

Resumo (POO)

• Introdução

A programação orientada a objetos surgiu na década de 1960, mas se tornou popular nos anos 1980 com a linguagem Smalltalk e a disseminação de conferências e publicações. Esse paradigma revolucionou o desenvolvimento de software ao introduzir um modelo baseado em objetos, permitindo a criação de sistemas escaláveis e organizados.

• Porque é popular?

- Escalabilidade: Permite desenvolver pequenos programas até grandes sistemas
- Abstração: Representa a programação do modo como lidamos com problemas do vida real
- Bibliotecas e frameworks:
- Gerenciamento de complexidade: Ajuda a estruturar melhor trabalhos grandes

• Relação entre linguagem e pensamento

- A forma como pensamos e resolvemos problemas é influenciada pela linguagem que utilizamos
 - Linguagem de programação moldam a maneira como os programadores estruturam seus algoritmos
 - Um exemplo clássico é a comparação entre FORTRAN e APL, onde a abordagem natural de APL resultou numa solução mais eficiente para um problema de análise de DNA.
- POO como um novo paradigma

POO é um paradigma de programação, ou seja, um modelo para resolver problemas computacionais. Ele organiza sistemas como uma comunidade de objetos que interagem para executar tarefas. Diferente da abordagem tradicional, onde dados são manipulados por procedimentos extensos, na POO o objeto possui seus próprios dados e métodos para operar sobre eles.

• Conceitos fundamentais da (POO)

- Objetos e classes
- Um objeto representa uma entidade do mundo real e contém dados e métodos para manipulá-los. Todos os objetos pertencem a uma classe, que define suas características e comportamentos
- Mensagens e métodos
- Objetos se comunicam enviando mensagens que invocam métodos de outros objetos
- Essa interação define como os objetos realizam ações no programa.
- Encapsulamento
- Os detalhes internos de um objeto são escondidos, permitindo que apenas métodos específicos acessem seus dados. Isso reduz dependências e melhora a modularidade do código.

4- Herança

- Permite que classes herdem características e comportamentos de outras classes
- Facilita a reutilização de códigos e a criação de hierarquias organizadas.

5- Polimorfismo

- Objetos de diferentes classes podem responder às mesmas mensagens de formas distintas
- Permitindo flexibilidade e reutilização de código.

Morale da história

2211257