

Programação Funcional

Paradigmas de Linguagens Computacionais (IF686)

Centro de Informática da UFPE - 2015.1

Fabrizio Batista Pereira (fbp)

Leandro Ferraz Farias (lff2)

Thaís Freire Cavalcante (tfc2)

Introdução

“Um paradigma de programação é um **padrão** de resolução de problemas que se relaciona a um determinado gênero de programas e linguagens” (TUCKER; NOONAN, 2010).

- Programação **funcional**;
- Programação imperativa;
- Programação orientada a objetos;
- Programação lógica.

Definição

"A programação funcional modela um problema computacional como uma coleção de **funções matemáticas**, cada uma com um espaço de entrada (**domínio**) e resultado (**faixa**)."

(TUCKER; NOONAN, 2010)

Exemplos: Lisp, Scheme, **Haskell** e ML.

Diferenciais

A programação funcional possui como foco **avaliar funções**, enquanto a programação imperativa, por exemplo, consiste em executar instruções.

Em linguagens puramente funcionais uma função apenas realiza cálculos e retorna o resultado, **sem que haja alteração de estados**, o que pode ocorrer, por exemplo, na programação imperativa.

Vantagens

Por não alterar estados, esse paradigma possui o benefício da **transparência referencial**, visto que uma função chamada duas vezes com os mesmos parâmetros retorna o mesmo resultado. Assim, o compilador pode avaliar mais facilmente o **comportamento do programa**.

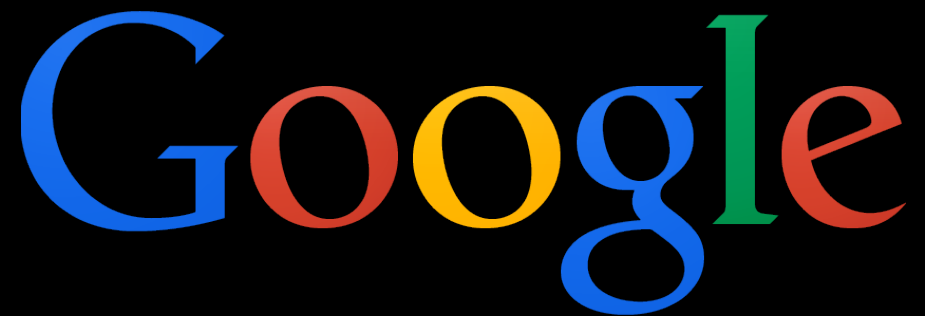
Dessa forma, também se torna mais fácil construir **funções mais complexas** a partir de funções mais simples, o que permite a **modularização**.

Origens

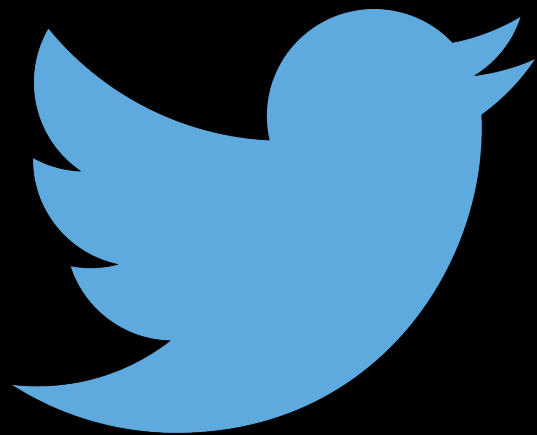
Assim como a programação lógica, a programação funcional se desenvolveu através de programadores de **inteligência artificial**.

A primeira linguagem do paradigma funcional, **Lisp** ("List Processor"), foi projetada em 1960 por John McCarthy.

Aplicações



Publica artigos
mostrando como utiliza o
paradigma funcional.



Twitter utiliza a linguagem
Scala no back-end.

Aplicações



Criou a linguagem funcional
Erlang.



Lançou **F#**, linguagem
funcional para
plataforma .NET.

Referências

AVELINO, Guilherme Amaral. **Tópicos em Linguagens de Programação**: Teresina: UFPI, 2011. Color. Disponível em: [http://www.ufpi.br/subsiteFiles/gaa/arquivos/files/introducao-aula1\(2\).pdf](http://www.ufpi.br/subsiteFiles/gaa/arquivos/files/introducao-aula1(2).pdf)>. Acesso em: 18 mar. 2015.

LIPOVACA, Miran. **Learn You a Haskell for Great Good!: A Beginner's Guide**. San Francisco: no Starch Press, 2011.

TUCKER, Allen B.; NOONAN, Robert E.. **Linguagens de Programação**: Princípios e Paradigmas. 2. ed. Porto Alegre: Amgh, 2010.

YORGEY, Brent. **Haskell Basics**. 2014. Disponível em: <https://www.fpcomplete.com/school/starting-with-haskell/introduction-to-haskell/1-haskell-basics>>. Acesso em: 18 mar. 2015.