

①

## Resumo 3

②

Thinking // Object // OrientedHistória da POO:

→ Surgiu na década de 1960, popularizou-se nos anos 1980 com linguagens como SmallTalk e C++

Popularidade:

→ POO escala bem para projetos problemas complexos, ajuda a resolver a "crise do software" e promove reutilização de código

Impacto da linguagem no pensamento:

- Influência do idioma na percepção do mundo (exemplo das aranhas e a neve)
- Programadores tendem a resolver problemas de acordo com as facilidades da linguagem utilizada

Novo paradigma:

- Paradigmas de programação anteriores: Imperativo, lógico, funcional
- POO é uma nova forma de organizar conhecimentos e resolver problemas



— / — / —  
②

S T Q Q

## Objetos como agentes:

→ Programas são compostos por agentes (objetos) que interagem entre si

→ A comunicação ocorre por mensagens, que acionam métodos específicos

## Mensagens VS. Chamadas de procedimento:

→ Em POO, há um receptor explícito da mensagem

→ A interpretação da mensagem depende do objeto que a recebe (polimorfismo)

## Classes e instâncias:

→ Todo objeto pertence a uma classe

→ Classes organizam o comportamento compartilhado dos objetos

## Herança:

→ Classes podem herdar atributos e métodos de classes superiores na hierarquia

→ Métodos podem ser sobrescritos para comportamentos específicos



③

## Resumo dos princípios da POO (Alan Kay):

- Tudo é um objeto
- Objetos comunicam-se enviando mensagens
- Cada objeto tem sua própria memória
- Todo objeto pertence a uma classe
- Classes organizam comportamento compartilhado
- Classes formam hierarquias, permitindo heranças

## Computação como simulação:

- POO modela sistemas como um conjunto de agentes autônomos interagindo
- Se aproxima da forma como resolvemos problemas no mundo real

## História da POO:

- 1960s: conceitos fundamentais (classe, herança)
- 1970s: Smalltalk avança a ideia da computação como simulação
- 1980s: C++ populariza o POO no mundo corporativo
- 1990s - 2000s: POO torna o paradigma dominante