**Atividade Avaliativa**

1. **Por que a programação orientada a objetos é importante?**

Ela possibilita que os desenvolvedores modelem objetos e entidades do mundo real em software, facilitando a representação de sistemas complexos. Assim, a abstração permite que sejam destacadas as características e comportamentos relevantes de um objeto, simplificando o código e tornando-o mais legível.

1. **Qual a diferença entre programação orientada a objetos e programação estruturada?**

Quando programamos em POO pensamos mais em como manipular e conectar os objetos estabelecendo relações entre eles. Já a Programação Estruturada possui um código estruturado com um começo e um fim, onde os eventos vão ocorrendo em uma ordem pré-determinada, seguindo a sintaxe da linguagem escolhida.

1. **O que são classes e objetos dentro da programação orientada a objetos?**

* Os objetos possuem características próprias, denotadas por atributos.
* Os objetos podem ser categorizados, agrupados, e uma classe descreve todos os objetos de um tipo particular.

1. **Quais são os pilares da programação orientada a objetos?**

* Classes: Uma classe é uma estrutura que define as propriedades (atributos) e comportamentos (métodos) de objetos que serão criados a partir dela. É como um molde ou um plano que descreve um tipo de objeto.
* Objetos: Um objeto é uma instância de uma classe. Ele é criado a partir da definição da classe e contém os dados (atributos) e os comportamentos (métodos) definidos nela.
* Associação: A associação descreve a relação entre duas ou mais classes. A relação pode ser de diferentes tipos, como "um para um", "um para muitos", ou "muitos para muitos".
* Encapsulamento: O encapsulamento é o conceito de esconder os detalhes internos de uma classe e fornecer uma interface pública para interação com ela. Isso protege o estado interno do objeto e evita que ele seja modificado diretamente.
* Herança: A herança é um mecanismo que permite que uma classe (subclasse) herde propriedades e comportamentos de outra classe (superclasse). Isso promove o reuso de código e a especialização das classes.
* Polimorfismo: Polimorfismo permite que uma mesma operação se comporte de diferentes maneiras, dependendo do objeto que a invoca. Em POO, isso é geralmente alcançado através da sobrescrita de métodos.

1. **Quais são os benefícios de se utilizar programação orientada a objetos?**

Ela permite dividir o sistema em classes e objetos independentes, facilitando a compreensão e a manutenção do código. Além disso, outra vantagem do uso da POO é que a herança e a criação de hierarquias de classes permitem a reutilização eficiente de código.