

Nome: Luiz Afonso Silva Costa

Paradigmas Imperativos

Paradigma Procedural ou Estruturado:

Foi o primeiro paradigma a ser implementado, é baseado na sequência de ordens, ligado diretamente ao tempo verbal imperativo que contém esta ideia. Este paradigma é um dos mais predominantes até hoje. Como conceito temos a “Máquina de Turing”, sendo esta uma abstração para as funções computáveis.

Suas vantagens são:

- Eficiência (modelo, arquitetura, de Von Neumann).
- Modelagem natural de aplicações do mundo real.
- Paradigma muito bem estabelecido.

Suas desvantagens são:

- Implementação excessivamente operacionais.
- Definem “como fazer” e não “o que fazer”.
- Foca em “como realizar a tarefa” ao invés de “o que deve ser feito”.

Paradigma Orientado a Objetos:

O conceito da programação orientada a objetos aproxima-se da “nossa realidade”, ou seja, através da criação de objetos e da interação destes que podem conter atributos diferentes, porém herdando um mesmo método.

Suas vantagens são:

- Sua manutenção é mais fácil, sendo que uma mudança pode alterar as demais partes que utilizam o mesmo objeto.
- O código pode ser reutilizado em outros sistemas, podendo implementar o objeto de um sistema em outro.
- Modelagem natural também, se preocupando mais com a funcionalidade.

Suas desvantagens são:

- Sua execução é mais lenta em relação ao paradigma procedural, estruturado, devido à sua complexidade.
- Seu desempenho pode ser inferior ao paradigma procedural, estruturado, que executa uma linha após a outra e não possui muitos desvios.

Paradigma Funcional

Este paradigma contém sua computação realizada por meio da aplicação de funções a argumentos, diferente dos paradigmas imperativos, as funções geram um resultado que depende apenas dos valores de entradas, no entanto não há “efeitos colaterais”. Seu mapeamento é próximo do modelo matemático de funções.

Suas vantagens são:

- Concorrência é explorada de forma natural.
- Construção rápida de programas.

Suas desvantagens são:

- Implementação geralmente ineficiente.

Paradigma Lógico

O paradigma lógico não indica “o que fazer”, diferente de alguns paradigmas citados anteriormente. É muito utilizado em desenvolvimentos de inteligência artificial, obtendo resultados a partir de avaliações lógicas, utilizando uma base de conhecimento para realizar as inferências.

Suas vantagens são:

- Tem seu destaque em situações que podem ser resolvidas por meio de dedução e inferência.
- Se aproxima do raciocínio humano.

Suas desvantagens são:

- Problemática para expressar algoritmos complexos.

Linguagens de Programação:

Linguagem C

Data de criação: 1972

Paradigmas suportados: Imperativo (Procedural ou Estruturado)

Linguagem Python

Data de criação: 1991

Paradigmas suportados: Imperativo (Procedural ou Estruturado e Orientado a Objetos) e Funcional.

Linguagem Ruby

Data de criação: 1995

Paradigmas suportados: Multiparadigma

Linguagem Java

Data de criação: 1995

Paradigmas suportados: Imperativo (Orientado a Objetos)

Linguagem C++

Data de criação: 1979

Paradigmas suportados: Imperativo (Procedural ou Estruturado e Orientado a Objetos)

Linguagem C#

Data de criação: 2000

Paradigmas suportados: Imperativo (Procedural ou Estruturado e Orientado a Objetos) e Funcional

Linguagem PHP

Data de criação: 1995

Paradigmas suportados: Imperativo (Procedural ou Estruturado e Orientado a Objetos) e Funcional

Linguagem SQL

Data de criação: 1974

Paradigmas suportados: Multiparadigma

Linguagem Go

Data de criação: 2009

Paradigmas suportados: Imperativo (Procedural ou Estruturado e Orientado a Objetos) e concorrente

Linguagem Swift

Data de criação: 2014

Paradigmas suportados: Imperativo (Procedural ou Estruturado e Orientado a Objetos) e Funcional

Linguagem Prolog

Data de criação: 1972

Paradigmas suportados: Lógico e declarativo

Referências

Slides da “Aula 02 - Tipos de Linguagens - Paradigmas de Programação”

<https://www.tiobe.com/tiobe-index/>

<https://cadernoscomp.com.br/linguagens-de-programacao/>

<https://medium.com/@sergiocosta/paradigma-funcional-3194924a8d20>

<https://apexensino.com.br/o-que-e-programacao-orientada-objetos/>

https://fit.faccat.br/~guto/artigos/Artigo_Paradigmas_de_Programacao.pdf

