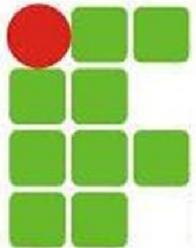
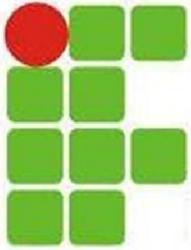

Programação Orientada a Objetos

Professor: Diego Oliveira



**Conteúdo 07:
Construtores**

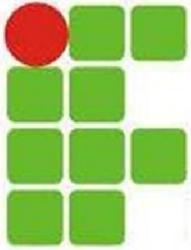




Construtores

- Construtores são utilizados basicamente para inicializar as variáveis de uma classe
- O Java cria automaticamente o construtor padrão Classe(), que não recebe nenhum parâmetro e inicializa as variáveis com seus valores padrão como 0 e NULL
- Para inicializar as variáveis com valores diferentes, precisamos criar nossos próprios construtores

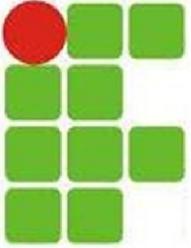




Construtores

- O Construtor é um método:
 - Obrigatório para toda classe
 - Utilizado para inicializar as variáveis da classe
 - Que pode chamar outros métodos
 - Sem tipo de retorno
 - Que aceita parâmetros
 - Que aceita sobrecarga
- ***OBS.: ao criar qualquer construtor, o Java não criará o construtor padrão Classe()***



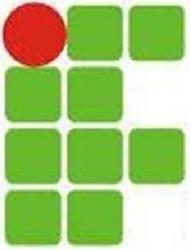


Construtores

- Observe a classe Pessoa:

```
1 public class Pessoa {
2     private String nome;
3     private int idade;
4     private double peso;
5
6     + public String getNome() { ...3 lines }
9
10    + public void setNome(String nome) { ...3 lines }
13
14    + public int getIdade() { ...3 lines }
17
18    + public void setIdade(int idade) { ...3 lines }
21
22    + public double getPeso() { ...3 lines }
25
26    + public void setPeso(double peso) { ...3 lines }
29
30 }
```



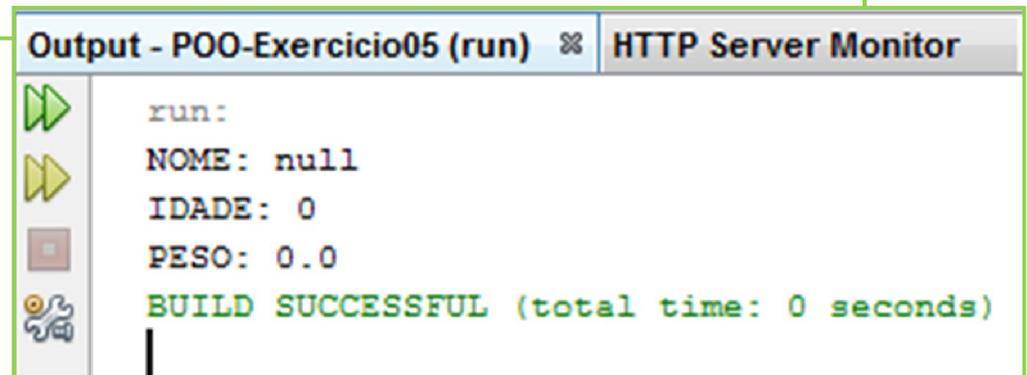


Construtores

- Ao criarmos um novo objeto do tipo Pessoa:

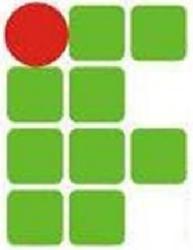
```
1 public class TestePessoa {
2     public static void main(String[] args) {
3         Pessoa p1 = new Pessoa();
4         System.out.println("NOME: " + p1.getNome());
5         System.out.println("IDADE: " + p1.getIdade());
6         System.out.println("PESO: " + p1.getPeso());
7     }
8 }
```

- A saída é assim:



```
Output - POO-Exercicio05 (run) HTTP Server Monitor
run:
NOME: null
IDADE: 0
PESO: 0.0
BUILD SUCCESSFUL (total time: 0 seconds)
```





Construtores

- Se criarmos um construtor vazio:

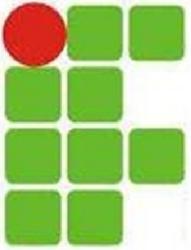
```
1 public class Pessoa {
2     private String nome;
3     private int idade;
4     private double peso;
5
6     public Pessoa() {}
7
8 }
9
10 public String getNome
```

Output - POO-Exercicio05 (run) HTTP Server Monitor

```
run:
NOME: null
IDADE: 0
PESO: 0.0
BUILD SUCCESSFUL (total time: 0 seconds)
```



- A saída continua do mesmo jeito:



Construtores

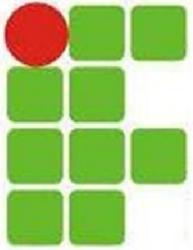
- Porém se alterarmos o construtor:

```
1  public class Pessoa {
2      private String nome;
3      private int idade;
4      private double peso;
5
6      public Pessoa() {
7          this.nome = "Diego";
8          this.idade = 30;
9          this.peso = 70;
10     }
```

Output - POO-Exercicio05 (run) ✖	HTTP Server Monitor
 run:	
 NOME: Diego	
 IDADE: 30	
 PESO: 70.0	
 BUILD SUCCESSFUL (total time: 0 seconds)	



- A saída será modificada:



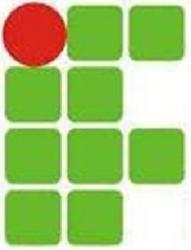
Construtores

- O ideal é termos construtores com parâmetros:

```
6      public Pessoa(String nomeRecebido,  
7                                     int idadeRecebida,  
8                                     double pesoRecebido) {  
9          this.nome = nomeRecebido;  
10         this.idade = idadeRecebida;  
11         this.peso = pesoRecebido;  
12     }
```



- Assim é possível inicializar um objeto Pessoa com os valores desejados



Construtores

- Vejamos o exemplo de inicialização de uma Pessoa com parâmetros:

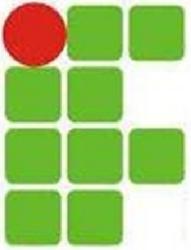
```
1 public class TestePessoa {
2     public static void main(String[] args) {
3         Pessoa p1 = new Pessoa("Diego Oliveira", 40, 80.0);
4         System.out.println("NOME: " + p1.getNome());
5         System.out.println("IDADE: " + p1.getIdade());
6         System.out.println("PESO: " + p1.getPeso());
7     }
8 }
```

Output - POO-Exercicio05 (run) ⌵ HTTP Server Monitor

```
run:
NOME: Diego Oliveira
IDADE: 40
PESO: 80.0
BUILD SUCCESSFUL (total time: 0 seconds)
```



- A saída será:

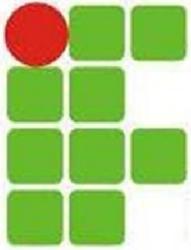


Construtores

- Uma classe pode ter diversos construtores:

```
6  [-] public Pessoa(String nomeRecebido){
7      this.nome = nomeRecebido;
8  }
9
10 [-] public Pessoa(String nomeRecebido, int idadeRecebida){
11     this.nome = nomeRecebido;
12     this.idade = idadeRecebida;
13 }
14
15 public Pessoa(String nomeRecebido,
16                int idadeRecebida,
17                double pesoRecebido){
18     this.nome = nomeRecebido;
19     this.idade = idadeRecebida;
20     this.peso = pesoRecebido;
21 }
```



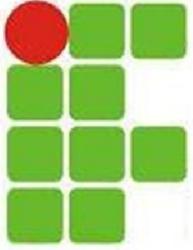


Exercício

- Crie uma classe que possua pelo menos 3 atributos de tipos diferentes
- Crie pelo menos 4 construtores:
 - Um que receba cada um dos 3 parâmetros separadamente
 - Um que recebe os 3 parâmetros juntos
- Crie uma classe de teste que demonstre a utilização dos 4 construtores



• ***OBS.: não esqueça de utilizar os SET e GET***



Perguntas?

