PROGRAMAÇÃO ORIENTADA A OBJETOS
ENCAPSULAMENTO

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

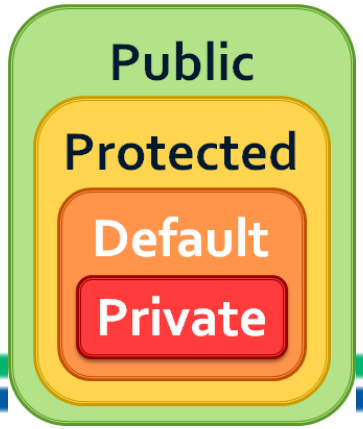
HABNER FABRÍCIO BOESING
habner.boesing@unoesc.edu.br

OBJETIVOS

- Pilares da Orientação a Objetos:
  - Encapsulamento

ENCAPSULAMENTO – PROTEÇÃO DE DADOS

- O *encapsulamento* utiliza-se dos níveis de acesso à atributos e métodos para definir camadas adicionais de proteção aos dados, evitando assim que uma classe modifique o valor de um atributo que possa estar sendo utilizado por outra classe.
- Para isso é necessário entender que existem três tipos diferentes de níveis de acesso em Java:
  - **public** - Público: permite que os atributos ou métodos da classe possam ser utilizadas por todas as outras classes do projeto. Representado pelo símbolo +
  - **protected** - Protegido: permite que os atributos ou métodos da classe possam ser utilizados pelas outras classes no mesmo pacote ou por qualquer classe "filha" desta classe. Ver conceito de Herança. Representado pelo símbolo #
  - **default** – Padrão: permite o acesso somente à classes que estejam no mesmo pacote.
  - **private** - Privado: limita o uso de atributos ou métodos somente à classe atual, impedindo que demais classe tenham acesso. Representado pelo símbolo -



Visibilidade	public	protected	default	private
A partir da mesma classe	✓	✓	✓	✓
Qualquer classe no mesmo pacote	✓	✓	✓	⊘
Qualquer classe filha no mesmo pacote	✓	✓	✓	⊘
Qualquer classe filha em pacote diferente	✓	✓	⊘	⊘
Qualquer classe em pacote diferente	✓	⊘	⊘	⊘

UNOESC

ENCAPSULAMENTO

- Implementam o encapsulamento baseado em propriedades privadas, ligadas a métodos especiais chamados *getters* e *setters*.
- Evita o acesso direto a propriedade do objeto.
- Desta forma para enviar um valor para o atributo utiliza-se um *set* e para pegar o valor utiliza-se um *get*.

```java
public class Joystick {
    private String modelo;
    private String cor;
    private int porcentagemBateria;

    public String getModelo() {
        return modelo;
    }
    public void setModelo(String modelo) {
        this.modelo = modelo;
    }
    public String getCor() {
        return cor;
    }
    public void setCor(String cor) {
        this.cor = cor;
    }
    public int getPorcentagemBateria() {
        return porcentagemBateria;
    }
    public void setPorcentagemBateria(int porcentagemBateria) {
        this.porcentagemBateria = porcentagemBateria;
    }
```

UNOESC

ENCAPSULAMENTO

- Na classe Principal muda-se a forma de indicar os atributos do objetos.
- Se antes era possível acessar livremente apenas digitando o nome do objeto e o atributo. Exemplo: **js1.modelo = "PS4"**
- Agora evita-se o acesso direto ao atributo, fazendo com que seja utilizado um método público chamado setModelo() que de forma interna na classe Joystick acesse o atributo modelo e altere sua propriedade.
  Exemplo: **js1.setModelo("PS4")**

Sem Encapsulamento

```
public class Principal {
    public static void main(String[] args) {

        Joystick js1 = new Joystick();
        js1.modelo = "PS4";
        js1.cor = "Preto";
        js1.porcentagemBateria = 100;
        System.out.println("Modelo: " + js1.modelo);
```

Com Encapsulamento

```
public class Principal {
    public static void main(String[] args) {

        Joystick js1 = new Joystick();
        js1.setModelo("PS4");
        js1.setCor("Preto");
        js1.setPorcentagemBateria(100);
        System.out.println("Modelo: " + js1.getModelo());
```

ATIVIDADE PRÁTICA

- 1) Crie uma classe para representar uma **pessoa**, com os atributos privados **nome**, **anoNasc** e **altura**. Crie os métodos públicos necessários para **sets** e **gets** e também um método para imprimir todos dados de uma pessoa. Crie um método para calcular a idade da pessoa.
- 2) Crie uma classe **Televisao** e uma classe **ControleRemoto** que pode controlar o volume e trocar os canais da televisão. O controle de volume permite:
  - aumentar ou diminuir a potência do volume de som em uma unidade de cada vez;
  - aumentar e diminuir o número do canal em uma unidade
  - trocar para um canal indicado;
  - consultar o valor do volume de som e o canal selecionado.