

Paradigmas de Linguagens Computacionais

Primeiro Trabalho de PLC

Grupo:

- George Belo Guedes (gbg)
- Mateus de Freitas Leite (mfl3)
- Rodrigo Farias de Figueiredo (rff2)

Tópicos

- Paradigma de linguagem de programação: conceito
- Programação funcional
 - Origens
 - Características
 - Diferença entre programação funcional e imperativa
- Funcional vs Imperativo: vantagens
- Empresas
- Sistemas desenvolvidos

Paradigma de programação: conceito

- Os paradigmas de linguagens de programação podem ser definidos como estilos de programação que se caracterizam por seus conceitos-chave.
- Alguns paradigmas de programação:
 - Imperativo
 - Orientado a objetos
 - Funcional
 - Concorrente.
- Diferem entre si na abstração que proporciona ao programador.

Programação Funcional

Origens

- Lisp: primeira linguagem funcional
 - Desenvolvida por John McCarthy no MIT, ao fim da década de 1950
 - Apesar de não ser puramente funcional, ela introduziu as principais características que são hoje encontradas nas linguagens funcionais mais modernas.
- Posteriormente, surgiram as linguagens Scheme e ML.
- Haskell: lançada no fim da década de 80 com intuito de juntar as ideias provenientes de pesquisas de programação funcional.

Programação Funcional



Características

- Baseia-se na ideia de calcular.
- Os programas são basicamente definições de dados e funções
- Busca evitar estados ou dados mutáveis (*side-effects*).
- A execução consiste em avaliar expressões, onde os subprogramas são vistos como funções que recebem a entrada e retornam a solução.

Programação Funcional



Diferença entre programação funcional e imperativa

- Programação funcional busca evitar *side-effects*; imperativa valoriza as mudanças de estado do programa.
- Diferenças de performance.

Funcional vs Imperativa: vantagens

- Suporte à integração de subprogramas:
 - Funções de alta ordem.
 - Avaliação preguiçosa.
- Linguagens imperativas → Forte dependência da ordem na qual os comandos são executados.
 - Menor relevância em linguagens funcionais.
- Linguagens funcionais evitam *side-effects*, fornecendo assim transparência referencial, o que torna tarefas como verificação de erros e otimização mais fáceis.

Empresas

- Ericsson e Facebook, Motorola, Amazon, Yahoo, T-Mobile, Nortel, Électricité de France, WhatsApp, Rakuten e Bet365 usando Erlang;
- Twitter, usando Scala (parte funcional);
- Microsoft e Intel, usando linguagens da família ML;

Sistemas Desenvolvidos

- Servidores Web, bases de dados distribuídas, chat, servidores para jogos.
- Compiladores, analisadores(verificação de hardware e software).
- Provedores de teoremas, Bioinformática, sistemas de finanças.