

Linguagens de Programação

OBJETIVO

Apresentar os paradigmas de programação existentes, incluindo o conceito em si. As linguagens de programação são baseadas em um ou mais paradigmas, sendo o último caso conhecido como linguagem multiparadigma.

Introdução à Linguagens de Programação

Por que falar de história da computação?

- Compreender as dificuldades enfrentadas;
- Fundamentos da computação;
- O processo de pensamento.

Problemas Computacionais:

Objeto de discussão que possui instrução passo a passo que são mais facilmente resolvíveis em ambiente computacional.

Problemas de Decisão (Problema Decidível):

- Carácter Lógico - Sim ou Não
- Ideia
 - Pertencimento

Exemplo: “dado um número n inteiro positivo, determine se n é primo. ”

Problemas de Busca:

- Relacionamento binário
- Objetivo
 - Semelhante ao nome
 - X está em A ?
- Recorrente em teoria de grafos.

Exemplo: Clique.

Problemas de Otimização:

- Objetivo
 - Maximizar, ou
 - Minimizar uma função
- Aplicações

O que é uma linguagem de programação?

Método padronizado composto por um conjunto de regras sintáticas e semânticas de implementação de código-fonte.

Como um computador entende o programa?

Um programa é um amontoado de palavras senão for possível que o computador entenda. Para isso existe um processo de tradução.

Processo de Tradução/Compilação:

- **Tradução:**
 1. Geração do programa objeto.
 2. Execução do programa objeto.
- Execução mais rápida e Programas menores

Exemplo: C++ e Java

- **Interpretação:**
Programa fonte executado diretamente.
 - Maior Flexibilidade, porém, deixa o sistema mais lentos.

Exemplos: JavaScript, Python e Ruby.

Características de um programa

LEGIBILIDADE

“Estado que é legível”

- Facilidade de leitura
- Compreensão
- Ortogonalidade
 - Coerência nas instruções
- Definições adequação das estruturas.

REDIGIBILIDADE

“Facilidade de escrita de código”

- Pode conflitar com a legibilidade
- Ortogonalidade
 - Coerência nas instruções
- Simplicidade de escrita
- Suporte à abstração
- Reuso do código
- Expressividade

CONFIABILIDADE

“Faz o que foi programado para fazer”

- Possui:
 - Verificação de tipos
 - Trata exceções
 - Uso de ponteiros
 - Compatibilidade entre compiladores

CUSTO

“Análise de impacto”

- Possui:
 - Treinamento
 - Codificação
 - Compilação
 - Execução
 - Infraestrutura

Análises de código

Quando vai para o Compilador:

- Lexical Analyzer
- Syntax Analyzer
- Semantic Analyzer

Análise Léxica

- Particionar
- Classificar
- Eliminar

Padrão - gramática:

- Depende da linguagem de programação utilizada.

Erro de semântica:

- Não faz o que é esperado.

Paradigmas de programação

DEFINIÇÃO FORMAL - PARADIGMA

“Forma de resolução de problemas com diretrizes e limitações específicas de cada paradigma utilizando linguagem de programação.”

- **Procedural:** Chamadas sucessivas e procedimentos separados.
Exemplos: Fortran e Lua
- **Funcional:** Instruções são baseadas em funções.
Exemplos: Lua, JavaScript, Python e Java
- **Estruturado:** Estruturado de blocos aninhados.
Exemplos: C++, JavaScript, Java e C
- **Computação Distribuída:** Funções executadas de forma independente.
Exemplos: Ada
- **Lógico**
Exemplos: Java
- Mais usados:
 - **Orientação à Objeto**
Exemplos: Python, Java, Lua e C++
 - **Estruturado**
Exemplos: C++, JavaScript, Java e C

Paradigma Estruturado (Linguagem C)

“Ênfase em sequência”

Conceito:

- **Sequência**
- **Decisão**
 - Teste Lógico
- **Iteração**
 - Funções, Laços, Condições.

Utilização:

- Problemas simples e diretos
- Aprender programação

POO ainda não compreendida por muitos.

Orientação à Objeto

Paradigma de programação baseado na utilização de objetos e suas interações.

- Análogo ao mundo real.

O que é um objeto?

“Um objeto é descrito por características específicas, comportamentos e estado. ”

- O que tenho
- Sou capaz de fazer
- Como faço