

Programação Funcional

Profa. Juliana Kaizer Vizzotto

juvizzotto@inf.ufsm.br

Disciplina de Paradigmas de Programação

Motivação

- Programação declarativa preocupa-se em especificar o problema e a solução e não com a maneira com a qual esta solução será encontrada. Uma linguagem declarativa preocupa-se *com o que deve ser computado* e não como um valor deve ser computado.
- Na programação imperativa o programador se preocupa em especificar uma lista de instruções que devem ser executadas em uma certa ordem.
- OBJETIVO da disciplina: estudar Programação Funcional com a Linguagem Haskell.

Motivação para Programação Funcional

- Ler o artigo de John Hughes, *Why Functional Programming Matters*, disponível no site da disciplina.
- Vale a pena aprender LISP pela experiência esclarecedora que o aprendizado proporciona. Essa experiência irá torná-lo um melhor programador para o resto de sua vida, mesmo que você não use muito a linguagem LISP.

LISP × HASKELL

- A função fatorial de um número natural é o produto de todos os n primeiros números naturais. Ex: $fat(3) = 3 * 2 * 1$
- Definição da função fatorial em Lisp:

```
(defun fatorial(n)
  (if (<= n 1)
      1
      (* n (fatorial (- n 1)))))
```
- Definição da função fatorial em Haskell

```
fatorial 0 = 1
fatorial n = n * fatorial (n - 1)
```

Programação Funcional

- Lisp foi a primeira linguagem de programação funcional. Ela foi descrita por MacCarthy do MIT em 1960.
- Nesta disciplina vamos estudar uma linguagem de programação funcional moderna, chamada Haskell, que possui várias características interessantes:
 - Linguagem puramente funcional: Não existe atribuição!
 - Sistemas de tipos avançados: Polimorfismo de tipos, classes de tipos.
 - Avaliação Lazy: argumentos são avaliados somente quando necessário. Possibilita o trabalho com estrutura de dados infinitas.
 - IO usando mônadas

O que é Programação Funcional?

- Programação funcional é um estilo de programação que enfatiza a avaliação de expressões. Uma expressão seria composta da aplicação de funções e argumentos.
- Uma linguagem funcional é uma linguagem que estimula ou até obriga o programador escrever programas pensando apenas em funções e nos valores que elas computam.

Linguagens Funcionais

- Funções são os elementos principais neste tipo de linguagem.
- Funções podem receber funções como argumentos e uma função pode devolver como resultado outra função.
- Estruturas de dados podem conter funções como elementos.

Vantagens de Linguagens Funcionais

- Programas são pequenos e com alto poder de expressão.
- Suporta componentes reusáveis de software.
- Permite prototipação rápida.
- Permite verificação formal de programas. As funções em uma linguagem funcional são funções matemáticas puras, não existe atribuição. Uma função que recebe os mesmos argumentos sempre devolve o mesmo resultado.

???

- Se linguagens funcionais são tão legais como é que eu nunca ouvi falar delas?
- Um paradigma de programação leva um longo tempo até ser adotado. Ex: orientação a objetos surgiu nos anos 60.
- Linguagens funcionais são ensinadas na maioria dos cursos de computação da EUROPA e EUA, e em alguns cursos no Brasil.
- Várias das coisas legais que encontramos nas linguagens de programação modernas surgiram nas linguagens funcionais. Ex:
 - Garbage Collection (Java, Python, Perl, etc)
 - Funções Anônimas (Python, PHP)
 - Programação genérica, polimorfismo de tipo (Java 1.5)

◀ ▶ ↻ 🔍

???

- Se linguagens funcionais são tão legais por que ninguém usa????
- Paulo Coelho vende muito mais que Dostoevsky. Será que ele é um escritor melhor? :-)
- Linguagens funcionais são utilizadas em várias empresas e projetos:
 - O LISP foi utilizado por Paul Graham para desenvolver o sistema de e-commerce via web, que posteriormente foi vendido para o Yahoo por US\$40 milhões, na época do boom da internet.
 - Autocad possui partes implementadas em LISP
 - Super Mario 64 usa LISP internamente
 - ERLANG: Linguagem funcional desenvolvida pela Ericson

◀ ▶ ↻ 🔍