

POO

O que é POO?

- Paradigma que organiza o código em classes e objetos.
- Cada objeto tem atributos e métodos.
- Utiliza conceitos como encapsulamento, herança e polimorfismo para tornar o código mais modular e reutilizável.

Porque é útil?

- Código mais modular e reutilizável.
- Facilita a manutenção e escalabilidade.
- Maior organização e estrutura no desenvolvimento.
- Encapsulamento melhora a segurança dos dados.

O que é uma classe?

- É um modelo que define como um objeto deve ser.
- Descreve os atributos (características) e métodos (ações) que os objetos dessa classe terão.
- É uma estrutura que serve para criar objetos com comportamentos e dados específicos.

O que é um objeto?

- É uma instância de uma classe.
- É uma entidade que possui atributos (dados) e métodos (ações) definidos pela classe.
- É um "exemplo concreto" de uma classe, com valores específicos para os seus atributos.

Como criar um objeto?

- Criamos um objeto usando a palavra-chave `new`, que chama o **construtor da classe**.

O que é um método?

- São **funções** dentro da classe que definem **ações** que um objeto pode realizar.

O que é uma instância de uma classe?

- É um objeto criado a partir dessa classe.
- A classe define a estrutura e o comportamento, e a instância é um objeto concreto que segue essa estrutura e pode armazenar dados específicos.
- Quando um objeto é criado estamos a instanciar a classe.
- **Exemplo:** `Department dpt1 = new Department ("DEI", "Departamento Engenharia Informática");`

O que é a palavra `this`?

A palavra-chave

`this` refere-se **ao próprio objeto que está a chamar o método**.

Usamos

`this` quando precisamos distinguir **atributos da classe** de **parâmetros do método**.

O que é a palavra `static`?

Um membro

`static` pertence **à classe, e não a um objeto específico**.

Isso significa que podemos acessar

métodos estáticos sem criar um objeto.

Qual a diferença entre `==` e `equals()`?

- `==` **compara referências na memória** (se são o mesmo objeto).
- `equals()` pode ser sobrescrito para comparar o conteúdo dos objetos.
- Sem o override, `equals()` age como `==` (comparação de referência).

- Sobrescrever `equals()` é útil para comparar objetos baseando-se nos seus atributos.

O que é o encapsulamento?

- É um dos princípios da POO.
- Consiste em esconder os detalhes internos de implementação de um objeto e expor apenas o necessário para o exterior.
- É feito através de modificadores de acesso, como `private`, `protected` e `public`, que controlam o acesso aos atributos e métodos da classe.
- O objetivo do encapsulamento é proteger os dados, garantindo que só possam ser modificados ou acessados de maneira controlada, o que aumenta a segurança e a integridade do código.

GET

Métodos

getter são usados para **obter** o valor de um atributo de uma classe. Eles são chamados de **métodos de acesso** e não alteram nada no objeto, apenas retornam o valor de um atributo privado.