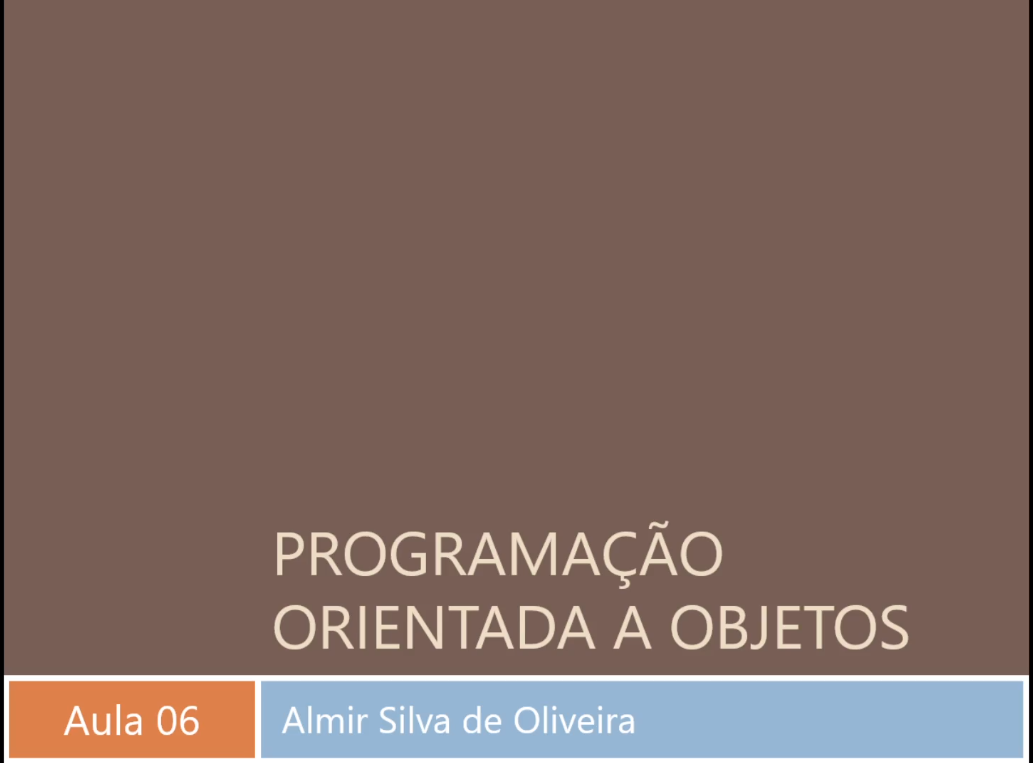
**03-POO - Métodos Construtores- AULA 06.**

****

**Vimos métodos que retornam valor e métodos que não retornam valor**

**retornam valor**

**por ex.. no arquivo CalculadoraSub**

**public** **class** CalculadoraSub {

**int** obterSubracao(**int** a, **int** b) {

**int** res = a - b;

**return** res;

}

}

**e no arquivo principal Teste\_Sub:**

**public** **class** Teste\_Sub {

**public** **static** **void** main(String[] args) {

// instanciar uma classe ou objeto

CalculadoraSub calc = **new** CalculadoraSub();

**int** resultado = calc.obterSubracao(15, 5);

System.***out***.println(resultado);

}

}

**NÃO retornam valor (QUE APENAS EXECUTAM UM PROCEDIMENTO)**

**por ex.. no arquivo CalculadoraSub**

**public** **class** CalculadoraSub {

**void** imprimirSub(**int** a, **int** b){

**int** res = a - b;

System.***out***.printf("%d - %d = %d", a, b, res);

}

}

**e no arquivo principal Teste\_Sub:**

**public** **class** Teste\_Sub {

**public** **static** **void** main(String[] args) {

// instanciar uma classe ou objeto

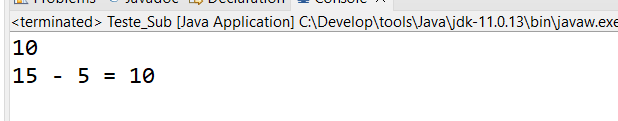
CalculadoraSub calc = **new** CalculadoraSub();

//Método - apenas executa instruções (procedimento)

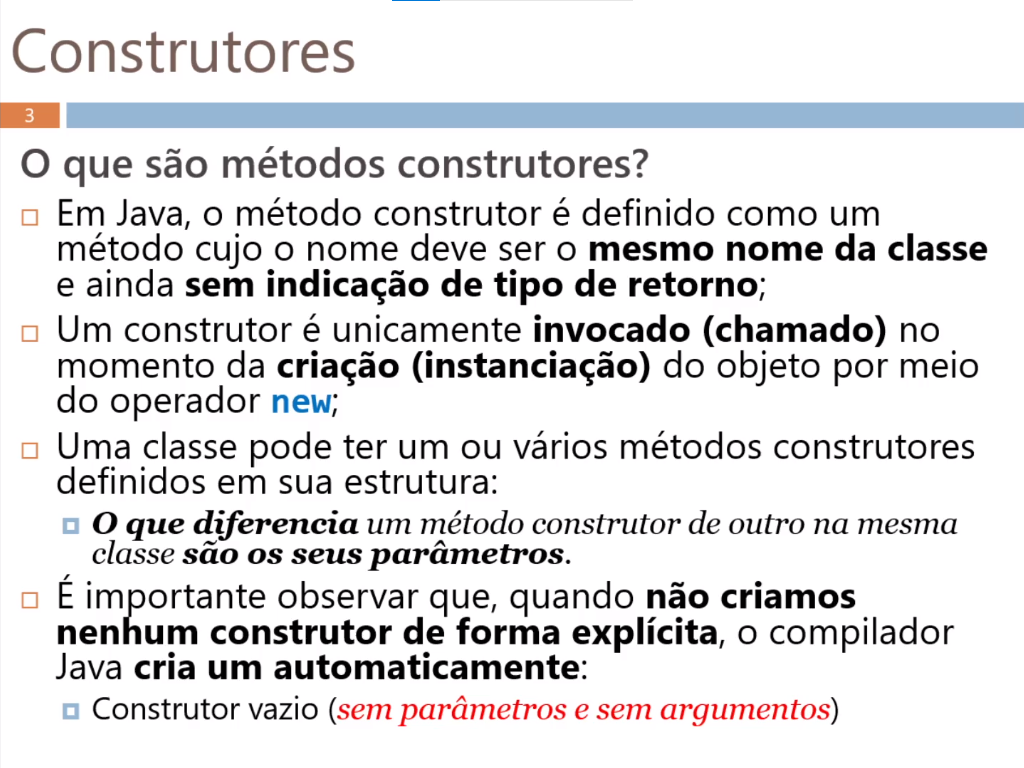
calc.imprimirSub(15, 5);

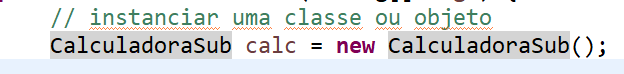
}

}

****







METODO CONSTRUTOR TEM A MISSAO DE INICIALIZAR O OBJETO A SER INSTANCIADO OU QDO ESTE É CRIADO.

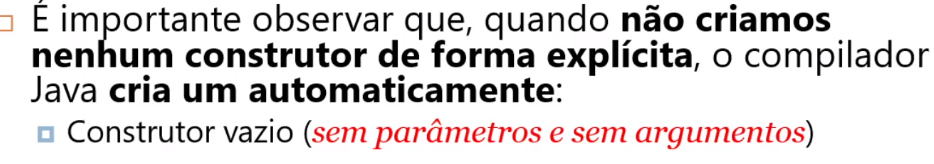
**METODO CONSTRUTOR APENAS INICIALIZA O OBJETO**

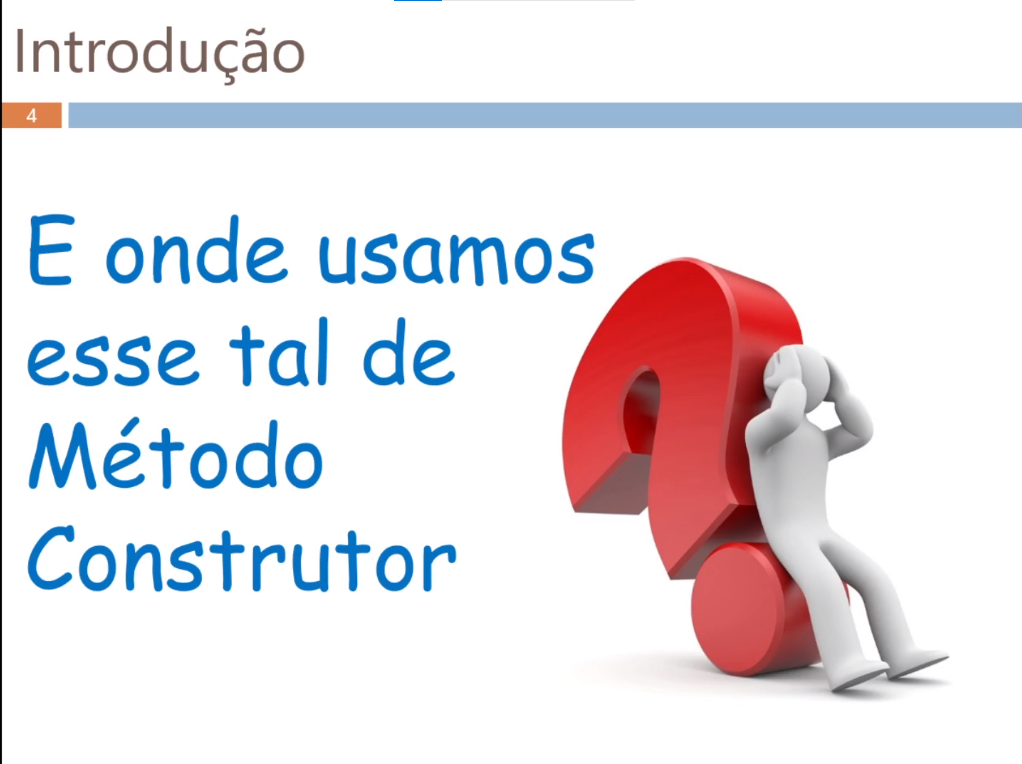
Scanner leitor = **new** Scanner(System.***in***)

CalculadoraSub calc = **new** CalculadoraSub();

ELE PODE OU NÃO RECEBER PARÂMETROS

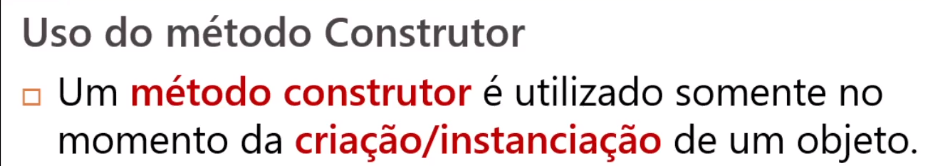
🡺 ();OU (System.***in***);

****

****

Scanner leitor = **new** Scanner(System.***in***)

CalculadoraSub calc = **new** CalculadoraSub();



EX:

Aluno maria =

CLASSE Nome da variável

Tipo de dado repr. Este obj na programaçao

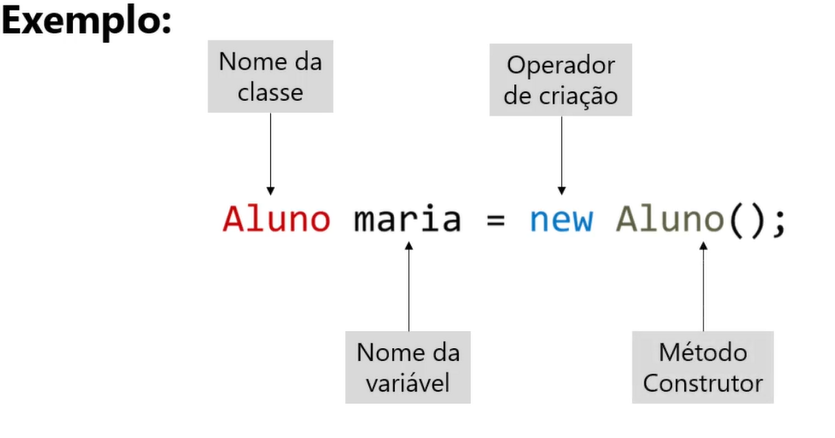
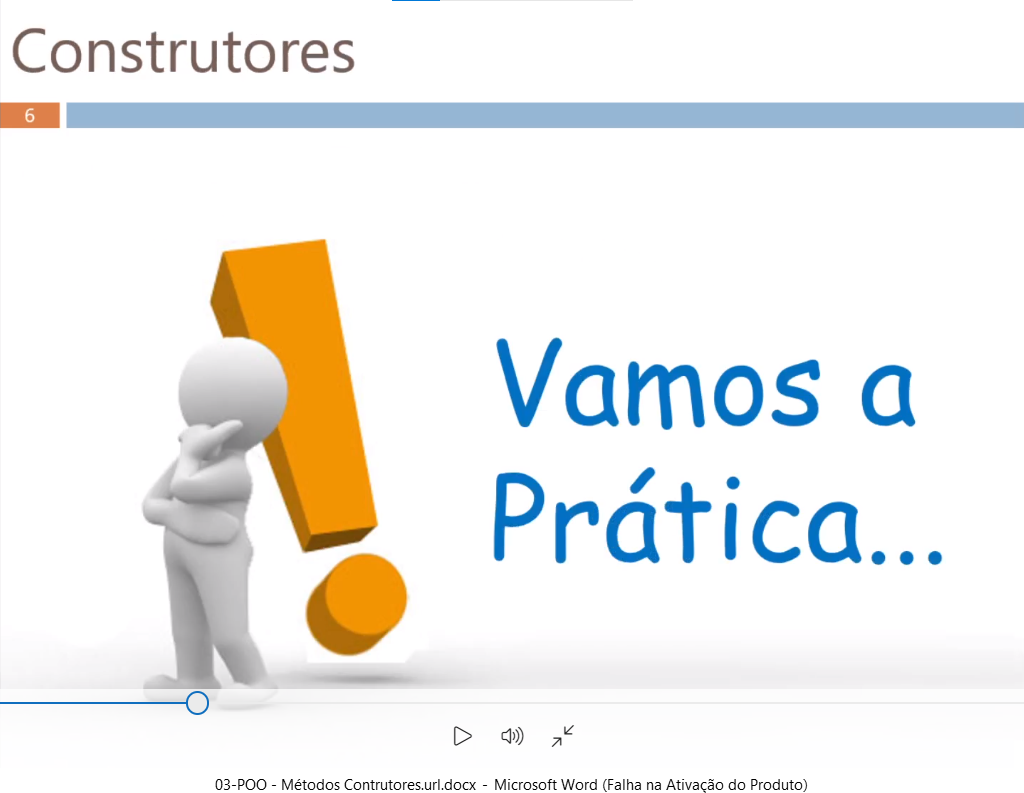
**new** Aluno() =

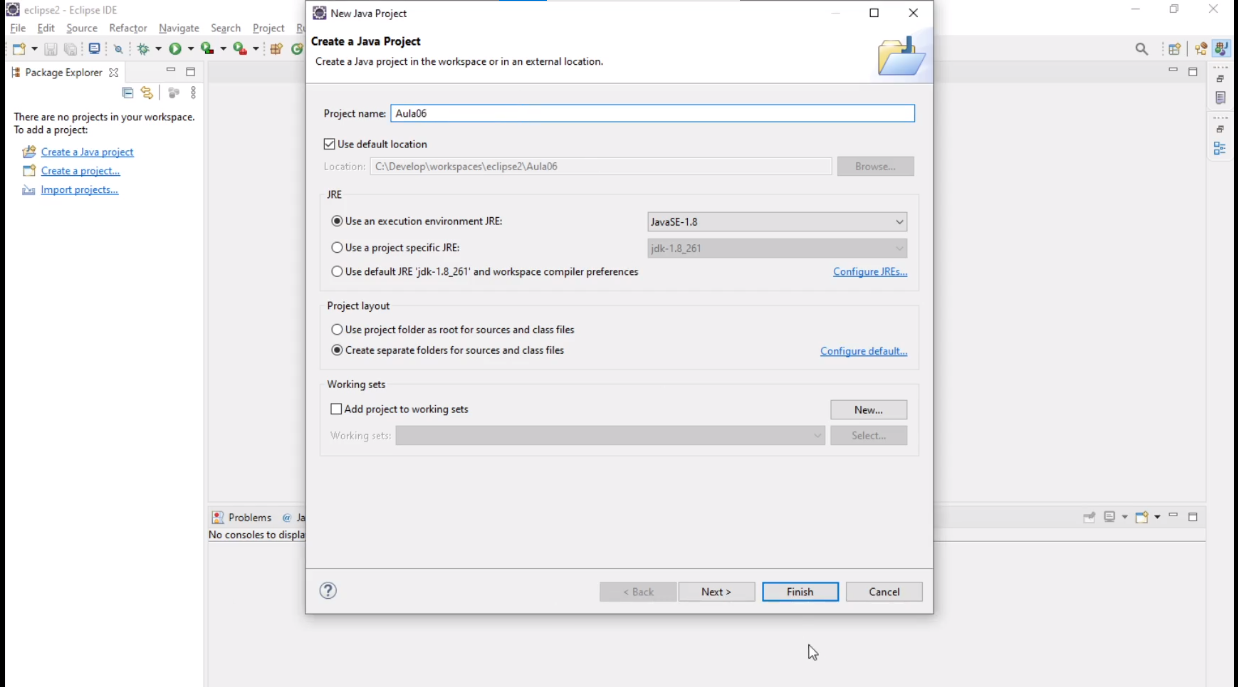
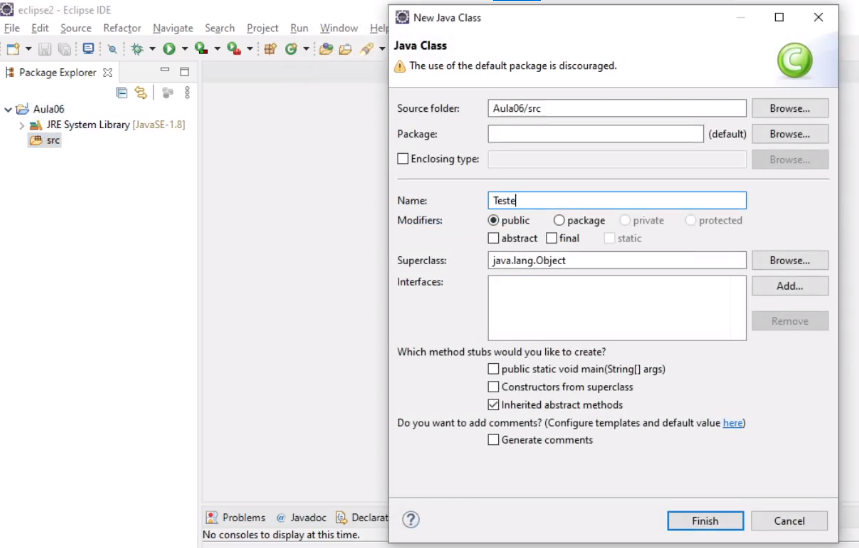
Operador Este é efetivamente o

Que efetivamente construtor

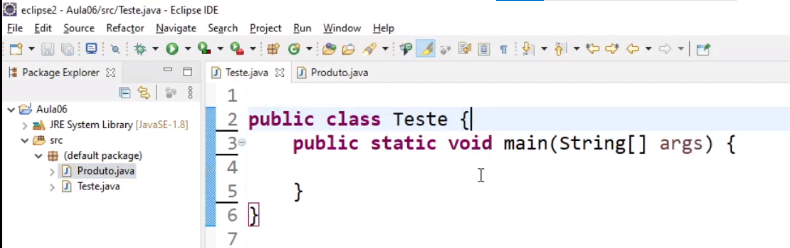
Gera este obj na

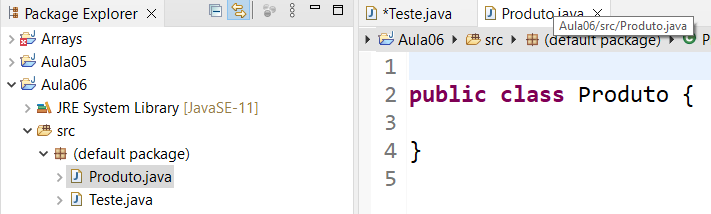
memoria

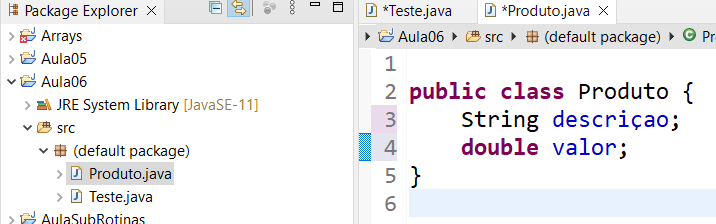
 

Vou criar uma classe que vai representar um objeto produto.

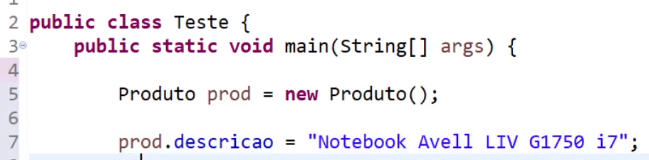


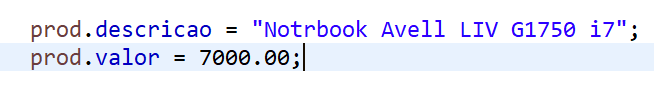


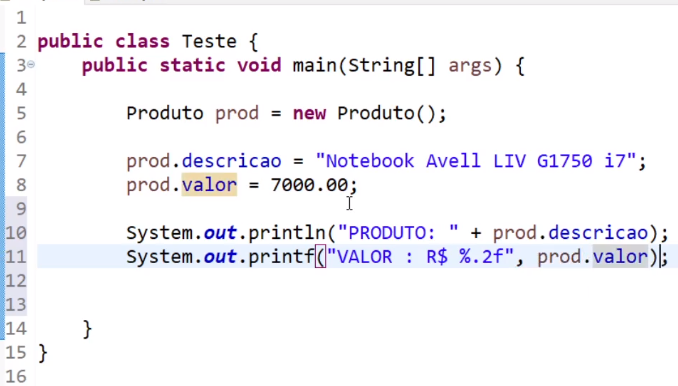


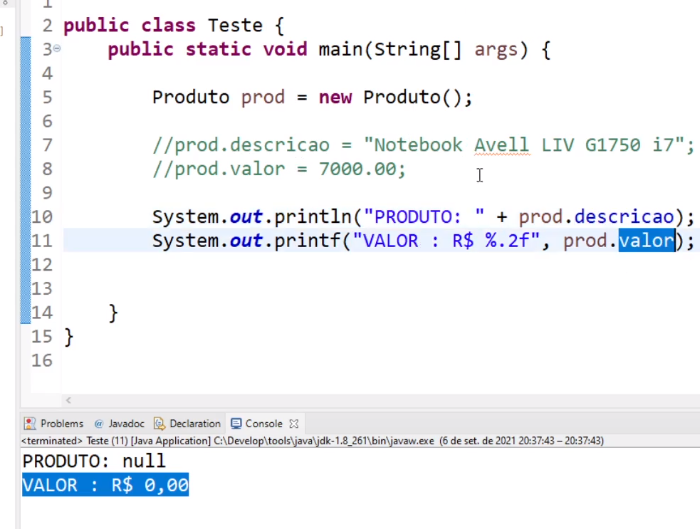
🡺 etodo construtor

🡺 aqui já esta instanciado nosso objeto através da nossa variável “prod”





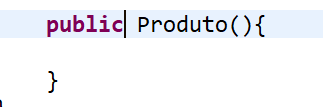




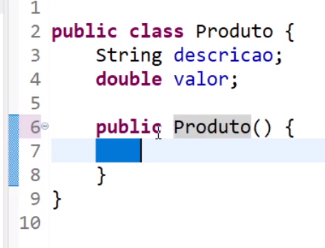
Quero inicializar o objeto já com algum valor...

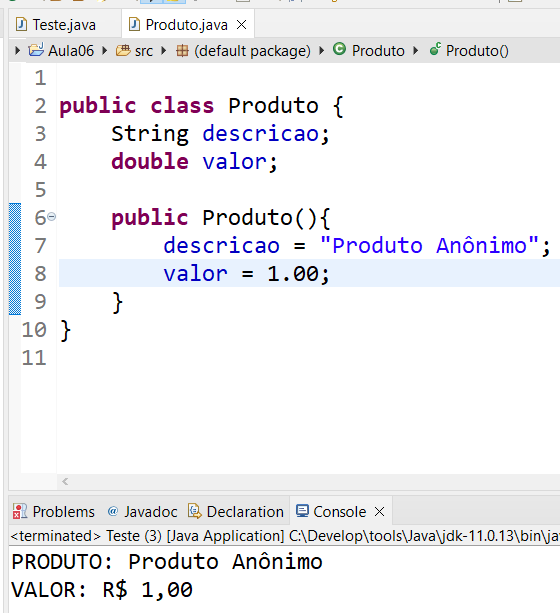
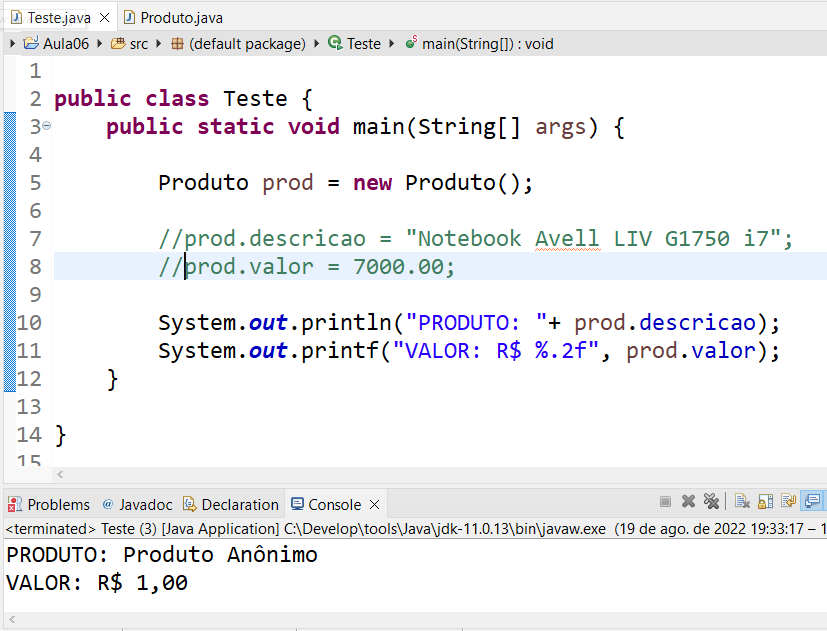
Preciso criar um método consultor 🡺 que é

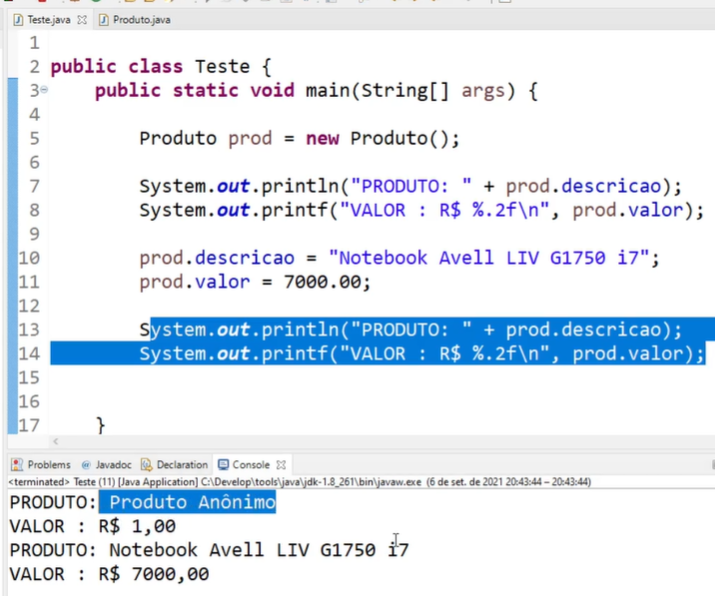
**Produto()**

🡺 método consultor padrão ou vazio

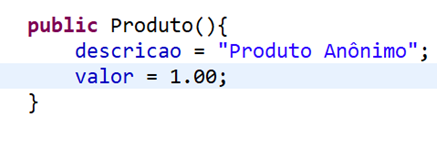
Porque ele não tem entrada de parâmetros () e nem tem implementação{}



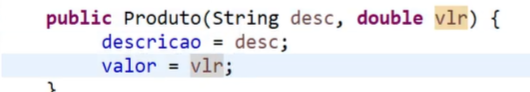




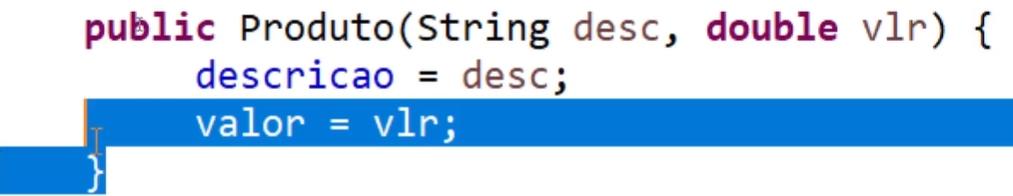
SÓ QUE EU NÃO QUERO OS VALORES DE ENTRADA SEJAM FIXOS COMO NA FIG ABAIXO;

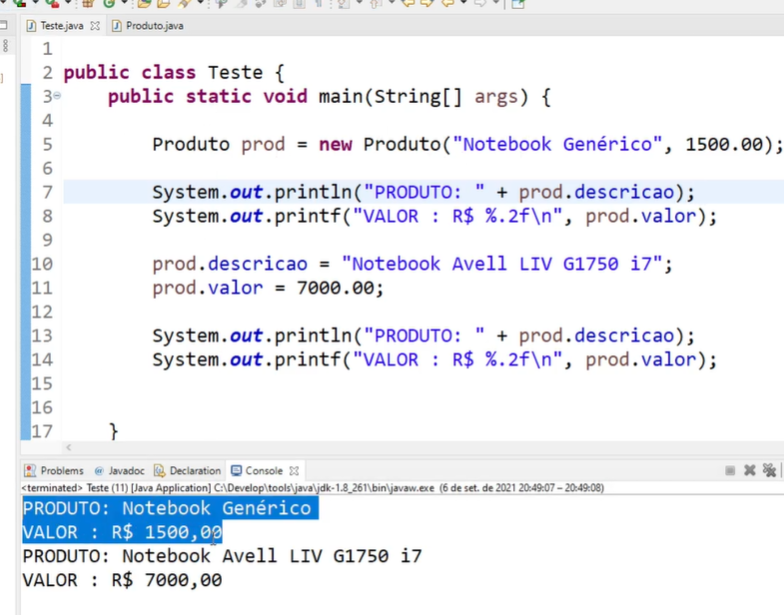


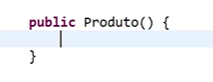
ENTAO EU FAÇO A ALTERACAO ABAIXO;



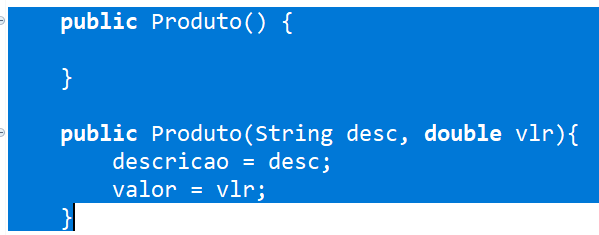
Alteramos o método consultor padrão () para o método consultor explicito Produto(**xxx** xxx,**xxx** xxx)







Método construtor vazio ou método construtor padrao;



Na figura acima temos a sobrecarga de métodos...

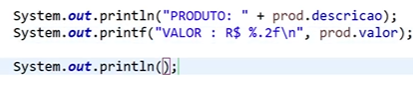
Suas entradas de dados tem de ser diferentes:

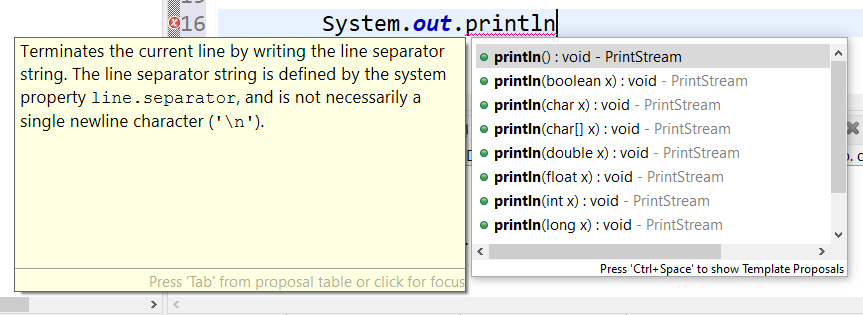
Assinatura de métodos

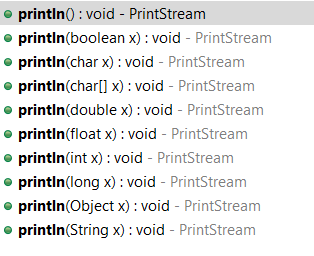
**e**

****

**Um outro exemplo de método sobregcarregado é o println**

****

****

****

**O METODO CONSTRUTOR SEMPRE SERÁ UTILIZADO COM O OPERADOR DE CRIAÇAO**

**🡺 new**

**O METODO CONSTRUTOR SÓ É PRA INICIALIZAÇÃO DE UM DETERMINADO OBJETO.**

**public** **class** Teste01 {

**public** **static** **void** main(String[] args) {

Produto prod1 = **new** Produto();

Produto prod2 = **new** Produto("Notebook Avell LIV G1750 i7");

Produto prod3 = **new** Produto("Notebook Genérico", 1500.00);

prod1.descricao = "Mouse Microsoft 3000pm";

prod1.valor = 350.00;

prod2.valor = 7500.00;

System.***out***.println("PRODUTO 1: "+ prod1.descricao);

System.***out***.printf("VALOR : R$ %.2f\n", prod1.valor);

System.***out***.println("PRODUTO 2: "+ prod2.descricao);

System.***out***.printf("VALOR : R$ %.2f\n", prod2.valor);

System.***out***.println("PRODUTO 3: "+ prod3.descricao);

System.***out***.printf("VALOR : R$ %.2f\n", prod3.valor);

}

}

**public** **class** Produto {

String descricao;

**double** valor;

**public** Produto() {

}

**public** Produto(String desc) {

descricao = desc;

}