

FIA/P GRADUAÇÃO

DOMAIN DRIVEN DESIGN

Prof. Me. Thiago T. I. Yamamoto

#06 – CONSTRUTORES

TRAJETÓRIA



- ✓ Orientação a Objetos
- ✓ Introdução ao Java
- ✓ IDE e Tipos de Dados
- ✓ Classes, atributos e métodos
- ✓ Encapsulamento
- ✓ Construtores

#06 - AGENDA

- Construtores
 - Sem parâmetros
 - Com parâmetros
- Exemplos





CONSTRUTORES

- Desempenham **papel essencial** no processo de **instanciação** de uma classe;
- Os **construtores** também são utilizados para **inicializar os atributos** com **valores padrão** ou com **valores informados**;
- São **métodos** especiais que são invocados juntamente com o operador **new**;
- **Construtores** não possuem **valor de retorno** (nem mesmo **void**) e possuem o **mesmo nome da classe**;
- Toda classe **tem pelo menos um construtor**;
- Quando **não** especificamos nenhum construtor, a linguagem Java fornece um **construtor default (padrão)**, que não recebe parâmetros (vazio);
- Se declararmos algum **construtor**, esse construtor **padrão não será mais fornecido pelo Java**;

- São métodos representados pelo mesmo da classe e que permitem parametrizar o objeto no momento da sua instanciação.

```
Produto objProduto = new Produto();
```

Método construtor sem parâmetros, caso você não crie o método, **ele será automaticamente gerado pela JVM**



```
public class Carro {  
  
    String modelo;  
    float motor;  
  
    Carro(){  
        System.out.print("Criando uma instância");  
    }  
}
```

- Utilização do construtor acima:

```
Carro carro = new Carro();
```




```
public class Carro {  
  
    String modelo;  
    float motor;  
  
    Carro(String modeloCarro){  
        modelo = modeloCarro;  
    }  
  
    Carro(String modeloCarro, float motor){  
        modelo = modeloCarro;  
        this.motor = motor;  
    }  
}
```

- Utilização do construtor acima:

```
Carro carro = new Carro("Gol", 1.8f);
```

```
public class Carro {  
  
    String modelo;  
    float motor;  
  
    Carro(String modeloCarro){  
        modelo = modeloCarro;  
    }  
  
    Carro(String modeloCarro, float motor){  
        this.modeloCarro;  
        this.motor = motor;  
    }  
}
```



- Utilização do construtor acima:


```
Carro carro = new Carro("Gol", 1.8f);
```

```
public class Pessoa {  
  
    String nome;  
    int idade;  
    Carro carro;  
  
    Pessoa(String nome, int idade, Carro carro){  
        this.nome = nome;  
        this.idade = idade;  
        this.carro = carro;  
    }  
}
```



- Utilização do construtor declarado no slide anterior:

```
Carro carro = new Carro("Gol", 1.8f);  
Pessoa pessoa = new Pessoa("Pedro", 20, carro);
```



- Passando **null** como parâmetro:

```
Pessoa pessoa = new Pessoa("Maria", 50, null);
```

PRÁTICA

- Implemente as seguintes classes em Java:

Funcionario
- nome : String - matricula : long - profissao : Profissao - salario : double
+ Funcionario() + Funcionario(matricula : long) + Funcionario(matricula : long, nome : String) + Funcionario(matricula : long, nome : String, profissao : Profissao) + exibirDados() : String

Profissao
- nome : String
+ Profissao() + Profissao(nome : String)

- Crie uma classe de teste para instanciar as duas classes;



Copyright © 2020 - 2025 Prof. Thiago T. I. Yamamoto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

*“Para conquistar o sucesso, você precisa aceitar todos os desafios que vierem na sua frente. Você não pode apenas aceitar os que você preferir”
- Mike Gafka*