**Pilares da Programação Orientada a Objetos**

Nome: Elizandro dos Santos Alexandre / ID:1000016888

Faculdade de Engenharia

Universidade Católica de Angola

Email: Elizandro14alexandre@gmail.com

**Abstract**: Este trabalho foi feito para entendermos o funcionamento da POO (programação orientada a objecto).

**Keywords: public**, **protected, extends.**

* **INTRODUÇÃO**

Programação orientada a objetos é uma péssima ideia, que só poderia ter nascido na Califórnia." por Edsger Dijkstra. Orientação a objetos é uma maneira de programar que ajuda na organização e resolve muitos problemas enfrentados pela programação procedural. forma de procedimentos, também conhecidos como métodos.

* **CONCEITOS**

**Polimorfismo**

Polimorfismo é a capacidade de um objeto poder ser referenciado de várias formas.

**Encapsulamento**

Encapsular é a técnica usada para esconder todos os membros de uma classe ou seja, não expor detalhes internos para o usuário, tornando partes do sistema mais independentes possível, além de esconder como funcionam as rotinas do nosso sistema.

**Herança**

É uma forma de reutilização de software na qual uma nova classe é criada, absorvendo membros de uma classe existente e aprimoradas com capacidades novas ou modificadas. Na Orientação a Objetos é permitido que uma classe herde atributos e métodos da

outra, tendo apenas uma restrição para a herança. Os métodos e atributos só podem estar com visibilidade **public** e **protected** para que sejam herdados. Vantagem: reutilização de códigos.

**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 como consideração as suas características e ações.

**Interface**

É um recurso muito utilizado em Java, bem como na maioria das linguagens orientadas a objeto.Interface é a maneira através da qual conversamos com um objeto “Autenticavel”. Lemos a interface da seguinte maneira: "quem desejar ser autenticável, precisa saber autenticar dado um **inteiro** e retornando um **booleano**". Uma interface pode definir uma série de métodos, mas nunca conter implementação deles. Ela só expõe o que o objeto deve fazer, e não como ele faz, nem o que ele tem. Como ele faz vai ser definido em uma implementação dessa interface.

**ReferênciasBibliográficas**

http://www.caelum.com.br/apostilas <https://www.devmedia.com.br/os-pilares-da-programacao-orientada-a-objetos/9264>