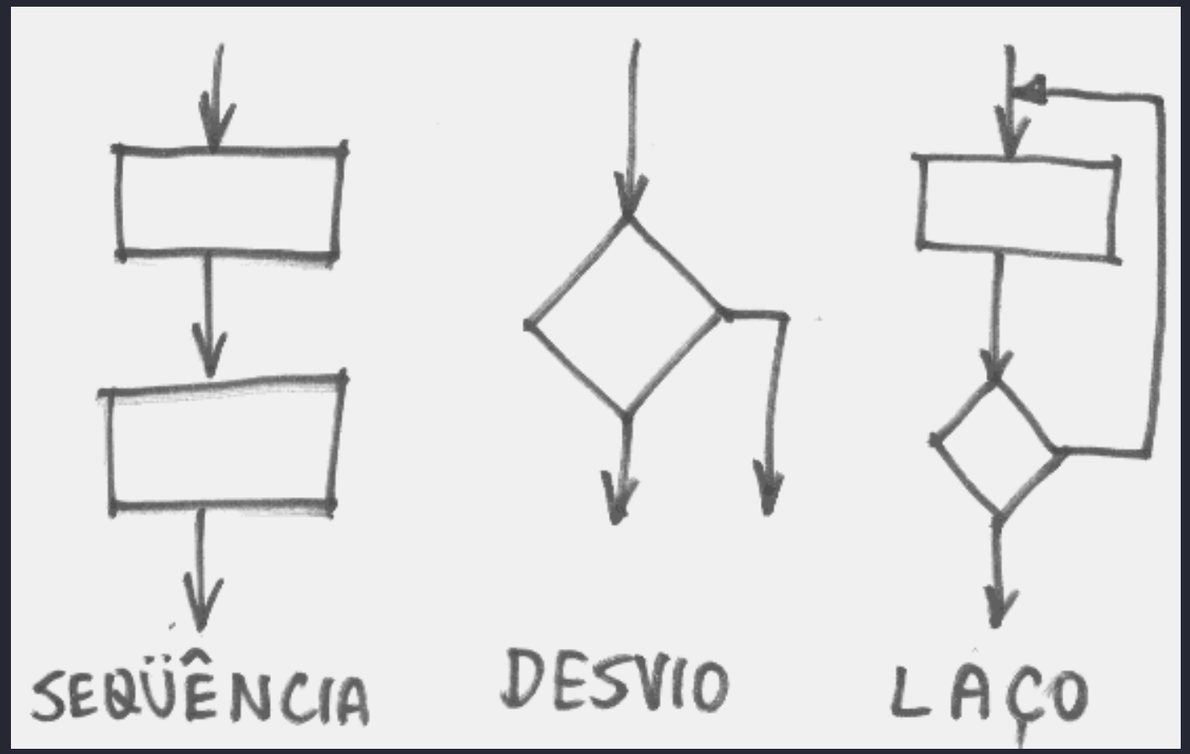
O que é POO? Quais são seus pilares?

Programação Estruturada



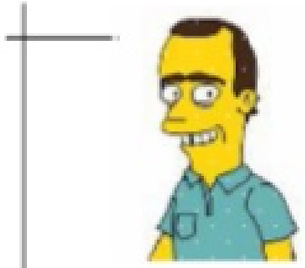
- A programação estruturada tem como principal foco as **ações**
  - Procedimentos e Funções
- Fornece maior controle sobre o fluxo de execução de um programa
  - Estruturas de sequência;
  - Estruturas de decisão;
  - Estruturas de repetição.

Programação Estruturada

- As linguagens estruturadas são de entendimento relativamente fácil
  - Por isso são utilizadas em cursos introdutórios.
- No entanto, são focadas em **como** uma tarefa deve ser feita
  - E não em **o que** deve ser feito.
- Mistura **tratamento de dados** e **comportamento** do programa.

- A programação estruturada ainda é muito influente para cada situação uma ferramenta.
- Para problemas simples e diretos, ainda é a melhor solução.

Orientação a Objetos



Professor

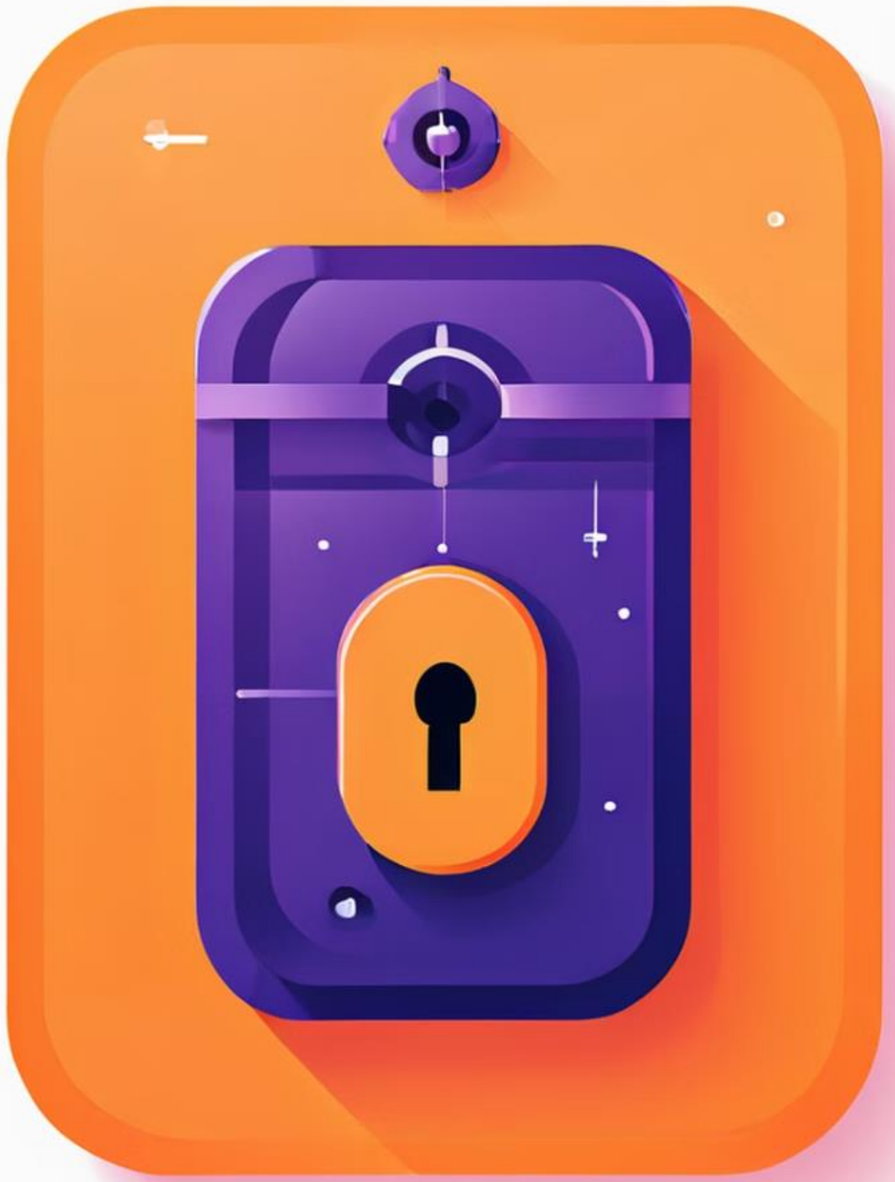
Aluno

Secretário

Professor	Aluno	Secretário
• nome	• nome	• nome
• email	• email	• email
• telefone	• telefone	• telefone
• idade	• idade	• idade
• sexo	• sexo	• sexo
• salario	• matrícula	• salario
• disciplina	• notas (2)	

Orirentação à Objetos

- Como melhor modelar o mundo real utilizando um conjunto de componentes de *software*?
- Considerando que nosso mundo é composto de **objetos**, porquê não utilizá-los?
- A ideia é modelar utilizando objetos, determinando como eles devem se comportar e como deve interagir entre si.
- Este paradigma de programação tenta ser o mais óbvio, natural e exato possível;
- São conceitos essenciais:
  - Classes e objetos;
  - Atributos, Métodos e Mensagens;
  - Herança e Associação;
  - Encapsulamento;
  - Polimorfismo.



Encapsulamento: Ocultação de Informações e Acesso Controlado

O encapsulamento protege os dados internos de um objeto, permitindo apenas acesso controlado através de métodos. Essa técnica garante a integridade e segurança dos dados, promovendo a modularidade e a organização do código.

1 **Proteção de Dados**

Protege os dados internos de um objeto de alterações não autorizadas.

2 **Controle de Acesso**

Permite o acesso aos dados apenas através de métodos definidos.

3 **Modularidade**

Promove a separação de responsabilidades e a organização do código.

4 **Reutilização**

Facilita a reutilização de código e a manutenção de programas.

Herança: Reutilização de Código e Hierarquia de Classes

A herança permite que uma classe (classe filha) herde características e comportamentos de outra classe (classe pai). Isso facilita a reutilização de código e a criação de hierarquias de classes, organizando o código de forma mais eficiente.

1

Classe Pai

Classe base que contém atributos e métodos que podem ser herdados.

2

Classe Filha

Classe que herda características da classe pai, podendo adicionar suas próprias características.

3

Reutilização de Código

Evita a duplicação de código e facilita a manutenção.

Polimorfismo: Comportamento Dinâmico e Sobrecarga de Métodos

O polimorfismo permite que objetos de diferentes classes respondam de maneiras diferentes ao mesmo método. Isso garante flexibilidade e extensibilidade ao código, permitindo que programas se adaptem a diferentes situações.

Sobrecarga

Permite a criação de métodos com o mesmo nome, mas com diferentes parâmetros.

Sobrescrita

Permite que uma classe filha redefina o comportamento de um método herdado da classe pai.

Flexibilidade

Facilita a adaptação do código a diferentes cenários e situações.

Abstração: Simplificação de Complexidade e Foco em Funcionalidades Essenciais

A abstração simplifica a complexidade do código, ocultando detalhes irrelevantes e focando nas funcionalidades essenciais. Ela facilita a compreensão e o uso de código complexo, tornando o desenvolvimento mais eficiente.

Simplificação

Oculta detalhes irrelevantes e destaca as funcionalidades essenciais.

Foco

Concentra o desenvolvimento nas funcionalidades principais, ignorando detalhes irrelevantes.

Eficiência

Facilita a compreensão e o uso de código complexo, tornando o desenvolvimento mais eficiente.

Vantagens da Programação Orientada a Objetos

Modularidade

A POO permite a criação de programas organizados em módulos independentes, facilitando a manutenção e a evolução do software.

Reutilização de Código

A POO facilita a reutilização de código e componentes, aumentando a produtividade e reduzindo o tempo de desenvolvimento.

Objetos

O que são objetos?

Objetos são entidades que possuem propriedades (atributos) e métodos (ações). As propriedades descrevem o estado do objeto, enquanto os métodos definem o que o objeto pode fazer.

1

Propriedades

Atributos que descrevem as características de um objeto. Exemplo: cor de um carro, velocidade máxima.

2

Métodos

Ações que um objeto pode realizar. Exemplo: acelerar, frear, abrir as portas.

Orirentação à Objetos

- Objetos são a chave para entender a OO;
- Se olharmos em nossa volta, encontraremos vários exemplos de objetos reais:
  - Celular;
  - Mesa;
  - Computador;
  - Janela;
  - Lâmpada…

Orirentação à Objetos

- Os objetos reais possuem duas caracterísicas
  - Estado (Atributos);
  - Comportamento.
- Por exemplo, um cachorro
  - Estado: nome, cor, raça, fome…
  - Comportamento: latindo, abanando o rabo, comendo…
- Uma bicicleta
  - Estado: marcha atual, freio, rotação…
  - Comportamento: mudando de marcha, freando…

Orirentação à Objetos

- Quais são as características de uma lâmpada?
- Quais são as características de um projetor?
  - E como tratamos a lâmpada do projetor?
- Objetos variam em complexidade
  - Porém, os detalhes relevantes dependem do contexto;
  - Esta análise de características é traduzível em orientação a objetos.

Atributos e Métodos

- Um objeto de *software* é **conceitualmente** similar aos objetos reais
- Objetos armazenam seu estado em **atributos**
  - Correspondentes as variáveis em programação estruturada.
- Objetos expõem seu comportamento através de **métodos**
  - Correspondentes às funções em programação estruturada.

Propriedades e métodos

Imagine um objeto "Carro". Ele possui propriedades como cor, modelo e número de portas, e métodos como acelerar, frear e ligar o motor.

Propriedade	Descrição
Cor	A cor do carro.
Modelo	O modelo do carro.
Número de Portas	O número de portas do carro.
Métodos	Ações que o carro pode realizar.
Acelerar	Aumenta a velocidade do carro.
Frear	Diminui a velocidade do carro.

Exemplos de classes e objetos utilizando carros

Para criar objetos, usamos o conceito de classes. Uma classe é como um molde para criar objetos do mesmo tipo. A classe "Carro" define as propriedades e métodos comuns a todos os carros.

Classe Carro

Define as propriedades e métodos para todos os objetos do tipo Carro.

- Cor
- Modelo
- Número de Portas
- Acelerar
- Frear

Objeto Carro1

Um carro específico criado a partir da classe Carro.

- Cor: Vermelho
- Modelo: Sedan
- Número de Portas: 4

Objeto Carro2

Outro carro específico criado a partir da classe Carro.

- Cor: Azul
- Modelo: Esportivo
- Número de Portas: 2

Aplicação prática do exemplo de carros em POO

Imagine um sistema de gerenciamento de veículos. Podemos usar a classe "Carro" para representar cada veículo, com suas propriedades e métodos específicos.

1

Criar um novo carro

Criar um novo objeto "Carro" com propriedades definidas pelo usuário.

2

Registrar o carro

Guardar as informações do carro no sistema.

3

Atualizar o estado do carro

Usar os métodos para atualizar o estado do carro, como acelerar ou frear.

Classes

Classes são como blueprints para criar objetos. Elas definem as características e comportamentos que os objetos daquela classe compartilham.

O computador como exemplo de classe

Imagine um computador como um objeto. Ele tem atributos, como marca, modelo, capacidade de armazenamento e processador. E também métodos, como ligar, desligar, abrir programas e navegar na internet.

Atributos

Marca, modelo, tamanho, memória, processador

Métodos

Ligar, desligar, abrir programas, navegar na internet

Atributos e métodos de uma classe

Os atributos são as características de um objeto, como a cor, o tamanho e o nome. Os métodos são as ações que um objeto pode realizar, como mover, falar ou calcular.

Atributos

Características do objeto

- Cor
- Tamanho
- Nome

Métodos

Ações que o objeto pode realizar

- Mover
- Falar
- Calcular

Instanciação de objetos a partir de uma classe

Criar um objeto a partir de uma classe é como usar um blueprint para construir uma casa. Cada objeto criado a partir da mesma classe terá as mesmas características e métodos, mas com valores diferentes.

1

Classe

Blueprint do objeto

2

Objeto 1

Instância da classe

3

Objeto 2

Instância da classe

Encapsulamento e abstração em classes

O encapsulamento protege os atributos de uma classe de serem modificados diretamente. A abstração esconde os detalhes internos da classe, revelando apenas os métodos que precisam ser acessados.

1

Encapsulamento

Protege atributos da modificação direta.

2

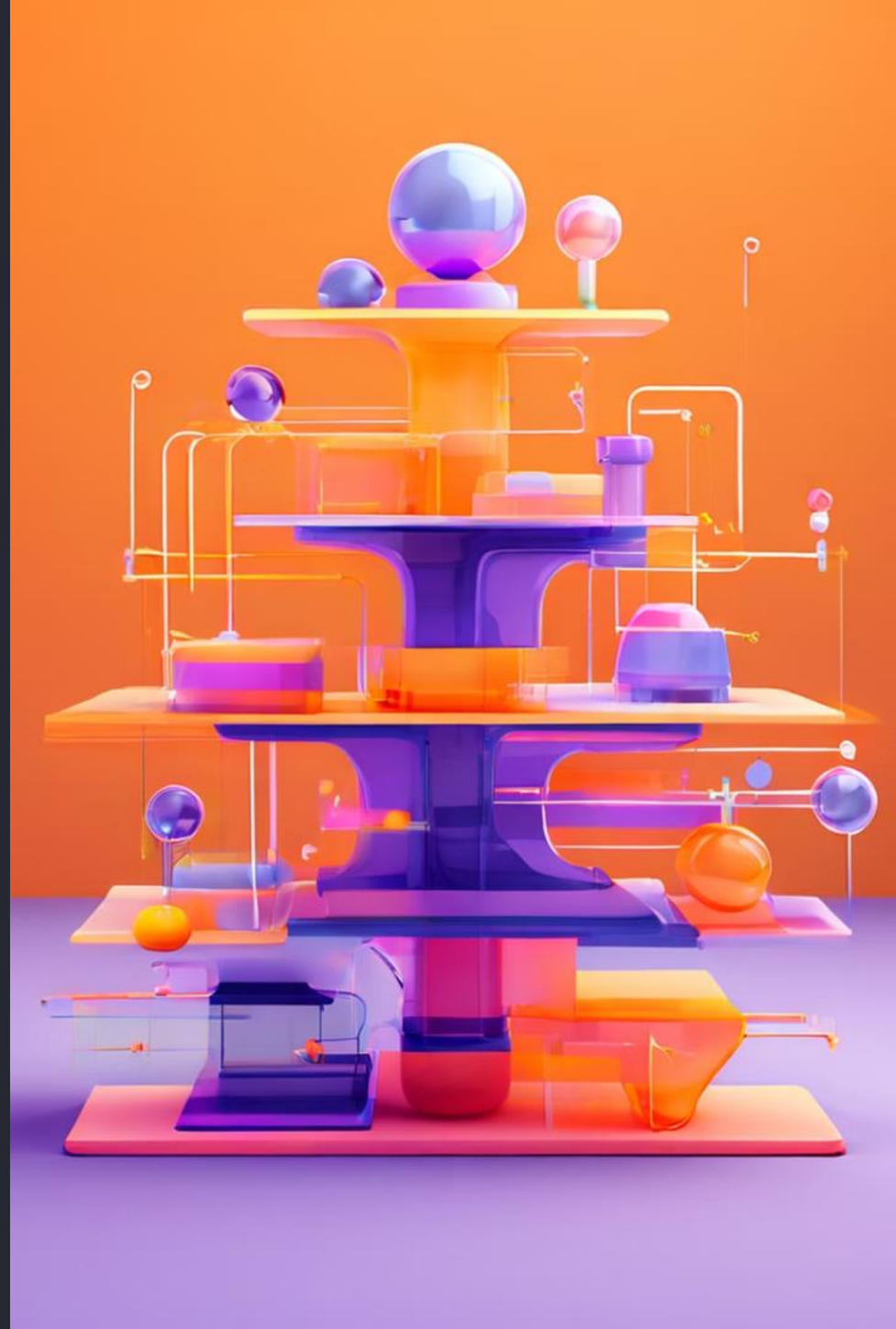
Abstração

Esconde detalhes internos da classe.

Herança e polimorfismo em classes

A herança permite que classes herdem características e métodos de classes pai. O polimorfismo permite que objetos de classes diferentes respondam de maneiras diferentes ao mesmo método.

Herança	Herda características e métodos de classes pai
Polimorfismo	Objetos de classes diferentes podem ter comportamentos diferentes



Perguntas?