SAMUEL COELHO CAMILO – 539682

RESUMO ESTENDIDO DE CONCEITOS SOBRE PROGRAMAÇÃO ORIENTADA A OBJETOS

- **CLASSE:** Um modelo para criação de objetos onde consta seus métodos, atributos. Usando esse modelo, podem se criar vários objetos dessa classe. Uma classe também pode ser usada para criar classes filhas
- **OBJETO:** Uma instância de uma classe, ou seja, um objeto criado a partir de uma classe, este objeto tem métodos e atributos conforme é descrito em sua classe. Além de ter métodos e atributos das classes pai da sua classe. É a base da programação orientada a objetos.
- **MÉTODO E ATRIBUTO:** métodos são “funções”, que um objeto pode ter, métodos podem receber parâmetros, retornar objetos ou outros dados simples, e em geral podem fazer algo, atributos são as “variáveis” do objeto, armazenando dados simples ou até outros objetos. Os métodos de um objeto/classe demonstram como aquele objeto se comporta, enquanto os atributos de um objeto/classe dizem oque esse objeto tem e quais caracteristicas definem esse objeto.
- **ENCAPSULAMENTO:** É um conceito sobre limitar o acesso de atributos e métodos de objetos por partes externas do código. Isso é útil para não só organizar o código mas também para proteger certos métodos e atributos de serem acessados por terceiros.
- **VISIBILIDADE DE ATRIBUTOS E MÉTODOS (PUBLIC, PRIVATE, PROTECTED):** Atributos e métodos que são marcados como *PUBLIC* podem ser acessados de qualquer lugar, quando são marcados como *PRIVATE*, significa que só podem ser acessados dentro da classe, caso sejam marcados como *PROTECTED*, poderão apenas ser acessados dentro do mesmo pacote.
- **HERANÇA:** É um conceito sobre como algumas classes podem herdar atributos e métodos de outras classes. Além de evitar repetição de código que já havia sido escrito para outra classe, heranças também ajudam a organizar o código e deixá-lo mais fácil de entender. Classes que Herdam de outras classes também permitem polimorfismo de seus objetos. Um objeto de uma classe terá todos os atributos e métodos da classe pai além dos próprios métodos e métodos.
- **POLIMORFISMO:** Refere-se a como objetos podem mudar para a forma de uma classe pai para se adequar a um método que recebe este como parâmetro. Isso evita criar vários métodos diferentes para cada um receber uma classe diferente com atributos e/ou métodos similares. Criando se um método que recebe uma classe pai, todos os objetos de classes filhas podem ser aceitos por aquele método.
- **CLASSES ABSTRATAS:** São classes que não podem ser usadas para criar objetos, a função dessas classes é terem classes filhas e essas classes poderam ser usadas para criar objetos. É um conceito útil quando se quer criar polimorfismo para métodos ou evitar repetir código mas mesmo assim não querer ter objetos dessa classe pai. É usada para conceitos abstratos como o conceito de mamiferos, vários animais são mamiferos, mas nenhum animal é apenas mamifero, então a classe dedicada para mamiferos deveria ser abstrata e classes filhas como lobo ou elefante seriam “concretas”, ou seja, classes que poderiam criar objetos.
- **CONSTRUTORES:** São métodos especiais que são “chamados” quando um novo objeto é criado, geralmente esses métodos buscam preencher atributos

necessários que este objeto possa ter. Esses métodos não tem retorno e tem como nome o mesmo nome da classe.

- **GET E SET:** Refere-se a métodos especiais que podem retornar valores protegidos (*GET*) e até alterar o valor de atributos protegidos (*SET*). Esses métodos são úteis para controlar quem pode ver tais métodos e quem pode mudar tais métodos, além de também serem usados para realizar uma função para toda vez que um atributo for acessado.
- **SOBRECARGA DE MÉTODOS:** É oque acontece quando existem dentro de um objeto mais de um método com o mesmo nome, só que com parâmetros diferentes, então se pode escolher qual método usar ao dar preencher os parâmetros. Isso pode também acontecer quando se cria um método para uma classe filha com o mesmo nome de um método de uma classe pai, mas com parâmetros diferentes. Isso é muito útil para dar a impressão que um único método pode receber diferentes tipos de parâmetros.
- **SOBRESCRITA DE MÉTODOS:** Quando uma classe filha tem um método com o mesmo nome e parâmetros que sua classe pai, o método em questão é sobrescrito, ou seja, apesar de ter o mesmo nome de um método que a classe pai tem, quando chamado, fará algo diferente que o método que a classe pai faria. Pode ser muito útil alterar o comportamento de uma classe pai na classe filha para melhor adequar com oque a classe filha deveria ser e fazer.

1. **PALAVRAS RESERVADAS (SUPER, THIS E FINAL):** *SUPER* é referente a classe pai, é usada de métodos para chamar os métodos sobrescritos da classe pai. Isso é útil para evitar copiar código. *THIS* é referente a própria classe e é usada para acessar os próprios métodos e atributos sem confundir com variáveis que possam ter o mesmo nome. *FINAL* é uma palavra que é usada para fazer com que uma classe não possa ter filhos, ou seja, nenhuma classe pode herdar de uma classe final. Isso é útil para proteger uma classe de ter seus métodos e atributos alterados por terceiros.

- **RELAÇÃO DE OBJETOS: “TER”, “USAR”, “SER”:** Objetos “TEM” atributos, isso significa que o atributo é parte desse objeto. Objetos “USAM” outros objetos, em outras palavras, objetos podem retornar objetos ou receber esses objetos como parâmetros de seus métodos ou talvez instanciar um objeto dentro de um método para usar de algum jeito. Objetos “SÃO” de uma classe, não só dá classe que foi usada para criar esse objeto mas também de suas classes pais.