

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL

AULA 04

LINGUAGEM DE PROGRAMAÇÃO 2
JAVA



PROF. JANIHERYSON FELIPE

CONTEÚDO DESSA AULA

- **ENTENDER O ENCAPSULAMENTO E SUA IMPORTÂNCIA;**
- **ENTENDER OS PACOTES EM JAVA;**
- **MÉTODOS GET/SET;**
- **DISCUSSÕES E DÚVIDAS GERAIS.**

CONSTRUTOR DE UMA CLASSE

Em poo, um construtor é um método especial dentro de uma classe que é **chamado automaticamente** quando um objeto da classe é **criado**. Sua principal responsabilidade é **inicializar** os atributos do objeto e realizar outras tarefas de inicialização necessárias.



CONSTRUTOR DE UMA CLASSE

```
class Carro{  
    String modelo;  
    String marca;  
    double capacidadeTanque;  
  
    public Carro(String modelo, String marca){  
        this.marca = marca;  
        this.modelo = modelo;  
        this.capacidadeTanque = 0;  
    }  
}
```

CONSTRUTOR DE UMA CLASSE

```
public class Main {  
    public static void main(String[] args) {  
        Carro uno = new Carro("Fiat", "uno");  
        System.out.println(uno.modelo);  
        System.out.println(uno.marca);  
        System.out.println(uno.capacidadeTanque);  
    }  
}
```

SOBRECARGA DE CONSTRUTOR

```
public Carro(String modelo, String marca){  
    this.marca = marca;  
    this.modelo = modelo;  
    this.capacidadeTanque = 0;  
}  
  
public Carro(){  
    System.out.println("Objeto vazio criado!");  
}
```

MODIFICADORES DE ACESSO

Em Java, existem quatro modificadores de acesso que podem ser aplicados a classes, métodos e variáveis:

- public
- private
- protected
- default (sem modificador)



MODIFICADORES DE ACESSO

public: O acesso é ilimitado. As classes, métodos e variáveis declaradas como públicas podem ser acessadas de qualquer pacote.

private: O acesso é restrito à própria classe. Isso significa que os membros privados não podem ser acessados por outras classes, mesmo que façam parte do mesmo pacote.

MODIFICADORES DE ACESSO

protected: O acesso é permitido dentro do mesmo pacote, bem como por subclasses, mesmo que estejam em pacotes diferentes. Porém, não pode ser acessado por classes que não sejam subclasses.

default (sem modificador): Também conhecido como pacote-private. O acesso é restrito ao próprio pacote.

PACOTES (PAKAGES)

É um mecanismo de organização de classes e interfaces relacionadas. Ele agrupa classes e interfaces relacionadas em um único namespace, proporcionando organização e modularidade ao código:

Vantagens:

- Organização de Código;
- Evitar Conflitos de Nomes;
- Encapsulamento e Modularidade;
- Reutilização de Código;
- Controle de Acesso;

SOBRECARGA DE MÉTODOS

Permite que uma classe tenha vários métodos com o mesmo nome, desde que tenham assinaturas diferentes. A assinatura de um método inclui o nome do método e a lista de tipos de parâmetros.

```
public class ExemploSobrecarga {  
    no usages  
    public int somar(int a, int b) {  
        return a + b;  
    }  
    no usages  
    public int somar(int a, int b, int c) {  
        return a + b + c;  
    }  
    no usages  
    public double somar(double a, double b) {  
        return a + b;  
    }  
}
```

SOBRECARGA DE MÉTODOS

Permite que uma classe tenha vários métodos com o mesmo nome, desde que tenham assinaturas diferentes. A assinatura de um método inclui o nome do método e a lista de tipos de parâmetros.

```
public class ExemploSobrecarga {  
    no usages  
    public int somar(int a, int b) {  
        return a + b;  
    }  
    no usages  
    public int somar(int a, int b, int c) {  
        return a + b + c;  
    }  
    no usages  
    public double somar(double a, double b) {  
        return a + b;  
    }  
}
```

MÉTODOS ESTATICOS (STATIC)

O modificador **static** em Java é usado para declarar membros de uma classe que pertencem à própria classe em vez de pertencerem a instâncias individuais da classe. Existem vários contextos em que o modificador static pode ser aplicado:

- Variáveis estáticas (ou campos estáticos):
- Métodos estáticos:
- Blocos estáticos:

MÉTODOS ESTATICOS (STATIC)

Variáveis estáticas (ou campos estáticos): Quando uma variável é declarada como estática em uma classe, ela é compartilhada por todas as instâncias dessa classe. Isso significa que, independentemente de quantas instâncias da classe você criar, haverá apenas uma cópia da variável estática.

```
public class ExemploStatic {  
    1 usage  
    public static int contador = 0;  
  
    no usages  
    public ExemploStatic() {  
        contador++;  
    }  
}
```

MÉTODOS ESTATICOS (STATIC)

Variáveis estáticas (ou campos estáticos): Métodos estáticos pertencem à classe em vez de pertencerem a instâncias individuais. Eles podem ser chamados diretamente usando o nome da classe, sem a necessidade de criar uma instância da classe. Métodos estáticos não podem acessar membros não estáticos da classe diretamente, pois eles não têm acesso ao contexto de uma instância específica.

```
no usages
public class ExemploStatic {
    no usages
    public static void metodoEstatico() {
        System.out.println("Este é um método estático.");
    }
}
```

MÉTODOS ESTATICOS (STATIC)

Blocos estáticos: Blocos estáticos são usados para inicializar variáveis estáticas ou para executar código que precisa ser executado apenas uma vez quando a classe é carregada pela primeira vez. Eles são executados quando a classe é carregada pela primeira vez na memória.

```
no usages
public class ExemploStatic {
    1 usage
    public static int valor;

    static {
        valor = 10;
        System.out.println("Blocos estáticos são executados quando a classe é carregada.");
    }
}
```


