

INE5416 – Paradigmas de Programação

Nome: Rafael Begnini de Castilhos - Matrícula: 20205642

Atividade 01

Questão 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.

PARADIGMA	CARACTERÍSTICAS	LINGUAGENS
Funcional	A solução de um problema é dividida em partes e calculada através de funções que retornam um valor.	Cobol, Fortran, ALGOL
Declarativo	As instruções focam no que deve ser feito e não na forma como deve ser feito, portanto existe uma camada maior de abstração.	Erlang, Prolog, LISP
Imperativo/procedural	Instruções escritas na ordem que devem ser executadas, código pouco legível e difícil de manter, bem eficiente para problemas rápidos e simples.	Haskell, Elixir, LISP
Lógico	Usa lógica simbólica para resolver um problema, partindo de proposições e validando-as para chegar ao resultado.	Mercury, Prolog, LISP, Planner, Ether,
Orientado a objetos	Tenta modelar o problema com entidades chamadas objetos que possuem atributos e métodos. É um padrão usado por grande parte da indústria (e foi criado justamente para suprir a necessidade de resolver problemas rapidamente e de forma padronizada).	Java, C#, Python, C++
Orientado a agentes	Usa o conceito de Agentes vindo da Inteligência Artificial, onde um Agente é um sistema capaz de executar ações de forma autônoma e impactar o resto do sistema.	AgentSpeak

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

Unlambda (Funcional): Linguagem funcional baseada em SKI Calculus com algumas outras funcionalidades modernas.

Brainfuck (Procedural): Linguagem procedural esotérica notada pelo seu extremo minimalismo que possui apenas 8 comandos e não é útil para o uso prático.

Elixir (Funcional): Linguagem de programação funcional, concorrente, de propósito geral que executa na máquina virtual Erlang.

OCaml (Funcional): Objective Caml é uma linguagem que permite dois tipos de compilação, para bytecode que corre numa máquina virtual (zinc) ou para código de máquina nativo para um grande número de plataformas. Ela não é uma linguagem puramente funcional, permitindo a existência de valores mutáveis bem como de efeitos colaterais (side-effects), tipicamente existentes apenas em linguagens imperativas.

Racket (Procedural/Funcional): Linguagem de programação multiparadigma que suporta programação funcional e procedural. Pertence a família de linguagens Scheme/Lisp. Um dos objetivos do projeto é servir como uma plataforma para a criação de linguagem, design e implementação.

Questão 3) O que é o paradigma de programação “orientado a agentes”? Pesquise sobre eles e cite algumas linguagens.

A principal característica desse paradigma está no fato de que agentes são entidades autônomas que têm um objetivo próprio. A arquitetura geral de um agente é também resultado da unificação de várias áreas da IA, e por isso considera que os agentes são entidades que percebem o ambiente onde estão imersos (eles têm sensores - algo equivalente a atributos dinâmicos, isto é, que se auto atualizam) e atuam nesse ambiente para se aproximar de seus objetivos individuais (eles têm atuadores - algo equivalente a métodos, mas que eles mesmo executam).

Esse paradigma ainda não é tão popular no mercado profissional, muito embora seja usado por grandes empresas (o Google é a mais notável) para resolver problemas de complexidade distribuída. Ele também é bastante usado em jogos digitais, para implementar a IA dos NPCs (*Non-Playable Characters*).

Exemplo de sistema de aeroporto: os aviões, as escadas rolantes do aeroporto, as esteiras de bagagem, os caminhões de combustível, etc. Tais agentes se comunicam tentando atingir seus objetivos com a ajuda de outros (o avião solicita abastecimento aos caminhões, e estes negociam com os aviões segundo proximidade, tempo para partida, etc).

Ainda não existe um consenso no uso de linguagem de modelagem para tais aplicações. Muitas linguagens de modelagem para sistemas multi-agentes tem sido desenvolvidas. Entre elas: ADELFE, ADEPT, AUML, CAMLE, DESIRE, FAF, Gaia, INGENIAS, MASCommonKADS, MaSE, MESSAGE, Prometheus e TROPOS. No entanto, nenhuma delas tem sido amplamente adotada.

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

Considere lido.