

DCC909 – Programação Funcional

AULA 01

Carlos Bruno Oliveira Lopes

Engenheiro de Computação

Mestre em Ciência da Computação

Plano de Ensino

- Objetivo
 - Desenvolver competências no processo de solução de problemas por computador utilizando a ótica do paradigma funcional

Plano de Ensino

EMENTA RESUMIDA

- Conceitos Básicos de Programação Funcional;
- Ferramentas para resolver problemas por computador;
- Tipos de Dados Numéricos;
- Expressões Lógicas;
- Definições Condicionais;
- Tipos Especiais;
- Recursão.

Plano de Ensino

Biblioteca Básica

SÁ, CLAUDIO CESAR DE; SILVA, MÁRCIO FERREIRA DA. **Haskell - Uma Abordagem Prática**. Editora: Novatec, 2006.

RICHARDS, B. **Introduction to Functional Programming Using Haskell**, (2º Edição). Editora: Prentice Hall PTR; 1998. Editora: Prentice Hall PTR.

FETHI, A; LAPALME G. **Algorithms : A Functional Programming Approach (1º Edição)**. 1999, Editora: Addison Wesley.

Biblioteca Complementar:

SIMON, T. **Haskell: The Craft of Functional Programming (2º edição)**. 1999, Editora: Addison Wesley.

Programação Funcional

Linguagem de programação

- Permitem a comunicação de ideias entre pessoas e computadores, mas, em um domínio de expressão reduzido quanto comparado aos da linguagem naturais.
- Elas facilitam a comunicação de ideias computacionais.



Programação Funcional

Paradigmas

- É um padrão de resolução de problemas que se relaciona a um determinado gênero de programas e linguagens.
- Há quatros principais paradigmas de programação:
 - Programação Imperativa
 - Programação Orientada a objeto
 - Programação Funcional
 - Programação Lógica

Programação Funcional

Programação Imperativa

- É o paradigma clássico fundamentado no modelo computacional de “von Neumann-Eckert”.
 - Programas e dados são indistinguíveis pela memória
 - Programa = uma sequência de comandos
 - Estado = valores de todas as variáveis quando o programa executa
 - Abstração procedural são usadas em “programas grandes”
- Ex.:
 - Cobol, Fortran, C, Ada, Perl, ...

Programação Funcional

Programação Imperativa

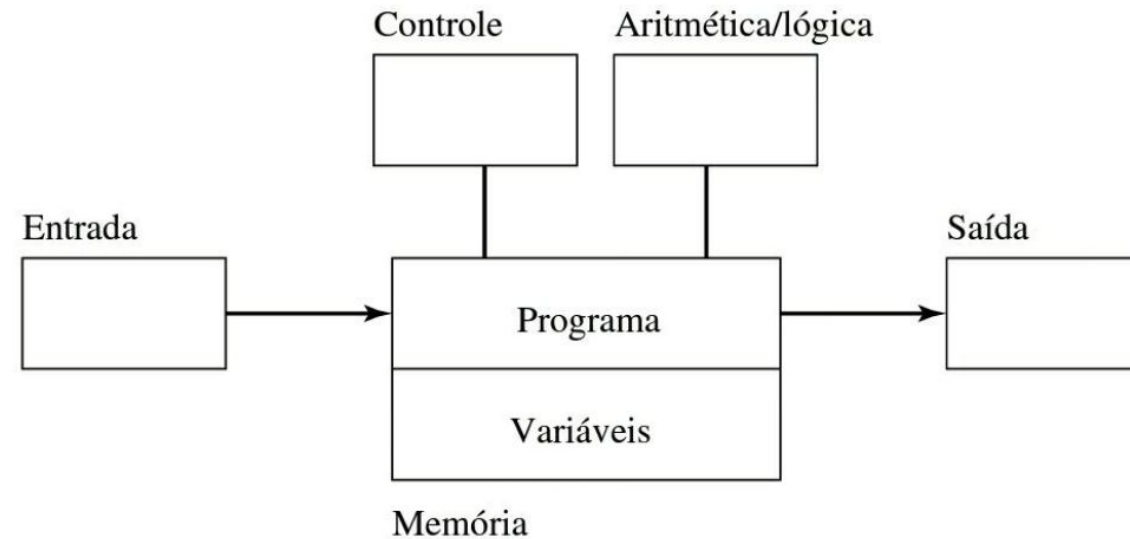


Figura 1 – Modelo computacional de von Neumann-Eckert

Fonte: Tucker. Linguagem de programação

Programação Funcional

Programação Orientada a Objetos (POO)

- Fornece modelo no qual um programa é uma coleção de objetos que interagem entre si, passando mensagens que transformam o seu estado.
- O estudo de orientação a objetos (OO), tem como componentes fundamentais:
 - Envio de mensagens;
 - Herança;
 - Polimorfismos.
- Ex.:
 - Smalltalk, C++, Java, C# e Python

Programação Funcional

Programação Lógica

- Permite a um programa modelar um problema declarando qual resultado o programa deve obter, em vez de como ele deve ser obtido.
- Quando estudamos a programação lógica, é visto:
 - Programas como conjuntos de restrições ou regras do problema;
 - Programas que retornam todas as soluções possíveis;
 - Programas que não determinísticos;
- Ex.:
 - Prolog

Programação Funcional

Programação Funcional

- Modela um problema computacional como uma coleção de funções matemáticas.
 - Entrada = domínio;
 - Saída = intervalo (faixa).
- Linguagem funcionais são caracterizadas por:
 - Composição de funções;
 - Recursão.
- Ex.:
 - Lisp, Scheme, ML, Haskell, ...

Programação Funcional

- Emergiu na década de 1960;
- Surgiu como necessidade dos pesquisadores no desenvolvimento de inteligência artificial e seus subcampos;
- Neste paradigma, a programação é modelada como uma função matemática mapeando entradas e saídas;
- A noção de estado nesse paradigma é dispensável;
- Portanto, o comando de atribuição torna-se desnecessário;
- Apesar de muitas linguagem funcional terem suporte a variável, atribuição e laço; em sua essencial (modelo original) não os faziam parte.

Programação Funcional

Funções e o cálculo Lambda

- Um possível definição de função: $\text{Square}(n) = n * n$
 - Nesta definição temos:
 - O nome da função (identificador);
 - Seus argumentos (entradas);
 - Expressão que defini a função;
 - E seu mapeamento de conjunto; domínio em reais para conjunto de número reais (contradomínio e intervalo): $\text{Square}: \mathbb{R} \rightarrow \mathbb{R}$

Definição: Dizemos que uma função é total se ela é definida para todos os elementos em seu domínio, e parcial em caso contrário;

- Ex.: A função Square é total sobre todo o conjunto de números reais;

Programação Funcional

Funções e o cálculo Lambda

- Em programação imperativa uma variável como x representa uma **localização de memória** e pode ter seus estados atualizados;
 - x representa pode representar um valor **armazenado em memória**;
- Na semântica matemática as variáveis representam expressões reais e são imutáveis.
 - Não há conceito de memória e atualização de valor em memória (célula de memória).
- Linguagens funcionais denominadas **puras** **eliminam o conceito de célula de memória de uma variável**; consequentemente, **elimina o operador de atribuição**.
- Linguagens funcionais denominadas **impuras** são o caso contrário as denominadas puras, ou seja, mantem o operador atribuição e as células de memória.

Programação Funcional

Funções e o cálculo Lambda

- Consequências Programação funcional pura.
 - Não possui conceito de estados;
 - A função depende apenas dos valores de entrada;
 - Não há computação prévia ou avaliação dos argumentos;
- **Transparência referencial**, pois seu valor depende somente dos valores de seus argumentos;

Programação Funcional

Funções e o cálculo Lambda

- A programação funcional é baseada no cálculo lambda desenvolvida por Church;
- A expressão lambda especifica os parâmetros e a definição de uma função sem precisar definir o seu nome (identificador):

$$(\lambda x \cdot x * x) \rightarrow f(x) = x * x$$

- onde, x é o parâmetro usado no corpo sem nome da função $x * x$;

Exemplo de aplicação:

$$((\lambda x \cdot x * x) 2) \text{ resultará em } 4.$$

Programação Funcional

• Funções e o cálculo Lambda

Definição de Church:

1. Qualquer identificador é uma expressão lambda.
2. Se M e N forem expressões lambda, então a aplicação de M a N , escrito como $(M\ N)$, é uma expressão lambda.
3. Uma abstração, escrita como $(\lambda x \cdot M)$, na qual x é um identificador e M é uma expressão lambda, é também uma expressão lambda.

Regras gramaticais BNF:

ExpressãoLambda \rightarrow variable $|$ $(M\ N)$ $|$ $(\lambda\ \text{variable} \cdot M)$

$M \rightarrow$ ExpressãoLambda

$N \rightarrow$ ExpressãoLambda

Programação Funcional

Funções e o cálculo Lambda

Regras gramaticais BNF:

ExpressãoLambda \rightarrow variable $|$ ($M\ N$) $|$ (λ variable $\cdot M$)

$M \rightarrow$ ExpressãoLambda

$N \rightarrow$ ExpressãoLambda

Ex.:

Na expressão $(\lambda x \cdot M)$, a variável x está ligada à subexpressão M .

x

$(\lambda x \cdot x)$

$((\lambda x \cdot x)(\lambda y \cdot y))$