**Parte Bernardo.**

**DIFERENÇAS ENTRE PROGRAMAÇÃO ESTRUTURADA E** **ORIENTADA A OBJETOS**

A programação orientada é baseada na ideia de que qualquer programa pode ser dividido em uma série de etapas lógicas sequenciais, enquanto a orientada a objetos introduz o conceito de objetos, que encapsulam dados e funcionalidades.

As principais diferenças entre esses dois paradigmas:

* **Abordagem:** A programação estruturada é centrada em funções, enquanto a orientada a objetos foca em objetos.
* **Modelagem:** A POO modela o programa como um conjunto de objetos interativos, enquanto a programação estruturada trata o programa como uma lista de instruções.
* **Complexidade:** POO é geralmente mais complexa devido aos seus conceitos de herança e polimorfismo.
* **Manutenção:** Códigos em POO tendem a ser mais fáceis de manter e atualizar devido à sua modularidade.

**VANTAGENS DA POO**

ENCAPSULAMENTO

O encapsulamento permite agrupar dados e os métodos que operam sobre esses dados em uma única entidade, conhecida como classe, trazendo vantagens como Proteção de Dados e Ocultação de Dados Internos

REUTILIZAÇÃO DE CÓDIGO

A reutilização de código possibilita a criação de classes genéricas que podem ser usadas em diferentes contextos:

* **Herança:** criar novas classes que herdam atributos e métodos de uma classe base. Isso elimina a necessidade de reescrever código semelhante, promovendo a eficiência e a consistência.
* **Polimorfismo:**  permite que objetos de diferentes classes sejam tratados de maneira uniforme, facilitando a criação de sistemas flexíveis e extensíveis.

MANUTENÇÃO FACILITADA

A POO contribui para um código mais organizado e mais fácil de manter:

* **Divisão Modular:** A separação de funcionalidades em classes modulares torna mais fácil identificar e corrigir erros específicos, sem afetar o funcionamento de outras partes do sistema.
* **Localização de Erros Simplificada:** O encapsulamento e a divisão modular permitem isolar e corrigir erros com maior precisão, minimizando os impactos colaterais.

COLABORAÇÃO EFICIENTE

A POO promove o desenvolvimento colaborativo, permitindo que equipes de programadores trabalhem em paralelo:

* **Divisão de Tarefas:** Os desenvolvedores podem se concentrar em classes específicas, aumentando a produtividade e permitindo o desenvolvimento simultâneo.
* **Integração Facilitada:** Através do uso de interfaces públicas, diferentes partes do sistema podem ser integradas de forma mais coesa.

MODELAGEM REALISTA

A POO permite modelar sistemas de forma mais próxima à realidade, usando objetos para representar entidades do mundo real:

* **Abstração:** Você pode abstrair os elementos essenciais de um sistema e criar classes que representam entidades do mundo real, tornando o código mais compreensível para desenvolvedores e stakeholders.
* **Modelagem Intuitiva:** A representação de objetos e suas interações no código se assemelha à forma como pensamos sobre o mundo, o que facilita o desenvolvimento e a manutenção.

**REFERÊNCIAS**

[**https://programae.org.br/programacao/qual-a-diferenca-entre-programacao-estruturada-e-orientada-a-objetos/**](https://programae.org.br/programacao/qual-a-diferenca-entre-programacao-estruturada-e-orientada-a-objetos/)

[**https://www.dio.me/articles/vantagens-da-programacao-orientada-a-objetos-poo**](https://www.dio.me/articles/vantagens-da-programacao-orientada-a-objetos-poo)