Bootcamp Java 23/09/22

Métodos Java

Objetivos

Como criar um método, e como utilizá-lo

Percurso

* Criação
* Sobrecarga
* Retornos

Requisitos

* Lógica de programação
* Java (noções básicas)
* IDE Java

Conceituação e criação de métodos

Objetivos

* Entender o que é um método
* Saber como definir e utilizar métodos
* Aplicar boas práticas em sua criação e uso.

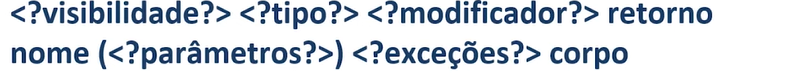
O que é

Uma porção de código, disponibilizada por uma classe. Este é executando quanto é feita uma requisição a ele. São responsáveis por definir e realizar um determinado comportamento.

Criação

Padrão de definição

Visibilidade tipo modificador ou retorno nome (parameros) exceções e seu corpo



O que fica entre interrogação são opcionais, mas o que não está é obrigatório.

Exemplos

Visibilidade: public, protected ou private

Tipo: concreto ou abstrato

Modificador: static ou final

Retorno: tipo de dado ou void(vazio)

Nome: nome fornecido ao método.

Parâmetros: parâmetros que pode receber

Exceções: exceções que pode lançar

Corpo: código que possui ou vazio.

Exemplos

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Métodos são parecidos com funções, com a diferença do paradigma da orientação a objeto.

Padrão usado no curso

Public static retorno nome (parâmetros) {}

Utilização

Para chamar o método devemos chamar ele através de uma classe ou objeto.

Chamando por classe

Nome da classe. Nome do método (parâmetros caso existam)

Chamando por objeto

Nome do objeto. Nome do método (parâmetros caso existam)

No curso usaremos o método a partir de classes

Particularidades

Assinatura: é a forma de identificar unicamente o método

Assinatura = nome+parâmetros

Método

Pulic double calcularTotalVenda(double precoItem1, double precoItem2, double precoItem3){...}

Assinatura

calcularTotalVenda(double precoItem1, double precoItem2, double precoItem3)

parecido com o protótipo em c

Construtor e destrutor: métodos especiais usados na interação a objeto

Mensagem: ato de solicitar ao método que ele execute. Esta pode ser direcionada a um objeto ou a uma classe.

Passagem de parâmetros

Por valor (cópia)

Por referência (endereço)

Exemplo

Int i =10;

Public void fazerAlgo (int i) {

I = i+10;

System.out.println(” Valor de i dentro: “+i);

Aqui dentro é uma cópia, não muda o valor original

}

System.out.println(” Valor de i for: “+i);

Aqui mostra o valor original

Boas práticas

* Nomes descritivos mais curtos
* Notação camelo
* Deve possuir entre 80 e 120 linhas
* Evite lista de parâmetros longas
* Visibilidades adequadas