

# Oficina de Java - fundamentos da orientação a objetos.

1. Abstração
2. Encapsulamento
3. Herança
4. Polimorfismo



# 1 Abstração

- Este é o segundo pilar da POO
- Ele se refere à capacidade de extrair características e comportamentos relevantes de um determinado objeto do mundo real. (SOUZA, O.)



## Como fazer uma boa abstração

- Para podermos criar um bom molde/modelo (classe), de um determinado objeto do mundo real, devemos extrair suas características, comportamentos e estado, de forma que possam replicá-lo e gerar outros objetos diferentes utilizando as mesmas características extraídas anteriormente.

## Exemplo

- Supondo que queremos criar um molde de um copo de vidro. Devemos primeiro identificar quais são suas características.
- O molde deve possuir as características, comportamento e estado que possibilite recriar fielmente seus pontos principais.





## Características do objeto copo de vidro

- Tipo de material
- diâmetro
- tamanho
- Cor

## Comportamentos de um copo de vidro

- incrementar líquido
- Derramar líquido
- Limpar
- Durabilidade





## Possíveis estados de um copo de vidro

- Ao observar a condição do objeto em um exato momento, pode-se identificar seu estado.
- Estado atual: copo de vidro está quebrado.

# Encapsulamento

- Se refere à visibilidade de uma determinada implementação.
- Quando criamos uma classe e queremos ocultar os detalhes internos de sua implementação, podemos utilizar deste princípio para fornecer métodos mais seguros, através dos **modificadores de acesso**. (SOUZA, O.)
- Java nos permite usar três modificadores de acesso: *public*, *protected* e *private*.

## **Public**

- Este modificador de acesso, permite que um método, atributo ou qualquer outro componente, seja visto em qualquer área do código.



## Exemplo associativo

- Imagine que ao *public* pode ser visto e utilizado em qualquer parte do código, sem restrição.
- Pode se associar a um orelhão, que pode ser utilizado por qualquer pessoa, sem restrição.

## Protected

- Esse modificador lhe permite mostrar detalhes de implementação das classes, apenas no seu determinado pacote.
- Imagine um pacote em java, como sendo uma pasta, esta pasta pode conter vários arquivos (classes). Literalmente o diretório atual de um determinado arquivo.

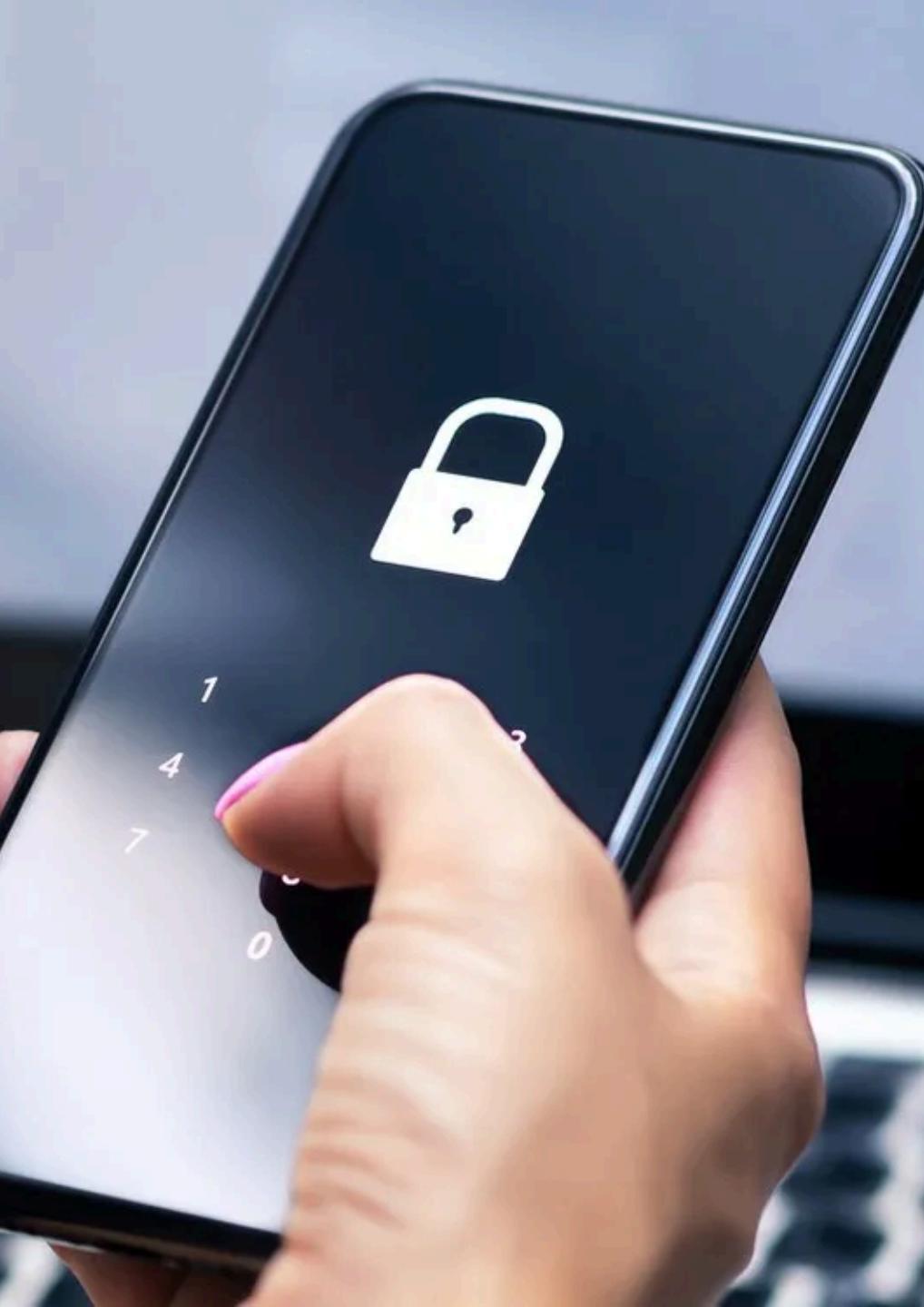
## Exemplo associativo

- Imagine um método com o modificador `protected`, como sendo um telefone domiciliar. Onde quem tem acesso a ele, é apenas as pessoas que moram na mesma casa que você.



## Private

- O modificador *private*, permite ocultação completa sobre a implementação de uma determinada coisa na classe, podendo ser um atributo, método, etc.



## Exemplo associativo

- Imagine algo com o modificador *private*, como sendo um seu smartphone pessoal, onde quem tem acesso é só você. Quem mora com você não tem acesso (*protected*), muito menos pessoas desconhecidas (*public*).

# Acessadores e modificadores

- Quando criamos um atributo em java, precisamos criar dois métodos para o mesmo, os famosos **gets and sets**.

## Acessadores (GETs)

- Os métodos gets servem para retornar o valor atual que está contido em uma determinada variável(atributo).

```
public class Pessoa {  
    private String nome; // Variável de instância  
  
    // Método acessador (get)  
    public String getNome() {  
        return nome;  
    }  
}
```

# Modificadores (SETs)

- Estes métodos são utilizados para modificar o valor de um determinado atributo.

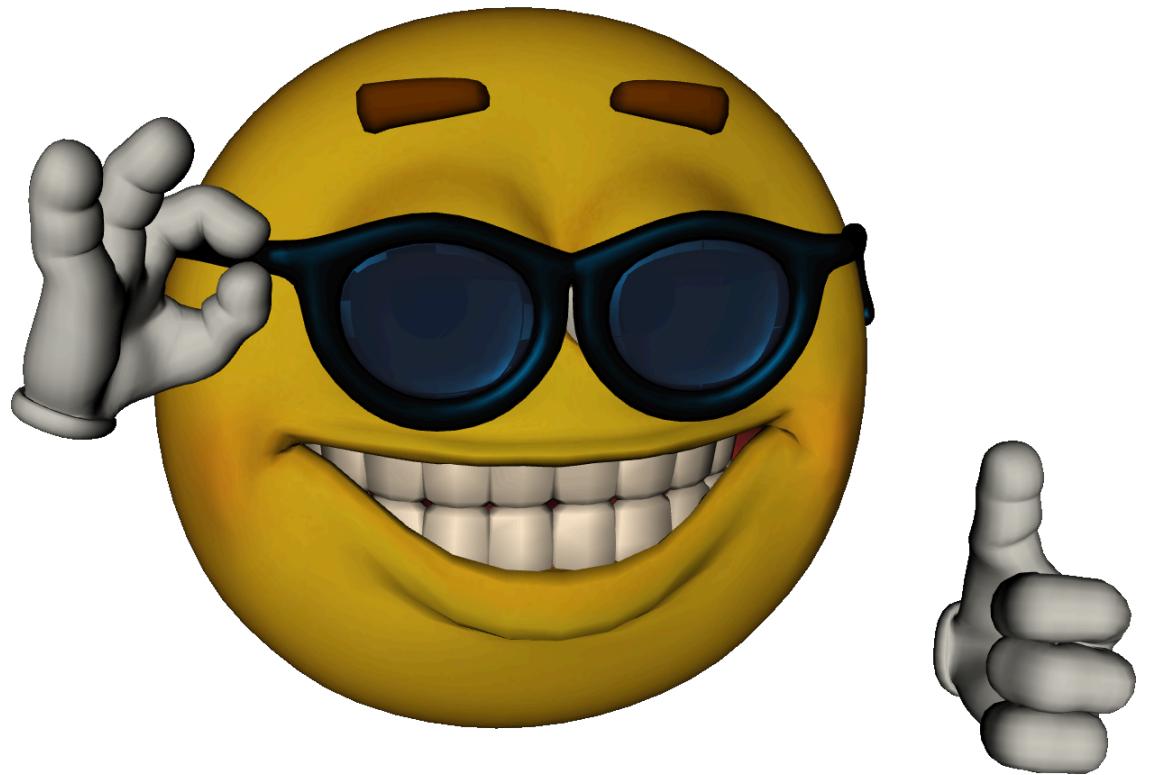
```
public class Pessoa {  
    private String nome; // Variável de instância  
  
    // Método modificador (set)  
    public void setNome(String novoNome) {  
        this.nome = novoNome;  
    }  
}
```

# Concluindo

- Um atributo privado, deve (normalmente), contém um método getter e setter, possibilitando que o mesmo seja acessado e modificado.

```
public class Pessoa {  
    private String nome; // Variável de instância  
  
    // Método acessador (get)  
    public String getNome() {  
        return nome;  
    }  
  
    // Método modificador (set)  
    public void setNome(String novoNome) {  
        this.nome = novoNome;  
    }  
}
```

Atividade :D



# 01 Abstração

- Com base no que foi estudado, crie um molde/classe em java para as imagens a seguir:



**GARRAFA TÉRMICA INOX SLIM  
1,8 LITROS INVICTA**



i-SOW

# HORÁRIO DAS AULAS

NOME ..... CLASSE .....

SEGUNDA

TERÇA

QUARTA

QUINTA

SEXTA




## 02 Encapsulamento

- Encapsule as classes criadas anteriormente, onde os atributos devem ser privados(podendo ser "acessados", somente por funções GETs e SETs) e os métodos devem ser públicos ou protegidos.

## Referências:

SOUZA, O. Os quatro pilares da POO em Java. Disponível em:

<https://www.dio.me/articles/os-quatro-pilares-da-poo-em-java>.