**Fatec Zona Sul – Dom Paulo Evaristo ARNS**

Silvana Sales de Menezes

**MEU ENTENDIMENTO SOBRE POO - PROGRAMAÇÃO ORIENTADA A OBJETO**

Trabalho apresentado à disciplina de Técnica de Programação I, do curso de Desenvolvimento de Software Multiplataforma, turma do 2.º semestre, período da tarde da Fatec Zona Sul.

Solicitante: Profº. Lelis Tetsuo Murakami

São Paulo

2023

**2º Semestre de DSM – Técnica de Programação I - Profº LELIS TETSUO MURAKAMI**

**Aluna: Silvana Sales de Menezes**

***Meu entendimento sobre Programação Orientada a Objeto.***

A programação orientada a objeto é uma metodologia de programação que permite construir códigos que posteriormente poderão se reutilizado em outros programas.

Um objeto é a definição de algo que recebe seus atributos instânciados de uma classe, ou seja, a classe do objeto Aluno, recebe como atributo suas caracteríscas cadastrais, tais como, nome, cpf, email.

A sintaxe dessa etapa deve ser a seguinte:

*pacote* ***estudar****;*

*public class Estudar() {*

*private String disciplina; // Atributos*

*private int tempo;*

*private boolean estudando;*

*public void fazerExercicios(){ // Métodos Públicos*

*teste1 = disciplina + tempo*

*}*

*public Estudar (String disciplina, int tempo, boolean estudando) { // Métodos Especiais*

*this.disciplina = disc;*

*this.tempo = temp;*

*this.estudando = estd;  
 }*

*}*

Os ***atributos*** acima foram criados com visibilidade ***privada***onde somente essa classe terá acesso, essa é uma forma de proteger essas informações do acesso de outras classes, presente no método de ***encapsulamento****,* que é um dos pilares da POO.

Existem também a visibilidade ***pública*** onde a classe atual e todas as outras tem acesso e a visibilidade ***protegida***, onde a classe atual e todas as suas subclasses tem acesso.

Os ***métodos públicos***, são comportamentos pertinentes ao objeto que será instânciado pela classe.  
Os ***métodos especiais***, são os construtores do objeto, podemos definir o ***construtor*** com todos ou parte dos atributos que construirá o objeto, assim como os métodos de ***getters***e ***setters***, onde podemos pegar e modificar um atributo ou comportamento do objeto.

Assim que esses atributos e comportamentos são preenchidos, o objeto se torna real.

Outro pilar da POO é a ***Herança***, com ela podemos fazer com que a nova classe herde os atributos e comportamentos necessários da superclasse e acrescenta-se um atributo ou comportamento específico a ela, quando necessário.

Usamos a expressão ***extends*** e o nome da superclasse na subclasse para que os métodos e atributos sejam herdados.

A ***interface*** não tem atributos, recebe apenas os métodos abstratos definidos com visibilidade pública, que serão os métodos que existirão no objeto. (o código de cada método é feito no construtor na classe do objeto).

A classe que terá uma interface, recebe uma implementação, para isso utilizamos a expressão ***implements*** e seus atributos devem ser protegidos ou privados, nunca públicos.

Para criar um programa orientado a objeto, precisamos de um pacote onde conste a maquina virtual que ira ler e interpretar nosso código, afim de copilar e posteriormente executá-lo.