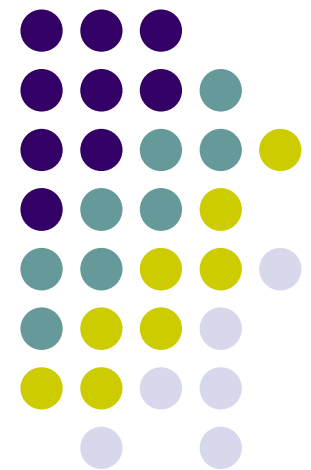


# Paradigmas de Linguagens de Programação

---

## 1. Introdução e Conceitos Básicos

Prof. Márcio D. Puntel  
Marcio.puntel@ulbra.edu.br

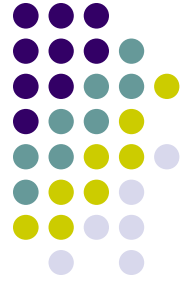




# O Papel de LPs no PDS

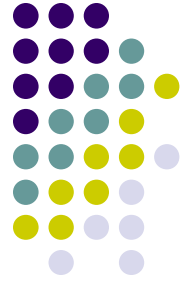
- O objetivo de LPs é tornar mais efetivo o Processo de Desenvolvimento de Software (PDS)
- PDS visa geração e manutenção de softwares de modo produtivo e com garantias de padrões de qualidade

# Propriedades desejadas em um Software:



- Confiabilidade
- Manutenibilidade
- Eficiência
- Desempenho

# Propriedades desejáveis em LPs



- Legibilidade
    - Marcadores de Blocos
- ```
if (x>1)
    if (x==2)
        x=3;
    else x=4;
```
- Desvios Incondicionais (goto)

# Propriedades desejáveis em LPs



- Facilidade de Aprendizado
  - Excesso de Características é Prejudicial

`c = c + 1;`

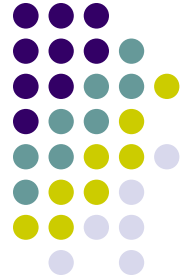
`c+=1;`

`c++;`

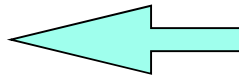
`++c;`

- Eficiência
  - Verificação Dinâmica de Tipos
- Confiabilidade
- Tratamento de Exceções

# Propriedades desejáveis em LPs



- Reusabilidade



**Um dos principais benefícios do  
Paradigma Orientado a Objetos**

- Redigibilidade

- Tipos de Dados Limitados (FORTRAN)
- Ausência de Tratamento de Exceções

- Portabilidade

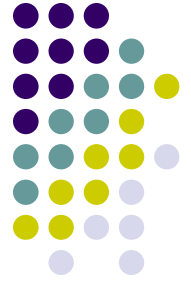
- Rigor no Projeto
- Pode Contrastar com Eficiência

# Conceitos Básicos



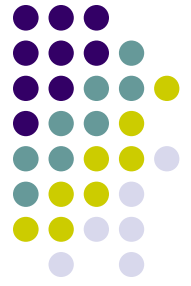
- Paradigma
- Léxico
- Sintaxe
- Semântica

# Paradigma



- Conjunto de regras que estabelecem fronteiras (domínios) e descrevem como resolver os problemas dentro destas fronteiras.
- Exemplo ou modelo que servem de norma.
- Forma de representar e manipular o conhecimento.





# Léxico e Sintaxe

- **Léxico:** é o conjunto de palavras que compõem a linguagem (também chamadas de **palavras reservadas**)
- **Sintaxe:** em LPs é a forma de suas expressões, de suas instruções e de suas unidades de programas.



# Semântica

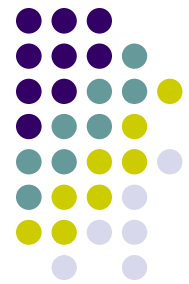
- Em LPs é o significado das expressões, instruções e definição de unidades de programas:

- Exemplo: a sintaxe do IF no Pascal é:

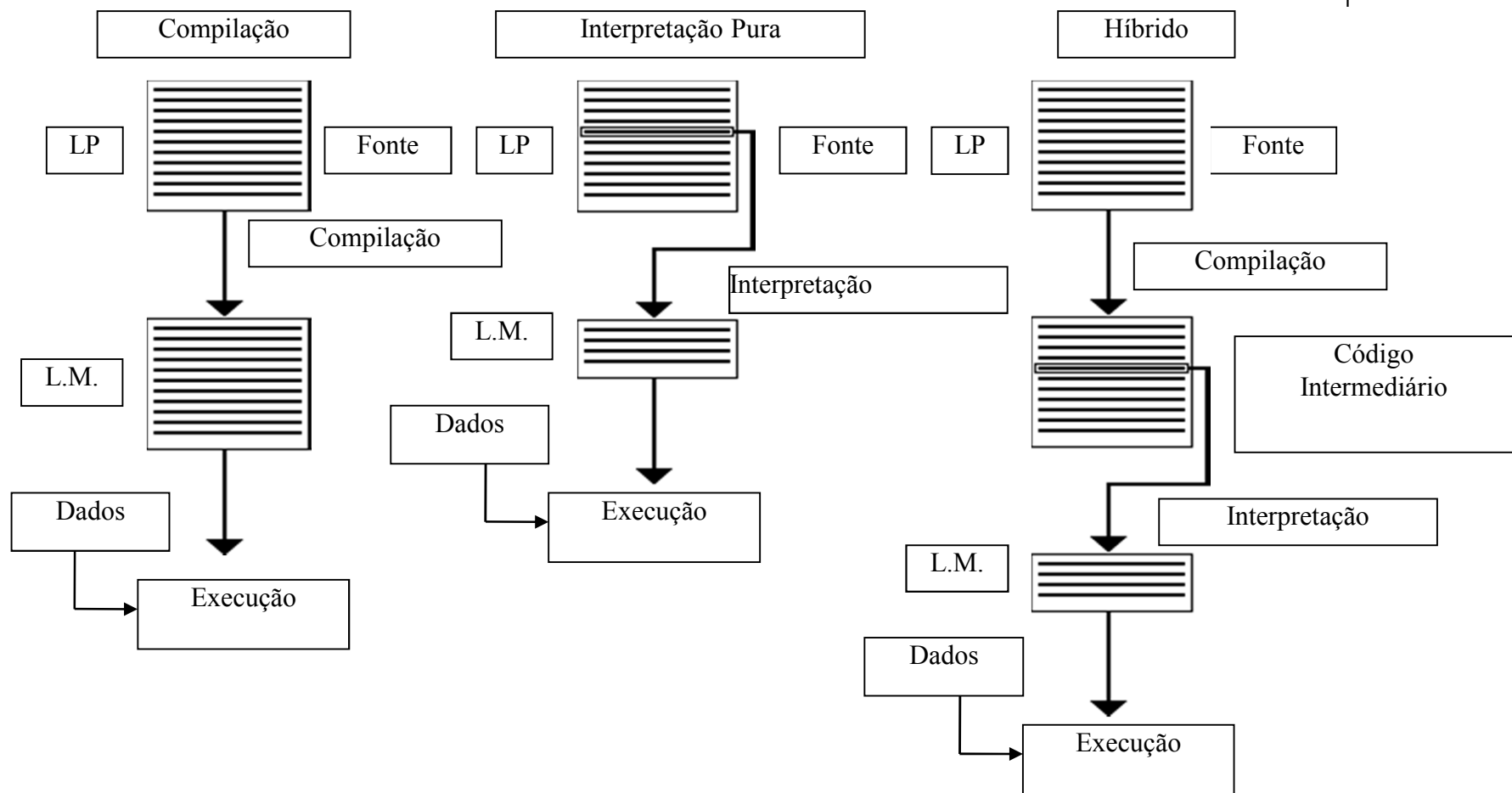
IF <expressão condicional> THEN  
<instrução>

- Significado: se o resultado da expressão condicional for verdadeiro, executa a instrução, do contrário não.

***Necessidade de Padronização (ISO, IEEE, ANSI)***



# Implementação de LPs

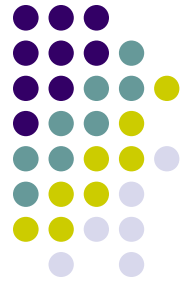




# Implementação de LPs

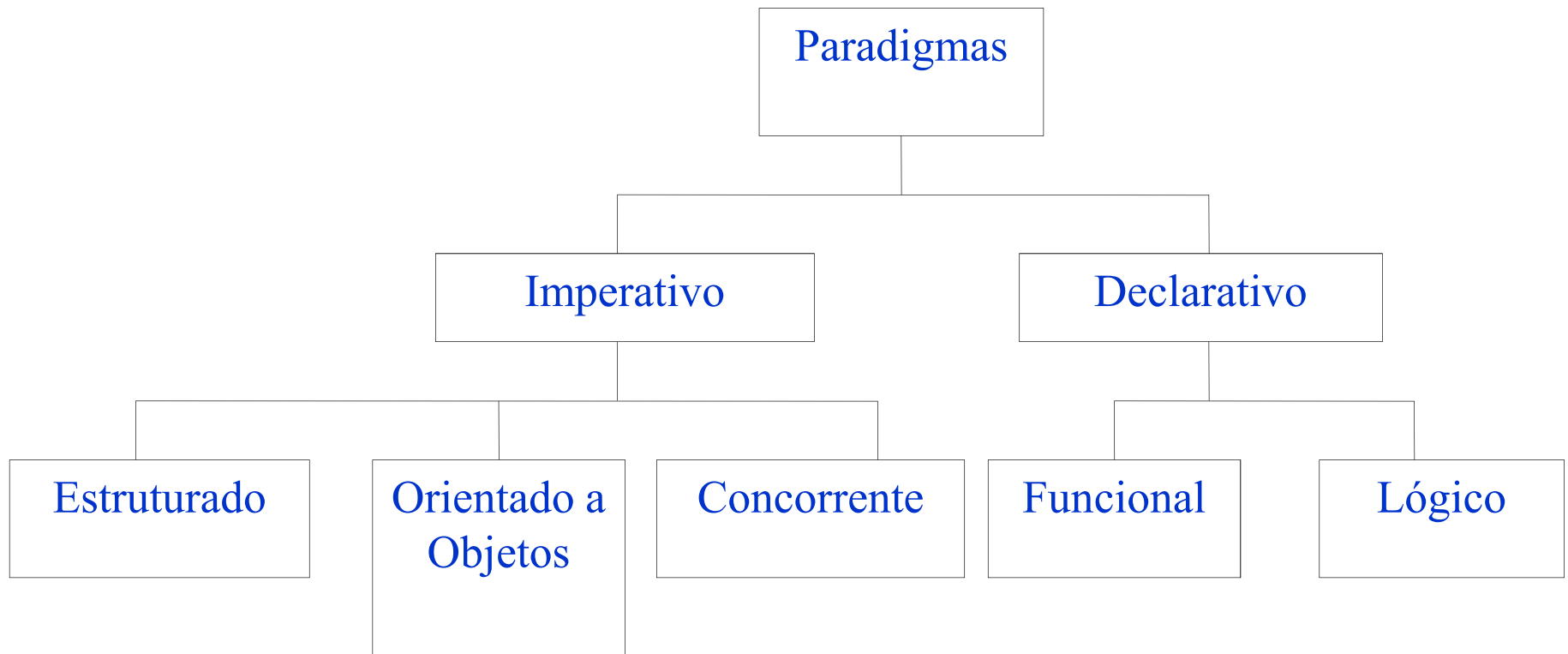
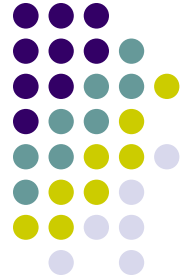
- Compilação
  - Eficiência
  - Problemas com Portabilidade e Depuração
- Interpretação Pura
  - Flexibilidade, Portabilidade e Facilidade de Depuração
  - Problemas com Eficiência
- Híbrido
  - Une Vantagens dos Outros Métodos
  - JVM

# Pense...



- Se as linguagens interpretadas são menos eficientes que as linguagens compiladas, por que elas foram criadas e ainda existem?

# Paradigmas de LPs





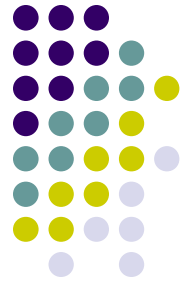
# Paradigmas de LPs

- **Imperativo**

- Processo de Mudanças de Estados
- Variável, Valor e Atribuição
- Células de Memória

- **Declarativo**

- Especificações sobre a Tarefa a Ser Realizada
- Abstrai-se de Como o Computador é Implementado



# Paradigmas de LPs: Imperativo

- **Estruturado**

- Refinamentos Sucessivos
- Blocos Aninhados de Comandos
- Desestímulo ao uso de desvio incondicional

- **Orientado a Objetos**

- Abstração de Dados

- **Concorrente**

- Processos Executam Simultaneamente e Concorrem por Recursos



# Paradigmas de LPs: Estruturado



- **Funcional**

- Programa Composto por Funções Matemáticas

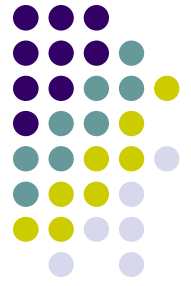
- **Lógico**

- Predicados
- Dedução Automática



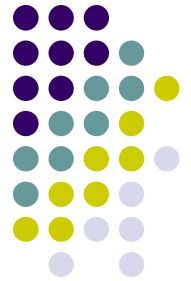
# Histórico: Evolução de LPs

- Dificuldade de Programação em Linguagens de Máquina
- Foco de Primeiras LPs era Eficiência de Processamento e Consumo de Memória
- Baixa Produtividade de Programação
  - Programação Estruturada
  - Tipos Abstratos de Dados
  - Orientação a Objetos



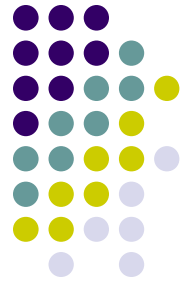
# Histórico: Origem de LPs

- FORTRAN (1954)
  - aplicações numéricas
- LISP (1958)
  - programação funcional
- ALGOL (1958)
  - programação estruturada
- COBOL (1959)
  - aplicações comerciais



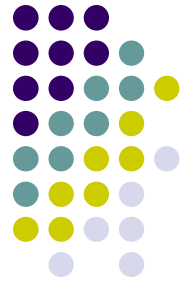
# Histórico: Origem de LPs

- BASIC (1964)
  - ensino para leigos
- PASCAL (1970)
  - ensino de programação estruturada
  - simplicidade
- PROLOG (1970)
  - programação lógica
- C (1971)
  - implementação de UNIX



# Histórico: Origem de LPs

- SMALLTALK (1969)
  - programação orientada a objetos
- ADA (1983)
  - programação concorrente
- C++ (1983)
  - disseminação da programação orientada a objetos
- Delphi (1995)
  - POO e POE



# Histórico: Origem de LPs

- JAVA (1995)
  - mais simples e confiável que C++
  - Internet
- PHP (1995)
  - Internet
- JavaScript
  - Internet
- ASP.NET (2000)
  - Internet

# Referências



- Material cedido de Prof. Fernando Prass  
(fprass@gmail.com)