

Encapsulamento

Programação Orientada a Objetos



O que é encapsulamento?

01.

O **encapsulamento** é um princípio fundamental da programação orientada a objetos em Java.

02.

Refere à ideia de que os dados de uma classe devem ser **protegidos** e **acessados** somente por meio de **métodos específicos**.

03.

Garante que os dados sejam usados corretamente e impede que eles sejam modificados de maneira inapropriada.

04.

Em Java, o encapsulamento é implementado usando **modificadores de acesso**

```
public class Pessoa {  
    private String nome;  
    public String sobrenome;  
    protected int idade;  
    String endereco;  
}
```

Modificadores de Acesso

	Classe (class)	Pacote (package)	SubClasse	Todos
Public				
Protected				
Default (padrão)				
Private				

01.

Public: um atributo ou método declarado como público pode ser acessado por **qualquer classe** no programa.

02.

Private: pode ser acessado dentro da **própria classe**. Isso ajuda a proteger os dados da classe de acesso indesejado ou modificações acidentais.

03.

Protected: pode ser acessado somente dentro da **própria classe** e por suas **subclasses**. Isso permite que as subclasses acessem e modifiquem os dados da classe pai.

04.

Default: pode ser acessado dentro do **mesmo pacote**. Isso ajuda a proteger os dados da classe de acesso indesejado de outras classes fora do pacote.

Getters e Setters

01.

Getter é um método que permite **acessar o valor** de um atributo privado de uma classe.

02.

Neste exemplo, o Getter "**getIdade()**" permite que outras classes acessem o valor atual do atributo "**idade**".

03.

Setter é um método que permite **modificar o valor** de um atributo privado de uma classe.

04.

O Setter "**setIdade()**" permite que outras classes modifiquem o valor do atributo "**idade**".

```
public class Pessoa {  
    private int idade;  
  
    public int getIdade() {  
        return idade;  
    }  
}
```

```
public class Pessoa {  
    private int idade;  
  
    public void setIdade(int novaIdade) {  
        idade = novaIdade;  
    }  
}
```

Construtor

```
public class Pessoa {  
    private String nome;  
    private int idade;  
  
    public Pessoa(String nome, int idade) {  
        this.nome = nome;  
        this.idade = idade;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
  
    public int getIdade() {  
        return idade;  
    }  
}
```

```
Pessoa pessoa1 = new Pessoa("João", 25);
```

- 01 Os construtores são métodos especiais usados para **criar** e **inicializar** objetos em Java.
- 02 Eles têm o **mesmo nome da classe** e são executados automaticamente quando um objeto é criado.
- 03 Os construtores podem ter **parâmetros para definir valores iniciais** para os atributos do objeto.
- 04 A palavra-chave "**this**" aponta para uma referência da classe atual



Neste exemplo, a classe **Pessoa** tem dois atributos privados: **nome** e **idade**



O construtor personalizado Pessoa(String nome, int idade) é usado para **inicializar esses atributos** com valores passados como parâmetros.



Foi criado um objeto Pessoa com o **nome "João"** e **idade 25**. Podemos então acessar esses valores usando os métodos **getNome()** e **getIdade()**.

Construtor Padrão X Personalizado

```
public class Retangulo {  
    private String cor;  
    private double largura;  
    private double altura;  
    private double perimetro;  
    private double area;  
  
    public Retangulo() {  
        this.cor = "Branco";  
        this.altura = 1;  
        this.largura = 1;  
    }  
}
```

```
public Retangulo(String cor, double largura, double altura) {  
    this.cor = cor;  
    this.largura = largura;  
    this.altura = altura;  
}
```

```
// Este construtor é usado para criar um retângulo com valores padrão,  
// ou seja, um retângulo branco com largura e altura iguais a 1.  
// Este construtor é usado para criar um retângulo com valores específicos,  
// ou seja, um retângulo com cor, largura e altura especificadas pelo usuário.
```

- Um construtor **padrão** é um construtor sem argumentos;
- Neste exemplo, foi realizado a criação do construtor padrão do retângulo. Os valores padrões do retângulo são:
Altura = 1; Largura = 1; e a Cor = "Branco";

- O construtor **personalizado** é usado para inicializar os atributos do objeto com valores específicos fornecidos pelo usuário.
- Neste exemplo, foi feito um construtor que irá pegar as informações do retângulo, são elas: A cor, largura e altura.



Acesse os materiais no GitHub :)

Exemplos de atividades/códigos com tudo que envolve o
novo Assunto (Encapsulamento)

https://github.com/IsabelaSilvaz/Monitoria_Java_POO/tree/main/Encapsulamento