Mecanismos OOP Em Java

Nome: Henriques Carvalho / Id:1000018895

Faculdade de Engenharia

Universidade Católica de Angola

Email: henriquescarvalho4@gmail.com

**Abstract**: Neste pequeno trabalho teórico iremos abordar sobre Mecanismos OOP em Java

**Keywords : Abstração, Polimorfismo, Herança, Interfaces e Encapsulamento**

1. **INTRODUÇÃO**

Programação orientada a objetos é um paradigma de programação que usa abstração para criar modelos baseados no mundo real. OOP permite melhorar a reusabilidade e extensibilidade dos softwares.

**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.

**Polimorfismo**

O Polimorfismo é a capacidade de uma variável se referir em tempo de execução a objetos de diversas classes. Ou pode ser o nome dado a capacidade que objetos diferentes tem de responder a uma mesma mensagem

O Polimorfismo ocorre quando um método que já foi definido no ancestral é redefinido no descendente com um comportamento diferente

**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.

**Encapsulamento**

É o princípio de projeto pelo qual cada componente de um programa deve agregar toda informação relevante para sua manipulação como uma unidade. O encapsulamento disponibiliza o objeto com toda a sua funcionalidade sem que você precise saber como ele funciona internamente, nem como armazena internamente os dados que você recupera.

**Herança**

Significa que todos os atributos e métodos programados no ancestral já estarão automaticamente presentes em seus descendentes sem necessidade de reescreve-los. 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 e há reutilização de códigos.

**Referências Bibliográficas**

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

Programção de computadores Vol.4 Centro Paula Souza

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