

Paradigmas de Programação (INE5616)

Atividade 1: Visão Geral de Paradigmas de Programação

Ramna Sidharta de Andrade Palma

February 2018

1. Faça um resumo/tabela sobre alguns paradigmas de programação e cite as principais características de cada um deles. Cite também linguagens de cada um dos paradigmas.

| Relação de linguagens e paradigmas | | | |
|------------------------------------|------------|--------|-----------|
| Procedural | Imperativo | Lógico | Funcional |
| C | Ruby | Prolog | Haskell |
| Fortran | Scala | Fril | Elm |
| ALGOL | Swift | Alma-0 | Mercury |
| COBOL | Unix Shell | ROOP | SAC |
| BASIC | BASIC | Curry | Curry |
| Pascal | MATLAB | | |

2. Pesquise sobre outras linguagens de programação existentes e seus respectivos paradigmas.

A seguir um breve comentário sobre algumas das linguagem presentes na tabela acima e que não são amplamente usadas para propósito geral.

Fril: linguagem de paradigma lógico, criada para programação com *cálculo de predicados de primeira ordem*. Esta linguagem inclui a semântica do **Prolog** como subset. A **Fril** possui suporte a Conjuntos Difusos (Fuzzy Sets) e metaprogramação. Ela foi desenvolvida por volta de 1980 na Universidade de Bristol.

Alma-0: sendo multiparadigma, ela suporta o paradigma imperativo, mas possui diversas características inspiradas na programação lógica. É uma linguagem fortemente tipada e de programação declarativa. A **Alma-0** fornece naturalmente suporte a construção alto-nível de árvores de busca.

Mercury: linguagem de programação funcional lógica. Sua primeira versão foi desenvolvida na Universidade de Melbourne, em 1995. É uma linguagem puramente declarativa e de características de tipagem forte, estática e polimórfica. A **Mercury** é baseada no **Prolog** e pode ser vista puramente como um subset daquela linguagem.

**3. O que é o paradigma de programação "orientado a agentes"?
Pesquise sobre ele e cite algumas linguagens.**

Este paradigma iniciou-se na Inteligência Artificial Distribuída, uma subárea da IA que visa problemas cuja complexidade decorre de um caráter de distribuição funcional ou física. A programação é pensada na arquitetura de software em que as entidades computacionais de maior interesse são chamados de "agentes" (ao invés de "objetos" ou "serviços").

A principal característica da programação orientada a agentes é que os agentes são entidades autônomas que têm um objetivo próprio. Esse objetivo é definido pelo projetista, e o agente busca atingi-lo da melhor forma possível. Portanto um agente age de forma independente, em prol de um objetivo atribuído a ele.

Alguns exemplos de linguagens da Orientação a Agentes são: AgentSpeaker, Jason, GOAL e SARL.

4. Leitura do capítulo 1 do livro: AHO, A.V.; SETHI, R. ULLMAN, J.D. Compiladores – Princípios, Técnicas e Ferramentas, Ed. Addison Wesley 2008 / LTC, 1995.