Introdução a linguagem de programação.

# Introdução

# História da computação

Rever a aula e anotar as história da computação.

# Linguagens

Em 1949, surgiu a primeira linguagem de programação (máquina) Assembly.

Anos 50:

* Cobol.
* Fortran.

Entre 60 e 70:

* C.
* Prolog.
* ML.

Déc. 90, começou a surgiu as linguagens de alto nível:

* C#.
* Java.
* Ruby.
* Python.
* JS.

2000:

* TS.
* Rust.
* GO.

Definição de linguagem de programação:

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

Como um computador entende o programa?

O código fonte é traduzido ou interpretado.

Processo de Compilação:

* Linguagem de alto nível (Programa Fonte) > Compilador > Assembly (Programa Objeto) > Linguagem de máquina.

A linguagem de alto nível é enviada para o compilador que vai executar a análise do programa. E esse compilador vai traduzir o programa de alto nível em um código de baixo nível (Programa Objeto).

Tradução:

* Geração do programa objeto.
* Execução do programa objeto.
* Benefícios:
  + Execução mais rápida.
  + Programas menores.
* Ex:
  + C#.
  + Java.

Interpretação:

* Programa fonte executado diretamente.
* Benefícios:
  + Maior flexibilidade.
* Ex.:
  + Ruby.
  + JS.
  + Python.

Processo de tradução:

Programa fonte > Compilador > Análise > Programa Objeto.

O programa fonte é enviado para o compilador e dentro do compilador é iniciado um conjunto de Análise: Análise Léxica, Análise sintaxe, Análise semântica. E no final o compilador vai gerar o código (Programa objeto).

Característica de um Programa

# Desenvolvimento de programas

* Legibilidade:
  + Facilidade de leitura.
  + Compreensão.
  + Ortogonalidade (Coerência nas instruções).
  + Definição adequada das estruturas.
* Redigibilidade: Facilidade de escrita de código.
  + Pode conflitar com a legibilidade
  + Ortogonalidade (Coerência nas instruções).
  + Simplicidade da escrita.
  + Suporte à abstração.
  + Reuso do código.
  + Expressividade.
* Confiabilidade: Faz o que foi programado para fazer.
  + Verificação de tipos.
  + Trata exceções.
  + Uso de ponteiros.
  + Compatibilidade entre compiladores.
* Custo: Análise de impacto
  + Treinamento.
  + Codificação.
  + Compilação.
  + Execução.
  + Infraestrutura.

Análise de Código

# Análise Léxica (Scanner ou leitura).

* É a primeira fase de um processo de compilação e a função dela é fazer a leitura do programa fonte, caracter por caracter, agrupar e produzir uma sequência de símbolos léxicos (Tokens).

Particionar > Classificar > Eliminar.

Os tokens: São identificadores, palavras reservadas, números, strings.

* Eliminar caracteres de espaços em branco, comentários etc.

# Análise Sintática

Sintaxe (Forma): Componente do sistema linguístico que interliga os constituintes da sentença, atribuindo-lhe uma estrutura.

Padrão: Depende da linguagem de programação utilizada.

# Análise semântica

Semântica (Significado): É o estudo do significado. Incide sobre a relação entre significante, como; palavras, frases, sinais e símbolos.

Erro de semântica: Não faz o que é esperado. Está relacionado com a lógica do programa.

Paradigmas de programação

# Paradigmas

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

## Classificação:

* Orientação à objeto.
* Ex.:
  + Python.
  + Lua.
  + C++.
  + Java.
* Procedural.
  + Chamadas sucessivas e procedimentos separados.
* Ex.:
  + Fortran.
  + Lua.
* Funcional.
  + Instruções são baseadas em funções.
* Ex.:
  + Python.
  + Lua.
  + JS.
* Estruturado.
  + Estrutura de blocos aninhados.
* Ex.:
  + JS.
  + Java.
  + C++.
  + C
* Computacional distribuída.
  + Funções executadas de forma independente.
* Ex.:
  + Ada.
* Lógico.
* Ex.:
  + Java.

# Paradigma Estruturado

Conceito: Ênfase em sequência.

* Sequência.
* Decisão (Teste Lógico).
* Iteração (Funções, Laços e condições).

Utilização:

* Problemas simples e diretos.
* Aprender programação.

# Orientação à Objeto

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

Objeto: É descrito por características específicas, comportamentos e estado.

Modelo (Classe), Atributos, Comportamento (métodos) e estado.

Poo - Classe:

* Alocação em memória.
* Operações associadas.
* Métodos associados a um objeto.

Estruturada - Variável:

* Alocação em memória.
* Operações associadas.

# Pilares de Orientação a Objeto

* Herança.
* Encapsulamento.
* Polimorfismo.
* Abstração.