguagens Orientadas a Objetos

Paradigmas de Linguagens de Programação

Linguagens Imperativas Procedurais

- O modelo de execução de um programa é dado pelas transições de estados das variáveis
- Exemplo: Fortran, Algol, Pascal, **C**
- Conjunto de procedimentos ou funções

```
int fact(int n){  
    int sofar = 1;  
    while (n>0) sofar *= n--;  
    return sofar;  
}
```

Paradigmas de Linguagens de Programação

Linguagens Orientadas a Objetos

- Objetos contém informações de estado privadas e se comunicam através da troca de mensagens
- Conceito de classe
- Exemplos de linguagem: C++, **Java**
- Exemplo: um objeto que guarda um valor inteiro e *sabe* como *retornar* ambos o valor inteiro e seu fatorial

```
public class MyInt{  
    private int value;  
    public MyInt(int value){  
        this.value = value;  
    }  
}
```

Linguagens Orientadas a Objetos

```
public class MyInt{
    private int value;
    public MyInt(int value){
        this.value = value;
    }
    public int getValue(){
        return value;
    }
    public int getFact(){
        return new MyInt(fact(value));
    }
    private int fact(int n){
        int sofar = 1;
        while (n > 1) sofar *= n--;
        return sofar;
    }
}
```

Paradigmas de Linguagens de Programação

- Código parece mais complicado
- O paradigma orientado a objetos ajuda a manter programas grandes organizados
- Difícil de ver as vantagens em programas pequenos.

Paradigmas de Linguagens de Programação

- Código parece mais complicado
- O paradigma orientado a objetos ajuda a manter programas grandes organizados
- Difícil de ver as vantagens em programas pequenos.

Paradigmas de Linguagens de Programação

Linguagens Declarativas: programação de alto-nível

- Expressar o mesmo problema com menos detalhes
- Foco em “o quê” (deve ser feito) ao invés de “como” (deve ser feito)

- 1 Linguagens Funcionais
- 2 Linguagens Lógicas

Paradigmas de Linguagens de Programação

Linguagens Funcionais

- Avaliação de funções
- Um programa é o resultado da avaliação de uma expressão
- Funções de alta-ordem
- Exemplos de linguagens: ML, Lisp, Scheme, **Haskell**
- **Recursão**

```
fact :: Int -> Int
fact x = if x <= 0 then 1 else x * fact(x -1)
```

```
(defun fact (x)
  (if (<= x 0) 1 (* x (fact (- x 1)))))
```

Paradigmas de Linguagens de Programação

- Fatorial parece bem mais elegante quando programado em linguagens funcionais. Por quê?
- Existem outros tipos de operações que são intrinsecamente melhor implementadas em paradigma imperativo. Quais?
- Por exemplo *multiplicação de matrizes*. Por quê?
- Acesso randômico em estruturas de dados

Paradigmas de Linguagens de Programação

- Fatorial parece bem mais elegante quando programado em linguagens funcionais. Por quê?
- Existem outros tipos de operações que são intrinsecamente melhor implementadas em paradigma imperativo. Quais?
- Por exemplo *multiplicação de matrizes*. Por quê?
- Acesso randômico em estruturas de dados

Paradigmas de Linguagens de Programação

- Fatorial parece bem mais elegante quando programado em linguagens funcionais. Por quê?
- Existem outros tipos de operações que são intrinsecamente melhor implementadas em paradigma imperativo. Quais?
- Por exemplo *multiplicação de matrizes*. Por quê?
- Acesso randômico em estruturas de dados

Paradigmas de Linguagens de Programação

Linguagens Lógicas

- Um programa é visto como um conjunto de fatos (verdades conhecidas) e regras (para inferir novos conhecimentos a partir de fatos e regras conhecidas)
- Base de conhecimento
- Facilidade de inferência
- Execução (*queries*): consulta à base de fatos e regras para verificação de uma proposição
- Exemplos de linguagens: **Prolog**, kanren

Paradigmas de Linguagens de Programação

Linguagens Lógicas

```
fact(X,1) :-  
    X == 1.  
fact(X,Fact) :-  
    X > 1,  
    NewX is X - 1,  
    fact (NewX,NF),  
    Fact is X * NF.
```

Máquina de inferência!

Paradigmas de Linguagens de Programação

Linguagens Concorrentes:

- Programas podem ter partes que executam ao mesmo tempo
- Exemplos de linguagens: Java, C+biblioteca, Haskell+biblioteca
- Exemplos de aplicações: Servidor Web (e.g, Apache), Jogos, Sistemas Operacionais, Computação Científica

- 1. Modern Programming Languages: A Practical Introduction
Adam Brooks Webber. 2002.