Pilares da Programação Orientada a Objetos

Nome:Yetsin Ferreira / Id:1000019388

Faculdade de Engenharia

Universidade Católica de Angola

Email:yetsinferreira@gmail.com

**Abstract**: Este pequeno trabalho teórico aborda sobre os pilares da POO(programação orientada a objecto)

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

1. **INTRODUÇÃO**

Programação orientada a objetos é um paradigma de programação baseado no conceito de "objetos", que podem conter dados na forma de campos, também conhecidos como atributos, e códigos, na forma de procedimentos, também conhecidos como métodos.

1. **CONCEITOS**

**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.

**Encapsulamento**

É a técnica utilizada para esconder uma ideia, ou seja, não expôr detalhes internos para o usuário, tornando partes do sistema mais independentes possível.

**Herança**

O significado de herança tem o mesmo significado para o mundo real. Assim como um filho pode herdar alguma característica do pai, 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.

**Polimorfismo**

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse.

**Interface**

É um recurso muito utilizado em Java, bem como na maioria das linguagens orientadas a objeto, para “obrigar” a um determinado grupo de classes a ter métodos ou propriedades em comum para existir em um determinado contexto, contudo os métodos podem ser implementados em cada classe de uma maneira diferente. Pode-se afirmar que uma interface é um contrato que quando assumido por uma classe deve ser implementado.

**ReferênciasBibliográficas** <https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_orientada_a_objetos> <https://www.devmedia.com.br/os-pilares-da-programacao-orientada-a-objetos/9264>