[**Programação Orientada a Objectos**](https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264)

Denzel Rafael Ganga Francisco/ 1000019626

Faculdade de Engenharia

Universidade Católica de Angola

Denzelrafaelf@gmail.com

1. **INTRODUÇÃO**

A [Programação Orientada a Objectos](https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264) também conhecida como POO, é onde o desenvolvedor tem de começar a pensar fora da caixa, a imaginar uma forma aonde será preciso recorrer ao mundo real para o desenvolvimento das aplicações, pois hoje toda a programação em Java é orientada a objectos.

**Polimorfismo**

Polimorfismo, ponto essencial na programação orientada a objectos. Na natureza, vemos animais que são capazes de alterar sua forma conforme a necessidade, e é dessa ideia que vem o polimorfismo na orientação a objetos. Como sabemos, os objectos filhos herdam as características e acções de seus “ancestrais”. Entretanto, em alguns casos, é necessário que as acções para um mesmo método seja diferente. Em outras palavras, o polimorfismo consiste na alteração do funcionamento interno de um método herdado de um objecto pai.

**Encapsulamento**

É um conceito de programação orientada a objectos que significa separar o programa em partes, o mais isolado possível. A idéia é tornar o software mais flexível, fácil de modificar e de criar novas implementações. O encapsulamento de dados leva ao importante conceito de POO de [ocultação de dados](https://pt.wikipedia.org/w/index.php?title=Oculta%C3%A7%C3%A3o_de_informa%C3%A7%C3%A3o&action=edit&redlink=1), tornando as informações privadas para apenas a quem as possui.

**Abstração**

É utilizada para a definição de entidades do mundo real. Sendo onde são criadas as classes. Essas entidades são consideradas

tudo que é real, tendo consideração as suas características e acções.

**Herança**

Tem o mesmo significado para o mundo real. Na Orientação a Objectos é permitido que uma classe herde atributos e métodos da outra, uma das grandes vantagens de usar o recurso da herança é na reutilização do código.

Ela é usada na intenção de reaproveitar código ou comportamento generalizado ou especializar operações ou atributos.

**Interfaces**

Podemos definir como interface o contrato entre a classe e o mundo exterior. Quando uma classe implementa uma interface, se compromete a fornecer o comportamento publicado por esta interface. As classes ajudam a definir um objecto e seu comportamento e as interfaces que auxiliam na definição dessas classes.

**Referências Bibliográficas**

[1]<https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_orientada_a_objetos>

[2]<https://www.devmedia.com.br/abstracao-encapsulamento-e-heranca-pilares-da-poo-em-java/26366>