

Linguagens de Programação e Programação Funcional

GSI004 — Aula 01*

Faculdade de Computação – UFU

*baseada nos slides do Professor Paulo Henrique Ribeiro Gabriel

30 de setembro de 2014

O que é uma linguagem de programação?

- Inicialmente, computadores eram programados utilizando sequências de números
- Chamávamos essas sequências de **linguagem de máquina**
- No entanto, sua leitura é complicada para seres humanos
 - ▶ Além de facilitarem o surgimento de erros
- Por esse motivo, buscou-se linguagens de mais alto nível, que abstraíssem os computadores, tornando a programação mais simples

O que é uma linguagem de programação?

Linguagem de Programação

Linguagem de Programação: sintaxe (gramática) e semântica (significado) utilizadas para escrever (ou codificar) um programa.

Uma linguagem de programação é também um código por meio do qual o ser humano se comunica com o computador, transmitindo instruções que devem ser seguidas para a resolução de um problema.

Paradigmas de Linguagens de Programação

De um modo geral, podemos classificar as linguagens de programação em duas classes:

- 1 Linguagens imperativas
- 2 Linguagens declarativas

Paradigmas de Linguagens de Programação

Linguagens Imperativas

- Programas possuem um conjunto de variáveis (**estado implícito**) que é modificado por comandos da linguagem
- Tais linguagens têm, em geral, uma noção explícita de **sequência de comandos** que permite o controle dos estados
- Linguagens imperativas expressam **como** alguma coisa será computada
- Exemplos: Fortran, Pascal, Ada, C, C++, Java, C#, etc. . .

Paradigmas de Linguagens de Programação

Linguagens Declarativas

- Programas **não** possuem um estado implícito
- Programas são construídos por meio de **expressões** (ou termos) em vez de comandos
- Linguagens declarativas expressam **o quê** será computado
- São divididas em dois grupos:
 - 1 Linguagens funcionais
 - 2 Linguagens lógicas

Paradigmas de Linguagens de Programação

Linguagens Funcionais

- O modelo de computação é o conceito matemático de **função**
- Ao computar uma função, aplica-se zero ou mais argumentos para computar **um único** resultado
- Linguagens puramente funcionais: FP, Haskell, Miranda
- Linguagens híbridas: Lisp, Scheme, Racket

Paradigmas de Linguagens de Programação

Linguagens Lógicas

- O modelo de computação é o conceito de **lógica** de predicados
- A computação é uma associação de um grupo de valores com *backtracking* para resolver valores adicionais
- Exemplos: Prolog, Parlog, KL1

Programação Funcional

- Este é um curso de **programação funcional**
- Programação:
 - ▶ Ênfase em **análise** e **solução** de problemas
 - ▶ **Desenvolvimento** de algoritmos e estruturas de dados que apoiem essa solução
 - ▶ Como tais algoritmos e estruturas de dados devem ser processados pelo computador
 - ▶ Foco no pensamento humano
- Funcional:
 - ▶ Programas são construídos pela definição de **funções matemáticas**
 - ▶ Programas funcionais consistem em **expressões** que utilizam essas definições
 - ▶ Foco em técnicas de solução de problemas, algoritmos, estruturas de dados e notações apropriadas à abordagem funcional

Programação Funcional

- Este **não** é um curso de **linguagens funcionais**
 - ▶ Não aprenderemos a fundo uma linguagem funcional
 - ▶ Embora precisamos de uma linguagem para nos auxiliar
 - ▶ Precisamos aprender a usar uma linguagem de modo que nossos programas sejam eficientes

Vantagens da Programação Funcional

- 1 Programas funcionais são mais fáceis de serem manipulados matematicamente que programas imperativos
- 2 Linguagens funcionais têm poderosos mecanismos de abstração
- 3 Programação funcional permite novas abordagens algorítmicas
- 4 Programação funcional permite novas abordagens para desenvolvimento de programas
- 5 Programas funcionais podem ser facilmente paralelizáveis
- 6 Programas funcionais são importantes em diversas áreas de pesquisa
- 7 Programação funcional tem relação direta com a Teoria da Computação
- 8 Programação funcional ajuda a “abrir a cabeça” de estudantes de computação

Desvantagens da Programação Funcional

- 1 Programas funcionais costumam ser lentos
- 2 Linguagens funcionais puras têm pouco uso na indústria
- 3 Programação funcional parece pouco natural

Um Pouco de História

Década de 1930

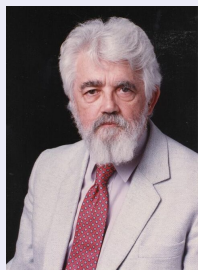
Alonzo Church desenvolveu o **cálculo- λ** , uma simples e poderosa teoria sobre funções



Um Pouco de História

Década de 1950

John McCarthy desenvolveu a linguagem Lisp, a primeira linguagem funcional, com influências do cálculo- λ , porém mantendo o conceito de atribuições de variáveis



Um Pouco de História

Década de 1960

Peter Landin a linguagem ISWIM, a primeira linguagem puramente funcional, totalmente baseada no cálculo- λ



Um Pouco de História

Década de 1970

John Backus desenvolveu a linguagem FP, com ênfase em funções de ordem superior



Um Pouco de História

Década de 1970

Robin Milner e seus colaboradores desenvolveram a linguagem ML, a primeira linguagem funcional moderna, que introduziu conceitos de inferência de tipo e tipos polimórficos



Um Pouco de História

Década de 1970-80

David Turner desenvolveu diversos estudos que culminaram na criação do sistema Miranda



Um Pouco de História

1985

Um comitê internacional de pesquisadores iniciou o desenvolvimento da linguagem Haskell, a qual seria um padrão entre as linguagens funcionais

Em 2003 publicaram a primeira especificação da linguagem (Haskell 98), a qual foi modificada em 2009 para dar origem ao padrão atual (Haskell 2010)

