

Maurício Dourado  
mauriciodourado@ifce.edu.br



# Paradigmas de Programação



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
CEARÁ

## Aula 01 - Apresentação



## **Objetivos**

---

- Estudos dos principais conceitos que permeiam as linguagens de programação nos diferentes paradigmas.
- Ao final da disciplina, o aluno deve ser capaz de analisar linguagens de programação sob diversos aspectos e escolher aquela que mais se adequa à implementação de um projeto.





# Conteúdo Programático

---

## Introdução

- Motivos para estudar os conceitos de linguagens de
- Programação;
- Fatores de influência no projeto de linguagens de programação;
- Critérios de avaliação de linguagens de programação;
- Evolução das linguagens de programação;
- Paradigmas de linguagens de programação;
- Métodos de implementação de linguagens de programação;
- Ambientes de programação.





# Ementa

---

## Variáveis

- Conceito de variável
- Conceito de vinculação
- Vinculação de armazenamento
- Tempo de vida de variáveis
- Escopo de identificadores
- Ambientes de referenciamento
- Vinculação de valores
- Inicialização de variáveis
- Constantes





# Conteúdo Programático

---

## **Tipos de Dados**

- Conceito de tipos de dados
- Tipos de dados primitivos
- Tipos de dados compostos
- Vinculação de tipos
- Inferência de tipos
- Verificação de tipos
- Tipificação forte
- Compatibilidade de tipos
- Conversão de tipos





# Conteúdo Programático

---

## **Expressões e Comandos**

- Expressões simples
- Expressões compostas
- Expressões com efeitos colaterais
- Avaliação curto-circuito
- Sobrecarga
- Instruções de atribuição
- Estruturas de controle
- Sequenciadores
- Saltos
- Escapes
- Exceções





# Conteúdo Programático

---

## **Abstrações**

- Conceito de abstração
- Abstrações de processos
- Procedimentos e funções
- Ambientes de referências locais
- Métodos de passagem de parâmetros
- Ordem de avaliação de parâmetros
- Polimorfismo

## **Tipos abstratos de dados**

- Abstração
- Modularidade
- Encapsulamento
- Ocultamento de informações
- Tipos abstratos de dados





## **Bibliografia**

---

- SEBESTA, R. W. **Conceitos de linguagens de programação**. 5. ed. Porto Alegre: Bookman , 2003
- MELO, A. C. V. de; SILVA, F. S. C. da. **Princípios de linguagens de programação** . São Paulo: Edgard Blücher , 2003

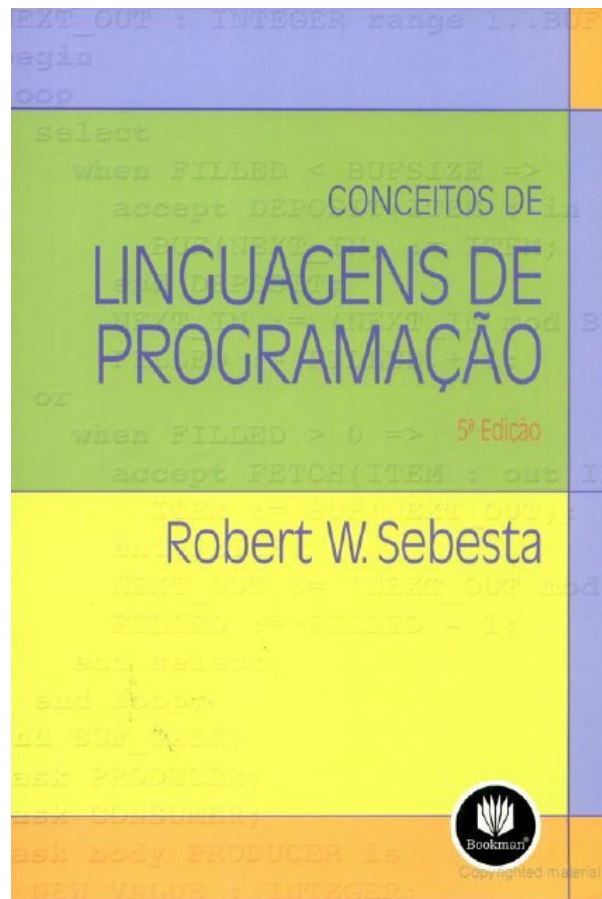






# Livro Adotado

---





# Avaliações

---





# O que é um paradigma?

---





## O que é um paradigma?

- Paradigma, é uma palavra de origem grega, que significa **modelo**.
  - Representação de um padrão a ser seguido.
  - Referência inicial como base de modelo para estudos e pesquisas.





## O que é um paradigma?

- Uma paradigma é, basicamente, um modelo que interpreta certa realidade.
- Ele possibilita a organização de ideias com o objetivo de entender essa realidade.





# **Paradigma de Programação**

- Determinação de uma visão que um programador possui sobre a estruturação e execução de um programa.





# **Tipos de Paradigmas de Programação**

- Imperativo
- Estruturado
- Orientado a Objetos
- De Passagem de Mensagens
- Orientado a Aspecto
- Funcional
- Restritiva
- Lógica





# História

---

- Os computadores eram inicialmente programados através de códigos binários;







## História

---

- Os computadores eram inicialmente programados através de códigos binários;
  - **Problema:** Mais propenso a erros.





## História

---

- Os computadores eram inicialmente programados através de códigos binários;
  - **Problema:** Mais propenso a erros.
- Linguagens de Montagem:
  - Substituições de funções por mneumônicos
  - Endereços de memória por identificadores.





# Linguagens de Montagens

- Algumas linguagens chegaram a suportar:
  - Programação modular;
  - variáveis externas;
  - seções comuns (globais);
  - reuso de código.
- Por que mudar?





# **Paradigma Estruturado**

---

- Descreve os passos;
- Fluxo é evidente;
- Não são admitidos saltos incondicionais;
- Eficácia e Eficiência depende do programador.
  
- Vantagens:
  - Modularização;
  - Reutilização do código;
  - Organização e facilidade de entendimento por um desenvolvedor ou equipe;
  - Manutenção facilitada e agilizada.
  
- Desvantagens:
  - Lógica menos óbvia (desvio dentro de loop)





# **Domínios da Programação**

---

- Aplicações Científicas
  - Qual a preocupação Principal?
  - Qual foi a primeira Linguagem?





# **Domínios da Programação**

---

- Aplicações Científicas
  - Qual a preocupação Principal?
    - Desempenho
  - Qual foi a primeira Linguagem?
    - Fortran





# Fortran

---

- IBM Mathematical **FOR**mula **TRAN**slation System
- Metade da década de 50 (IBM)
- Criada para o IBM 704
- Primeira Linguagem de alto nível amplamente utilizada e Primeira Imperativa
- Versões mais recentes suportam Orientação a Objetos





## **Domínios da Programação**

---

- Aplicações Comerciais
  - Qual a preocupação Principal?
  - Qual foi a primeira Linguagem?







# **Domínios da Programação**

---

- Aplicações Comerciais
  - Qual a preocupação Principal?
    - Relatórios Elaborados
  - Qual foi a primeira Linguagem?
    - Cobol





# COBOL

---

- Commom Business Oriented Language
- 1959 - Comissão CODASYL
- Propósito Comercial
- Rigidamente Padronizada
- A linguagem mais usada da história
- Utilizada nos dia de hoje (suporte a Orietação a Objetos)
- Características:
  - Acesso rápido a arquivos e bases de dados
  - Atualização rápida de arquivos e bases de dados
  - Geração de uma grande quantidade de informações
  - Saída com um formato compreensível ao usuário





# **Domínios da Programação**

---

- Inteligência Artificial
  - Qual a preocupação Principal?
  - Qual foi a primeira Linguagem?





# **Domínios da Programação**

---

- Inteligência Artificial
  - Qual a preocupação Principal?
    - Flexibilidade
  - Qual foi a primeira Linguagem?
    - Lisp





# LISP

---

- **LIST** Processing
- A lista é a estrutura de dados fundamental desta linguagem.
- Tanto os dados como o programa são representados como listas, o que permite que a linguagem manipule o código fonte como qualquer outro tipo de dados.
- Tipos de dados:
  - Átomo
  - lista.





## **Paradigma Imperativo**

- Todas as linguagens apresentadas seguem o paradigma imperativo.
- Mas o que é Paradigma Imperativo?





## Paradigma Imperativo

- Descreve a computação como ações, enunciados ou comandos que mudam o estado (variáveis) de um programa.
- E existe maneira de resolver o problema, computacionalmente, sem descrever os passos?





## **Paradigma Declarativo**

- Descreve-se o problema ao computador e não como resolvê-lo.
- O programa é estruturado como uma coleção de propriedades para encontrar o resultado esperado, e não um procedimento a se seguir.
- Exemplo?







## **Paradigma Declarativo**

- Descreve-se o problema ao computador e não como resolvê-lo.
- O programa é estruturado como uma coleção de propriedades para encontrar o resultado esperado, e não um procedimento a se seguir.
- Exemplo?
  - SQL





## **Mais Paradigmas**

---

- Existem ainda as famílias das linguagem lógicas e funcionais.
- O Que são elas? Exemplos?





# Mais Paradigmas

---

- Funcionais
  - Erlang (aplicações concorrentes)
  - R (estatística)
  - K (análise financeira)
- Lógicas
  - Planner
  - Prolog





# Paradigma x Linguagem de Programação

- Diferença entre Paradigmas e Linguagens de Programação:
  - Um paradigma é um estilo de programação;
  - linguagem é um método para expressar instruções ao computador.





## **Exercícios**

---

- 1) O que é uma linguagem imperativa ?
- 2) Qual a melhor linguagem de programação ?
- 3) Por que usar o Paradigma Estruturado?
- 4) O bom programador é definido como aquele que programa em várias linguagens ?
- 5) Por que a linguagem COBOL ainda hoje é muito utilizada ?
- 6) Se a linguagem BASIC não foi aceita no meio científico, por que foi amplamente utilizada ?
- 7) Qual o propósito da criação da linguagem PASCAL ?

