Modelos conceituais

Paradigmas de Programação
(CI1062)

Prof. Marcos Didonet

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Contexto: paradigma de programação

- Fornece e determina visão que o programador possui sobre estruturação e execução de um programa
  - Porque programar usando o paradigma estruturado ? Ou outros ? Orientação a objetos ? (que será visto na disciplina)
- Um paradigma não é MELHOR que outro, apenas mais adaptado para resolver determinado tipo de problema
  - Driver de impressora
  - Cálculos matemáticos
  - Busca e processamento de Dados
  - Inteligência artificial
  - Sistemas web
  - Sistemas de gestão
  - Etc

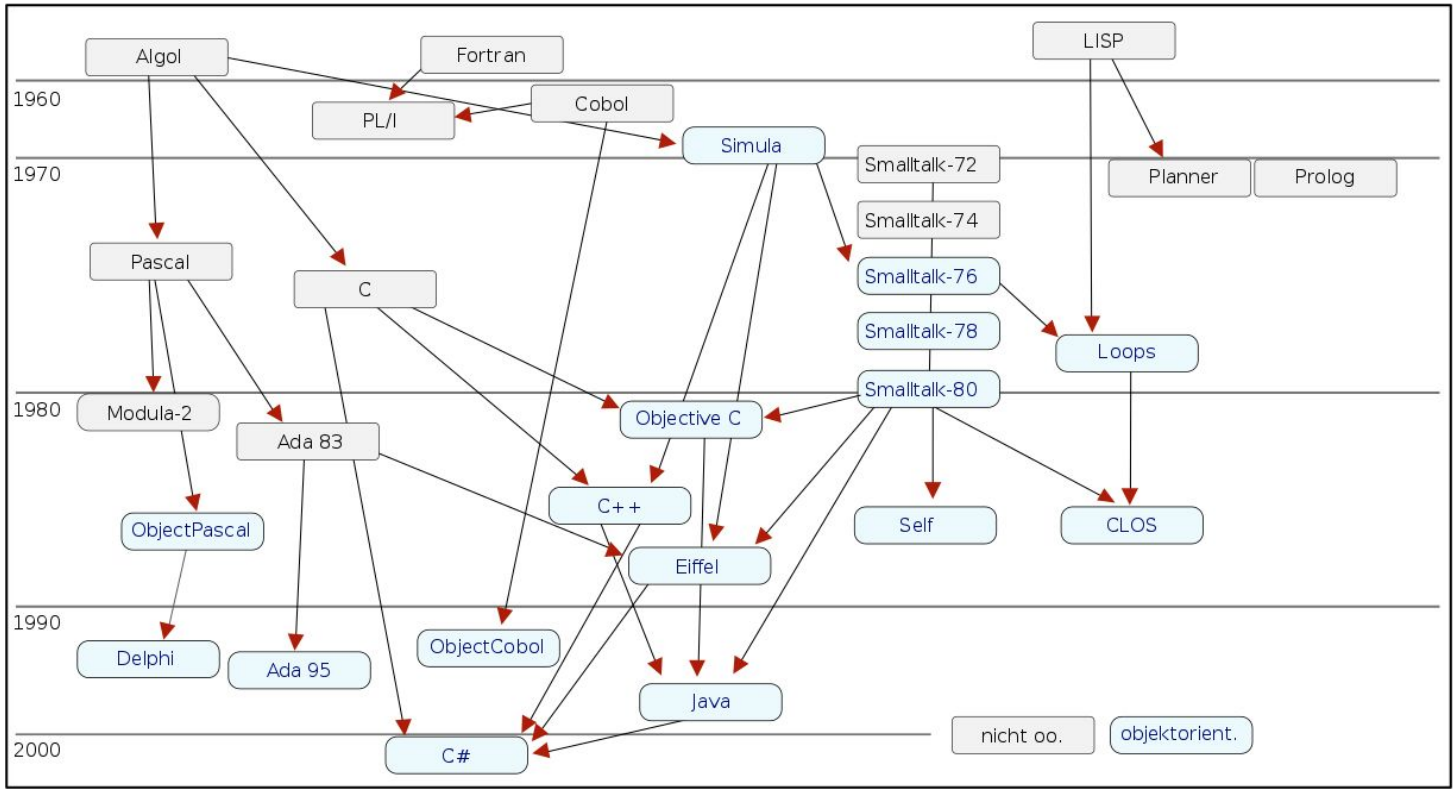
Paradigmas principais

- Imperativo: sequência de instruções que manipulam variáveis; (entrada; processamento;saída)
  - Não estruturado: Cobol, Assembly
  - Estruturado
    - Procedural/procedimental: exemplos: Pascal, C
    - **Orientada a objetos:** exemplos: **Java**, C#
    - Orientada a eventos: exemplos: Javascript
- Declarativo
  - **Funcionalista/Funcional:** exemplos: **Haskell**, ML, Erlang
- Híbridas
  - Java, Scala, Pascal, Javascript (além do paradigma PRINCIPAL)

Paradigma orientado a objetos

- **Objetivo**: eliminar o “gap semântico” que acontece ao modelar o mundo real e traduzir para um conjunto de instruções em linguagens de programação estruturada.
  - Simplificando: variáveis / procedimentos X objetos + comportamento

Algumas linguagens orientadas a objeto



Conceitos centrais em POO

- Abstração do mundo real
  - modelagem de entidades
- Fluxo de programa
  - Baseado em estados (objetos em memória com valores modificados)

Abstração/representação

- Observar a realidade (domínio) e procurar capturar sua estrutura (abstrair entidades, ações, relacionamento, etc, que forem consideradas relevantes para a descrição deste domínio).
- **Abstração**: características principais para diminuir a complexidade
  - Dependente do contexto e do ponto de vista do problema
  - Apenas o que for necessário
  - Exemplo: sistema acadêmico : o que é relevante ?
- **Representação:** definir um modelo conceitual com alguma linguagem (textual ou gráfica), capturando de maneira concreta as abstrações

Modelos conceituais

1. Classificação/Instanciação
2. Generalização/Especialização
3. Agregação(composição)/Decomposição
4. Associação

   https://app.diagrams.net/
   Ferramentas para criação de modelos conceituais:
   https://modeling-languages.com/text-uml-tools-complete-list/

Uma extensão

5. Comportamento (modelo “vivo”)
   a. Fluxo de programa baseado em estados