

ATIVIDADE PESQUISA – MATHEUS HENRIQUE BUTKOSKI SILVA

TADS – UNIVEL

1)

O propósito da POO (Programação Orientada a Objetos) é aproximar o mundo lógico da programação ao mundo real. É baseada no conceito de objetos, logo as características desses objetos serão seus atributos, enquanto suas funcionalidades serão seus métodos.

A POO, possui quatro pilares básicos: **Encapsulamento, Herança, Polimorfismo e Abstração**. Estes são essenciais para que uma linguagem possa ser considerada orientada a objetos.

Existem também aquilo que podemos chamar de classes, que é definido como a forma que um objeto apresenta. Exemplo: Carro, Avião, Televisão, etc...

2)

A POO, aponta para uma forma mais apropriada de dividir o programa em conceitos (classes), o que permite que cada classe tenha em si os dados e funcionalidades necessárias que pertencem somente a ela. Enquanto na Programação Estruturada o conceito de classes não opera, o que leva o programa, por muitas vezes, possuir grandes blocos de código fonte sem uma organização de dados.

As linguagens orientadas a objeto são sucessoras das estruturais, visto que aprender a lógica de programação se torna mais fácil na programação estruturada, pois os conceitos necessários são menores, mas para se ter um código com mais funcionalidades e organização se torna necessária a POO.

3)

Os ganhos ao se utilizar a POO são basicamente a **melhora na legibilidade** do código, pois é mais dividido e normalmente não possui excesso de código, **facilidade de manutenção e expansão** e **construção de abstrações** são mais difíceis de representar com programação estruturada.

O que pode afastar alguns e ser considerada como desvantagem é a **complexidade de aprendizado**, pois possui mais conceitos que os outros tipos de programação, **maior uso de memória** e **criação de programas extremamente simples**, pois por muitas vezes um programa que não exige tantas funcionalidades pode ser mais fácil e rápido de ser programado em programação estrutural.

4)

Os objetos são aquilo que possuem características que são chamadas de **atributos**, utilizando o exemplo de um carro: Placa, Marca, Cor, Ano, etc... Após definido, o objeto deve pertencer a uma classe que neste exemplo seria a classe Carros.

Em outras palavras, classe é a base que será utilizada para criar um objeto. Pode-se pensar também que classe é um molde que reúne as características de um grupo de objetos.

5)

Encapsulamento é um conceito que permite que certas propriedades sejam protegidas de acessos diretos. A propriedade contundia funcional, porém sem mostrar como. Isso garante uma camada extra de proteção, pois os detalhes de implementação não são revelados.

Pode ser garantida através da declaração de se o método é privado ou público.

Abstração é o conceito de criação de classes e objetos. A ideia é representar um objeto de forma abstrata, que será herdado por outras classes. Dessa forma, é possível abstrair objetos do mundo real e listar seus atributos e métodos.

6)

Modificadores de acesso são palavras-chave que definem níveis de acesso aos atributos, métodos e classes.

Private - É o mais restritivo e só são acessíveis pela classe que os declara.

Protected - Menos restritivo e permite acessos pela classe que os declara, subclasses em outros pacotes e outras classes dentro do mesmo pacote.

Public – O mais permissivo. Atributos, métodos e classes serão totalmente acessíveis e além disso os atributos e métodos serão herdados pelas subclasses.

Default – Modificador de acesso padrão, normalmente usado quando nenhum outro modificador for definido. Assim os atributos, métodos e classes estarão visíveis em todas as classes do pacote.

Referências:

<https://www.devmedia.com.br/programacao-orientada-a-objetos-x-programacao-estruturada/31852>

<https://blog.betrybe.com/tecnologia/poo-programacao-orientada-a-objetos/>

<https://medium.com/trainingcenter/modificadores-de-acesso-3f87133611c8>