

UFC – Campus Sobral
Técnicas de Programação
Professor Fischer Jonas Ferreira
Lista 4 – Exercício 1

| Aluno: Ryan Gomes Magalhães Lima | Matrícula: 538554 | Semestre: 2023.1 |

Escreva um pequeno resumo estendido que descreva e comente sobre cada um dos itens abaixo:

- **CLASSE:** Classe é um modelo de objeto formado a partir de atributos/variáveis (Ex: `public int numero`) e métodos/funções (Ex: `public void soma(int a, int b){}`). Permitindo, assim, com esse molde criar vários objetos que possuem os mesmos atributos e métodos, mas com valores diferentes.
- **OBJETO:** Objeto é uma instância de uma classe já criada, herdando todas as propriedades da classe.
- **MÉTODO E ATRIBUTO:** Método/Função é um bloco de comandos que realizar uma determinada tarefa, recebendo ou não parâmetros para poder trabalhar em cima e retornando (opcional) o que a tarefa deseja realizar, fazendo, assim, que o código fique mais organizado e sucinto.
- **ENCAPSULAMENTO:** Consiste em proteger os dados (atributos e métodos) de uma classe, fazendo com que só possam ser acessados pelo encapsulamento que o programador escolheu, sendo eles – `public`, `private` e `protected`.
- **VISIBILIDADE (PUBLIC, PRIVATE E PROTECTED):** No encapsulamento `Public` as informações da classe fica visível para todos, ou seja, qualquer projeto java pode acessar aquela classe. No `Private` os dados só podem ser acessados dentro da própria classe. E no `Protected` as informações podem ser acessadas tanto pela própria classe quanto para as suas subclasses.
- **HERANÇA:** A Herança permite que uma classe (subclasse ou classe filha) herde os atributos e métodos de outra classe (superclasse ou classe pai), podendo usar os atributos e métodos já criados na superclasse, adicionar novos ou modificar os existentes.
- **POLIMORFISMO:** É quando uma subclasse utiliza o mesmo método da subclasse, mas alterando o que o método faz. Ex: a superclasse `forma` tem o método `calcularArea()` e as subclasses `quadrado` e `triângulo` também possuem esse método, mas alterando como faz os cálculos e o que retorna.
- **CLASSES ABSTRATAS:** É uma classe que não pode se instância diretamente, ou seja, não pode criar um objeto de uma classe abstrata. Sendo usada, normalmente, para criar métodos abstratos que serão implementados nas subclasses.
- **CONSTRUTORES:** É um método que serve para inicializar a classe, possuindo o mesmo nome da classe, e parâmetros para inicializar os atributos para, assim, criar objetos válidos e que possam está prontos para execução.

- **GET E SET:** São métodos utilizados para acessar ou modificar os atributos de uma classe. O GET ("Pegar" em português) é usado para retornar o valor daquele determinado atributo, sendo útil para acessar atributos que tem encapsulamento private. O SET ("Definir" em português) é usado para alterar o valor de um atributo, recendo como parâmetro um valor do mesmo tipo daquele atributo que deseja alterar.
- **SOBRECARGA DE MÉTODOS:** Ela permite que uma classe possua 2 ou mais métodos com o mesmo nome, mas recebendo parâmetros diferentes para poder diferenciar uma da outra. Geralmente sendo usada nos construtores da classe, para criar objetos diferentes de uma mesma classe.
- **SOBRESCRITA DE MÉTODOS:** Ela permite que uma subclasse substitua um método da superclasse por sua própria implementação. Dessa forma, o método da classe filha deve retornar e receber os mesmos parâmetros da classe pai, mas alterando dentro do bloco de código, assim o método da classe filha será executada ao invés do método da classe pai.
- **PALAVRAS RESERVADAS (super, this e final):** São palavras-chaves que possuem um significado especial, não podendo ser usadas como identificadores (nomes de variáveis, classes, métodos, etc.) em programas. O "super" é uma palavra-chave que serve para referenciar uma classe pai a partir de uma classe filha, ex: no construtor da classe filha haverá o super() que vai receber parâmetros que servirão para inicializar o construtor da classe pai. O "this" é uma referência para o objeto atual da classe que está sendo usado, isso é necessário quando se vai trabalhar com variáveis que possuam o mesmo nome e que o programador queira diferenciar uma da outra. O "final" é usada para declarar uma variável, método ou classe que não pode ser alterado ou estendido de forma alguma; quando utilizada numa variável, a variável não pode ser alterada depois que se inicializa; quando utilizada no método, ele não pode ser sobrescrito na subclasse; já na classe, ela não pode ser estendida por outras classes.
- **RELAÇÃO DE OBJETOS ("TER", "USAR" E "SER"):** O "Ter" refere-se à posse ou propriedade de um objeto, ou seja, relacionado ao método set. O "Usar" refere-se à ação de utilizar ou acessar um objeto para um determinado fim, ou seja, relacionado ao método get. E "Ser" se refere à identidade ou natureza do próprio objeto sendo usado, ou seja, relacionado a criar uma nova instância de uma classe.